SC-RR-68-545
July 1968

PLANETARY QUARANTINE

AN INTERACTIVE COMPUTER INFORMATION SYSTEM FOR PLANETARY QUARANTINE FOR LUNAR PROGRAMS

A. L. Roark, 1741
A. L. Wyer, 1741
Sandia Laboratories
Albuquerque, New Mexico

SANDIA LABORATORIES

OPERATED FOR THE U. S. ATOMIC ENERGY COMMISSION BY SANDIA CORPORATION | ALBUQUERQUE, NEW MEXICO, LIVERMORE, CALIFORNIA

Issued by Sandia Corporation,
a prime contractor to the
United States Atomic Energy Commission

SC-RR-68-545

AN INTERACTIVE COMPUTER INFORMATION SYSTEM FOR PLANETARY QUARANTINE FOR LUNAR PROGRAMS

A. L. Roark

and

A. L. Wyer

Planetary Quarantine Department
Sandia Laboratories
Albuquerque, New Mexico

July 1968

Supported by Purchase Order R-09-019-040, Bioscience Division, Office of Space Science and Applications, NASA Headquarters.

PRECEDING PAGE BLANK NOT FILMED.

PREFACE

The NASA Planetary Quarantine Officer has three kinds of responsibility with respect to lunar exploration programs. He is responsible for:

1. Certifying that the biocontamination levels of American unmanned probes do not exceed a specified maximum.
2. Certifying that the biocontamination levels of American manned probes have been held to a practical minimum consistent with achieving the major objectives of the missions.
3. Maintaining an inventory of probable postlanding biocontamination levels at each manned and automated landing site and a total inventory for the moon.

Implementation of these responsibilities requires a system with certain capabilities, which are:

1. The ability to gather biocontamination data from spacecraft system elements and from environmental areas influencing spacecraft biocontamination levels. This implies bioassay capabilities.
2. The ability to translate bioassay data into confident estimates of biocontamination levels on spacecraft at several points during their missions. This implies predictive mathematical models and computerized calculations.
3. The ability to make available to the Planetary Quarantine Officer, and others, information for confident and timely decisions and certifications. This implies a data management system capable of receiving and storing data, using models for appropriate calculations, and storing and outputting the results in appropriate forms when needed.

This document describes such a data management system.

It has been the intent of the authors to define the system in sufficient detail in this document to permit program coding directly from the document. Thus, it is hoped that this document can become the central feature of a programming contract.

A number of subroutine models are referred to in this document, most specifically in the flow charts defining the details of the system. As of the time of publication, these models are not completely defined. In general, those models having to do with spacecraft biocontamination levels up to and including the moment of launch are at least conceptually defined and are under intense development. With one major exception, Martin Tierney's[1] well-developed model for lunar contamination from automated spacecraft, models considering all phases of the mission after launch are yet to be developed.

We wish to thank particularly the following people: Rudy Puleo, USPHS, head of the Sterility Control Laboratory at Cape Kennedy, and Dick Graves, USPHS, head of the inspection and certification activity associated with the Sterility Control Laboratory, who forced us to recognize the microbiological realities of planetary quarantine life, and Ray Brose of the Control Data Corporation, also at Cape Kennedy, who kept us aware of the realities of the CDC 3100.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

not done yet - quarantine

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

AN INTERACTIVE COMPUTER INFORMATION SYSTEM FOR PLANETARY QUARANTINE FOR LUNAR PROGRAMS

CHAPTER 1. INTRODUCTION

An interactive computer system is one which allows the user of a digital computer to have direct access to the computer and to carry on exchanges of information with the computer. The three major elements in such a system are the user, the hardware, and the software system. We shall discuss each of these elements in turn as they apply to an interactive information computer system for the Planetary Quarantine Office of NASA.

1.1 The User

The primary user of this system will be the NASA Planetary Quarantine Officer. The system must provide information to support him in:

a. Predicting and reporting the qualitative and quantitative bioburden on each Apollo spacecraft at launch.
b. Predicting and reporting the qualitative and quantitative bioburden on each Apollo spacecraft at the lunar surface.
c. Maintaining an inventory of biological contamination at each Apollo and automated landing site and a total inventory for the moon.
d. Predicting the probability of lunar sample contamination to aid in the back-contamination problem.

The main requirements imposed on an interactive system are that information be stored, that models be used to predict the various quantities listed above, and that, at his request, this information be made available to the Planetary Quarantine Officer.

Other primary users of the information system are the members of the Public Health Service Sterility Control Laboratory at Cape Kennedy. They have a need to obtain the same information as the Planetary Quarantine Officer. The Public Health Service has another important part in the exchange between the user and the computer; that is, providing the data from which the prediction and reports can be made.

1.2 The Hardware

In May 1967, a CDC 3100 computer became available for planetary quarantine use. Since that time, some modification of the equipment has taken place, and the configurations are now as follows:

Memory	32k words of 24 bits each
Magnetic tapes	Four 604 tape units
Disks	Two 852 storage units
Card reader	One 405
Paper tape station	One
On-line display in Washington	One CDC-210 with printer capability

An analysis of the needs of an Apollo information system by the Planetary Quarantine Department at Sandia Laboratory indicates that this hardware configuration should be adequate. The only other modification that might be advisable, depending upon the number of requests for information and the amount of time available, would be the addition of floating-point hardware.

1.3. The Software System

The standard software systems are MSOS, 3100-3200 FORTRAN, and COMPASS. The user software system is the vehicle by which the exchanges of information take place between the user and the computer. It should be designed to satisfy the needs of the user within the restrictions imposed by the hardware. In order to discuss more fully our views of this part of the system, we shall provide a specification for a software package that would satisfy the requirements we have discussed. Our specification consists of four parts:

a. A routine to establish the data files.
b. A routine to store information in the data files.
c. A routine to update the lunar inventory.
d. A communication system.

We have attempted to keep the system flexible. Thus, we have made provisions in the design for updating flight specifications and for changes in the models. We have also assumed that time will be our major concern (especially in the interactive mode) rather than restrictions on storage capabilities. In order to specify each part of this over-all system, we provide flow charts and present a discussion of each in the chapters to follow. We shall, in the remainder of this chapter, briefly outline the purposes which each of these routines fulfill.

The philosophy of this system is that all data, predictions, and flight plans concerning a particular Apollo mission will be stored on magnetic tape which will serve as archives of planetary quarantine information. We shall call these tapes the flight tapes. Environmental tapes contain all of the environmental information. We shall store on disks the current flight plans and predictions, which will be available to the interactive portion of this system, along with environmental information and directories of where information is stored. With these over-all thoughts in mind, we are now ready to outline briefly the tasks which each of the four parts of the system will fulfill.

1.3.1 File Preparation Routine

The duties of the File Preparation Routine can briefly be outlined as follows:

a. To read in and store the flight specifications on disks and flight tapes. Included in this information are module, suit, and astronaut identifications, predicted launch date, landing location, and launch pad identification.
b. To establish the directories.
c. To read in environmental location identifications and the building in which they are situated.
d. To sort flight tapes and disk records and to update directories if flight specifications should change.
e. To remove cancelled flights from the active files and directories.

This routine will also provide much of the logical capability for handling unusual conditions.

1.3.2 Data Storage Routine

The routine to store the data on the flight tapes and active disk files is the workhorse of the system. Input to the routine consists of an indication of which four types of data are to be stored (i.e., whether it is suit, module, environmental, or astronaut data) and the date on which the data were taken. The routines then branch to a subsection which processes the indicated type of data and stores the needed information in the correct files. This gives us modularity that will be useful in any modifications which may be necessary in the future.

In processing spacecraft data, the information which is read-in is the spacecraft identification, the size of the area sampled, the number of colonies observed, and the results of the qualitative tests on each colony. The routine will then use a subroutine now in development[2] to identify the organisms in each colony.

In order to help the Public Health Service personnel at Cape Kennedy in their consultations with other NASA groups and in their own decision-making processes, all predictions should be available to them automatically. The Data Storage Routine takes care of this task. It stores and lists the raw data, as well as the updated predictions.

The types of models, which will be FORTRAN subroutines, we now envision as being needed for these predictions are:

a. Quantitative estimation of bioburden on module or spacecraft on the sampling date.
b. Quantitative prediction of bioburden on module or spacecraft at launch.
c. Quantitative prediction of bioburden on module or spacecraft at lunar surface.
d. Qualitative estimation of bioburden on module or spacecraft on the sampling date.
e. Qualitative prediction of bioburden on module or spacecraft at launch.
f. Qualitative prediction of bioburden on module or spacecraft at lunar surface.
g. Probability of having identified all types of organisms on the spacecraft.

This list of models could easily be expanded. Also, each model will make use of revisable submodels.

The section of the routine which handles suit data is similar to that used for the spacecraft, and we shall not discuss it in more detail.

The section of the routine which handles environmental data is set up to process four types of environmental data: (1) stainless steel strips; (2) noncascaded air samplers, e.g., Reynier or Casellas; (3) agar plates; and (4) Andersen cascaded-sieve samplers. This section calls for FORTRAN subroutines which calculate the qualitative information and the quantitative parameters needed in the quantitative models. The raw data and updated predictions are printed and stored on the environmental tape. The qualitative information and some of the air-sampling quantitative data are stored in the active disk files. The exact form of the agar plate and the Andersen sampler data is still undefined.

The availability and formats of astronaut data are still undefined. No attempt has been made to discuss them, but a place has been left where they can be added later.

1.3.3 Lunar Inventory Routine

The Lunar Inventory Routine consists of a historical record of all contamination which has reached the lunar surface. This routine will store information concerning all non-Apollo lunar flights, e.g., Ranger, Surveyor, anchored interplanetary monitoring platform (AIMP), lunar orbiter, and unmanned (Luna) and manned Russian spacecraft, as well as all Apollo spacecraft which have reached the lunar surface.

Under the Apollo section we make provision for adding information to the lunar inventory for the following set of circumstances:

a. A normal Apollo landing (here, we transfer the information from an active file to the historical lunar inventory file).
b. An abnormal landing.
c. The return to the lunar surface of the lunar module ascent stage.
d. The Saturn SIVB stage, the Saturn-to-lunar-module adaptor, and the instrument unit hit the lunar surface.

All of the information is stored on disks for quick recovery. Provision is made to handle models (in subroutine form) to predict the quantitative load at the next Apollo landing site and to predict the probability of contamination of samples collected by men aboard the next Apollo spacecraft. Provision is also made to update the predicted quantitative microbial load alive on the lunar surface.

1.3.4 The Communication System

The communication system provides communication with the Planetary Quarantine Officer. It is a modular system with subsections providing access to Apollo information and to lunar inventory information.

As we see the over-all system on the 3100, this part of the program would be in MSOS as a background program, i.e., it is stored in memory as a part of the monitor and, when in use, "steals" time from the active programs in order not to interfere with normal use of the computer. In order to facilitate this and to conserve time, the Data Storage and Lunar Inventory Routines store the information which will be available over the remote access console. This program simply arranges the information into standard formats and transmits the messages.

In the actual communication, we prefer the question and answer approach; the computer asks for the information it needs from the user and the user responds. This is our philosophy here. The computer and the user thus carry on a structured but meaningful dialog. While the computer cannot generate ad lib responses, the conversation can be programmed to be both reasonable and confortably realistic. In designing the exact dialog we place definite limits on the vocabulary which the computer will understand, simply to make the programming and storage requirements less demanding.

1.4 Apollo Configuration and Flows

The system briefly outlined above has been designed to accept and handle microbiological assay data from nine subsections or modules of Apollo spacecraft and Saturn V launch vehicles and from astronauts and their suits. The system accommodates other information regarding mission schedules and the physical locations of modules as they progress through various facilities at the Kennedy Space Center. This section will identify the modules and will describe a typical flow from arrival at the Space Center to launch. The nine modules dealt with by this system are:

CM -- Apollo spacecraft command module.
SM -- Spacecraft service module.
CSM -- Command and service module, an entity resulting from mating the CM and the SM.
LMA -- Lunar module ascent stage.
LMD -- Lunar module descent stage.
LM -- Spacecraft lunar module, an entity resulting from mating the LMA and the LMD.
SLA -- Spacecraft lunar module adaptor.
IU -- Instrument unit.
SIVB -- Launch vehicle SIVB third stage.

The modules listed above are located, typically, in the following facilities during their processing through the space center:

MSOB -- Manned spacecraft operations building.

VAB -- Vehicle assembly building, in the north (VABN) or south (VABS) bay.

39A and 39B -- Launch pads.

The command module, service module, lunar module ascent and descent stages, spacecraft lunar module adaptor, and instrument unit typically start their space center processing in the manned spacecraft operations building. Each module is worked on and checked out individually. The lunar module ascent and descent stages are then assembled and the resulting lunar module is tested as a unit. The command module and service module are similarly checked out individually and are mated to form the command and service module, which is then tested as a unit.

Ultimately, all of the above modules are assembled into what is termed the "stack." The spacecraft lunar module adaptor is installed on the instrument unit. The lunar module is installed in the adaptor. The command and service module is attached atop the spacecraft lunar module adaptor.

The stack is transported to the vehicle assembly building where the Saturn V launch vehicle is being assembled. The SIVB stage is the topmost stage of the launch vehicle. The stack is installed atop the SIVB stage. The escape system attached to the top of the command module completes the space vehicle assembly.

When all vehicle assembly building operations are complete, the space vehicle is transported either to pad 39A or 39B on the launch complex. Following extensive checkouts and fueling operations, the vehicle is launched on its mission.

Microbiological contamination on the various modules is assayed at various intervals during space center processing. Assays are made during operation in the manned spacecraft operations building, in the vehicle assembly building, and on the launch pads.

CHAPTER 2. FILE PREPARATION ROUTINE

The purposes of the File Preparation Routine are to read in, store, list, and update Apollo flight specifications; to establish and update directories of flight information; to store environmental sampling location identifications; and to establish and update environmental directories. The purpose of this chapter is to describe parameter formats, directories and their formats, disk and tape record formats, file formats, and models, and then to discuss the programming specifications in the last section. Many of these formats will be used in succeeding chapters and our description will be referred to rather than repeated.

2.1 Parameter Formats

We shall assume that all of the variables in this system follow the standard 3100 FORTRAN formats as described in Reference 3. To provide a guiding philosophy in the design of variable identification, we shall use the following as basic rules:

a. All quantities predicted by, and parameters used in, quantitative microbial estimative and predictive models are real, floating point variables.
b. All quantitative counts from sampling procedures are inputted and stored as integer quantities.
c. All results of the series of qualitative tests are stored as octal variables.
d. All spacecraft module and space suit identifications are in the form of six characters. They are stored in a type real variable and are right-adjusted. They will have alphabetic characters and three numeric characters.
e. All astronaut identifications are stored as right-adjusted type real variables.
f. All dates are stored as six characters in a type real variable and are right-adjusted.
g. The latitude and longitude of the landing coordinates are each six characters, are stored in a type real variable, and are right-adjusted.
h. Each environmental location is stored in a type real variable and is right-adjusted.
i. Each qualitative identification of a microorganism is stored as sixteen characters.
j. Each "type" variable (e.g., subsections of the routine) is stored in a type real variable and is right-adjusted.
k. Each flight identification is stored in a type real variable and is right-adjusted.
l. The numbers of samples, strips, and type of microbes are a type integer variable.
m. The sampling method is stored in a type real variable and is right-adjusted.
n. The area sampled is a type real floating point number.
o. The location of a sample on an item is stored in a type real variable and is right-adjusted.

These are the specifications for the variables which are universal to the entire system. We shall refer to these in later sections.

2.2 Directory Formats

The system will store, on disks, directories of where information is stored and what relationships exist between various pieces of information. These directories are established by the File Preparation Routine and are updated by other programs when needed. The directories are all of variable length and the entries are consistent with the parameter formats specified in section 2.1. Any memory locations stored in directories (e.g., starting location of various records) are stored in one computer word and are right-adjusted. We shall now enumerate the directories and give the format of each.

2.2.1 Flight List

The flight list will be the catalog of all flights made to the lunar surface in the past or to be made in the future and for which data are available from the Public Health Service at Cape Kennedy. It will be arranged in chronological order. Each entry consists of the flight identification, landing date or predicted landing date, and location on disk of the beginning of the first record of information about the flight.

This directory will contain module identification only under very special circumstances. These are (1) the return to the lunar surface of the ascent stage of the lunar module, and (2) the impact on the lunar surface of the SIVB, SLA, and IU from an Apollo launch vehicle.

2.2.2 Module Directory

The basic purpose served by the module directory is to relate Apollo modules to Apollo flights and to direct the system to where information may be found concerning the microbial load on a module. Only modules which have not completed their flights will be entered in this directory. An entry will consist of module identification, identification of the flight to which the module is assigned, and the location on disk where information concerning this module can be found.

2.2.3 Environmental Location Directory

The environmental location directory will relate environmental sampling locations to buildings where they are located, i.e., in the MSOB, the VABN or VABS, pad 39A or 39B, the suit room, the Apollo lunar surface experiment package (ALSEP) room, or the van. An entry in this directory will consist of the environmental location identification, the building, and the location on disk where the record of information about this sampling location starts.

2.2.4 Astronaut Directory

The astronaut directory is of use only for informational purposes in that it only relates astronauts, flights, and suits and does not give locations of data on disks. The reason for this is that no astronaut (or suit) data will be made available directly over the remote console. It will be reflected only in predictions of sample contamination (see section 4.5.3). An entry in this directory will consist of astronaut identification, the identification of the suit he will use, and the identification of the flight to which he is assigned.

2.2.5 Module-Building Directory

The module-building directory will record the flow of the various pieces of hardware through Cape facilities. An entry will consist of the building identification followed by a list of the modules in that building as of the last update.

2.3 Storage Record Formats

In this section we discuss formats of data records which will be stored, updated, or sorted by the File Preparation Routine. A file is a series of storage records. The parameters appearing in each record will follow the format specifications of section 2.1. Some of these formats will be for tape storage while others will be for the records on the disks.

Before proceeding with the details of each type of record it might be well to note again that each flight will have a tape (or tapes) assigned to it. It would be helpful to have MSOS recognize, if possible, the label as being the flight identification and recognize the environmental tape (or tapes) by the label ENVIRON.

2.3.1 Active Apollo Flight Identification Record

This will be the first record on each Apollo flight tape and will be the first record in the immediate access file on disk for Apollo flights still at the Cape or enroute to the moon. The format will be:

a. Flight identification.
b. Launch date.
c. Landing coordinates.
d. Command module identification.
e. Service module identification.
f. Command and service module assembly identification.
g. Lunar module identification.
h. Lunar module ascent stage identification.
i. Lunar module descent stage identification.
j. Instrument unit identification.
k. SLA identification.
l. SIVB identification.
m. Astronaut 1.
n. Suit assigned to astronaut 1.
o. Astonaut 2.
p. Suit assigned to astronaut 2.
q. Astronaut 3.
r. Suit assigned to astronaut 3.
s. Astronaut 4.
t. Suit assigned to astronaut 4.
u. Astronaut 5.

v. Suit assigned to astronaut 5.

w. Astronaut 6.

x. Suit assigned to astronaut 6.

y. Launch pad identification.

2.3.2 Module Tape Record

A module tape record will be entered onto the appropriate flight tape each time new data concerning a module is read into the information system. The format of this type of record is:

a. Module identification.

b. Date of samples.

c. Sampling method.

d. Physical location of the module, i.e., part of building in which the module is located. This will be one of the environmental sampling locations.

e. Number of samples.

f. Location of sample on module.

g. Area sampled.

h. Number of colonies for the aerobic vegetative sample.

i. Number of colonies for the anaerobic vegetative sample.

j. Number of colonies for the aerobic spore former sample.

k. Number of colonies for the anaerobic spore former sample.

1. Qualitative identification of each colony.

(Repeat items f through 1 for each sample.)

m. Output from the quantitative model which estimates the loading on the date given in item two of this record. This will consist of forty-eight real floating point parameters.

n. Output from the quantitative model which predicts the loading on launch date. This will consist of forty-eight real floating point parameters.

o. Output from the quantitative model which predicts the loading on at-lunar-surface date. This will consist of forty-eight real floating point parameters.

p. Number of kinds of microorganisms as of the date of the samples.

q. A list of the kinds of microbes predicted to be on the module as of the date of the samples.

r. Number of kinds of microbes predicted to be on the module at launch.

s. A list of the kinds of microbes predicted to be present at launch.

t. Number of kinds of microbes predicted to be on the module at the lunar surface.

u. A list of the kinds of microbes predicted to be present at the lunar surface.

v. Predicted probability that all microbes have been identified.

The parameters in items p, r, and t cannot be numerically greater than one hundred and the lists stored in items q, r, and u can contain at most one hundred entries.

2.3.3 Suit Tape Record

A suit tape record will be stored on the appropriate flight tape each time new data concerning a suit is read into the information system. The first three entries in this record will consist of:

a. Suit identification.
b. Date of samples.
c. Type of data.

Item three will be a type real variable which is right-adjusted. There are two types of suit data which can be handled by this system. They are identified (and appear in item three) as SURFACE and LEAKAGE. As of now, the leakage data is not defined. The surface sampling data from the suits will generate a record identical to entries c through v in the module tape record (section 2.3.2).

2.3.4 Astronaut Tape Record

An astronaut tape record will be written in the appropriate flight tape each time new data concerning an astronaut is entered into the system. The first three entries in this record will be:

a. Astronaut identification.
b. Date of samples.
c. Sampling method.

The rest of this record is still undefined. The sampling method will be a type real variable which is right-adjusted. Some of the types which will be used and which must be recognized by the system are: NASAL, THROAT, SKIN, and FECAL. Each type will generate a distinct type of astronaut tape record.

2.3.5 Module Disk Record

The module disk record is the storage format which is used to store the information needed by the Communication Routine (chapter 5) concerning the microbial load on modules. It is also the format which will be used to store on disk the total microbial load information for the entire Apollo spacecraft. The format of this type of record is:

a. Module identification (a flight identification if record is for entire Apollo).
b. Date of the samples on which the predictions are based.
c. Module location.
d. The mean of the distribution of the number of aerobic vegetative cells on the module on the date of the sample.
e. The 90 percent confidence limit of the distribution of the number of aerobic vegetative cells on the module on the date of the samples.
f. The mean of the distribution of the number of aerobic spore formers on the module on the date of the sample.

g. The 90 percent confidence limit of the distribution of the number of aerobic spore formers on the module on the date of the samples.

h. The mean of the distribution of the number of anaerobic vegetative cells on the module on the date of the sample.

i. The 90 percent confidence limit of the distribution of the number of anaerobic vegetative cells on the module on the date of the samples.

j. The mean of the distribution of the number of anaerobic spore formers on the module on the date of the sample.

k. The 90 percent confidence limit of the distribution of the number of anaerobic spore formers on the module on the date of the samples.

l. The mean of the distribution of the number of aerobic vegetative cells on the module at launch.

m. The 90 percent confidence limit of the distribution of the number of aerobic vegetative cells on the module at launch.

n. The mean of the distribution of the number of aerobic spore formers on the module at launch.

o. The 90 percent confidence limit of the distribution of the number of aerobic spore formers on the module at launch.

p. The mean of the distribution of the number of anaerobic vegetative cells on the module at launch.

q. The 90 percent confidence limit of the distribution of the number of anaerobic vegetative cells on the module at launch.

r. The mean of the distribution of the number of anaerobic spore formers on the module at launch.

s. The 90 percent confidence limit of the distribution of the number of anaerobic spore formers on the module at launch.

t. The mean of the distribution of the number of aerobic vegetative cells on the module at lunar surface.

u. The 90 percent confidence limit of the distribution of the number of aerobic vegetative cells on the module at lunar surface.

v. The mean of the distribution of the number of aerobic spore formers on the module at lunar surface.

w. The 90 percent confidence limit of the distribution of the number of aerobic spore formers on the module at lunar surface.

x. The mean of the distribution of the number of anaerobic vegetative cells on the module at lunar surface.

y. The 90 percent confidence limit of the distribution of the number of anaerobic vegetative cells on the module at lunar surface.

z. The mean of the distribution of the number of anaerobic spore formers on the module at lunar surface.

aa. The 90 percent confidence limit of the distribution of the number of anaerobic spore formers on the module at lunar surface.

Entries bb through hh in this record are identical to entries p through v in the module tape record (see section 2.3.2).

2.4 Active Apollo Flight Files

As noted earlier we shall keep two types of Apollo flight files. One will be an archive on tape of all activities performed by the Public Health Service concerning the flight. The other is an immediate access file which will be on disk storage. Both of these files will be composed of records of the types previously defined (see section 2.3). We shall discuss each file independently.

2.4.1 Permanent Apollo Flight File

The permanent Apollo flight file will begin with an active Apollo flight identification record (see section 2.3.1). The remainder of the records in the file will consist of module tape records (2.3.2), suit tape records (2.3.3), and astronaut tape records (2.3.4) in random order. Once a record has been entered into the file it can be removed only by the File Preparation Routine (see section 2.6). This will occur when a module, astronaut, or suit is transferred to another flight.

2.4.2 Immediate Access Apollo Flight File

The immediate access Apollo flight file will provide the information for the Communication Routine (chapter 5). It will contain only the latest information about the flight. The format of this file is:

a. An active Apollo flight identification record.

b. A module disk record for each module and for the total flight.

2.5 Models

Models which develop quantitative microbial burden information can be classified as either estimative or predictive. Estimative models provide an approximation of the microbial burden at the time surface samples are taken. Predictive models provide an approximation of the microbial burden at future times, based on surface and environmental samples taken prior to the dates of the predictions.

Models will be self-contained FORTRAN subroutines. We will discuss here only what information they will need from the main program and what they will return to the main program.

2.5.1 Quantitative Prediction Model

This model has the task of using the output of the quantitative estimation model (see section 3.5.1) to predict the loading at a future date. Access to the following data is needed by this subroutine:

a. Module identification.
b. Date of samples.
c. Outputs from the quantitative estimation model.
d. Environmental tape and environmental disk records.
e. Apollo flight tape.
f. Physical location of the piece of hardware.

The outputs from the subroutine will be (1) ninety-six type real parameters which are to be stored on the Apollo flight tape for the at-launch load and the lunar surface load, and (2) the mean and 90 percent confidence quantities in each of the four categories of data for the at-launch and the lunar surface loads.

2.6 The File Preparation Routine

Figure 1 through 1J present flow diagrams of the File Preparation Routine. The discussion following the figures is devoted to the various details of the program which are not explicit in the figures.

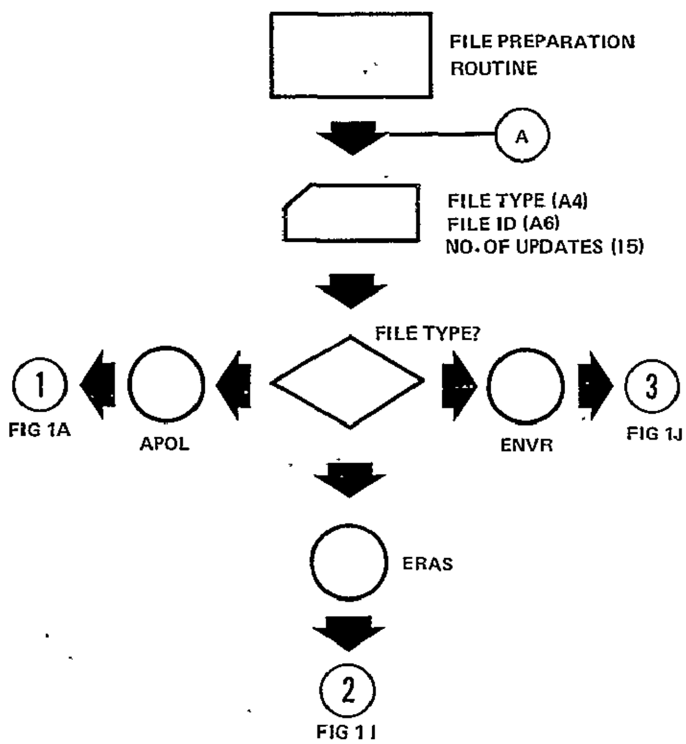


Figure 1. File Preparation Routine

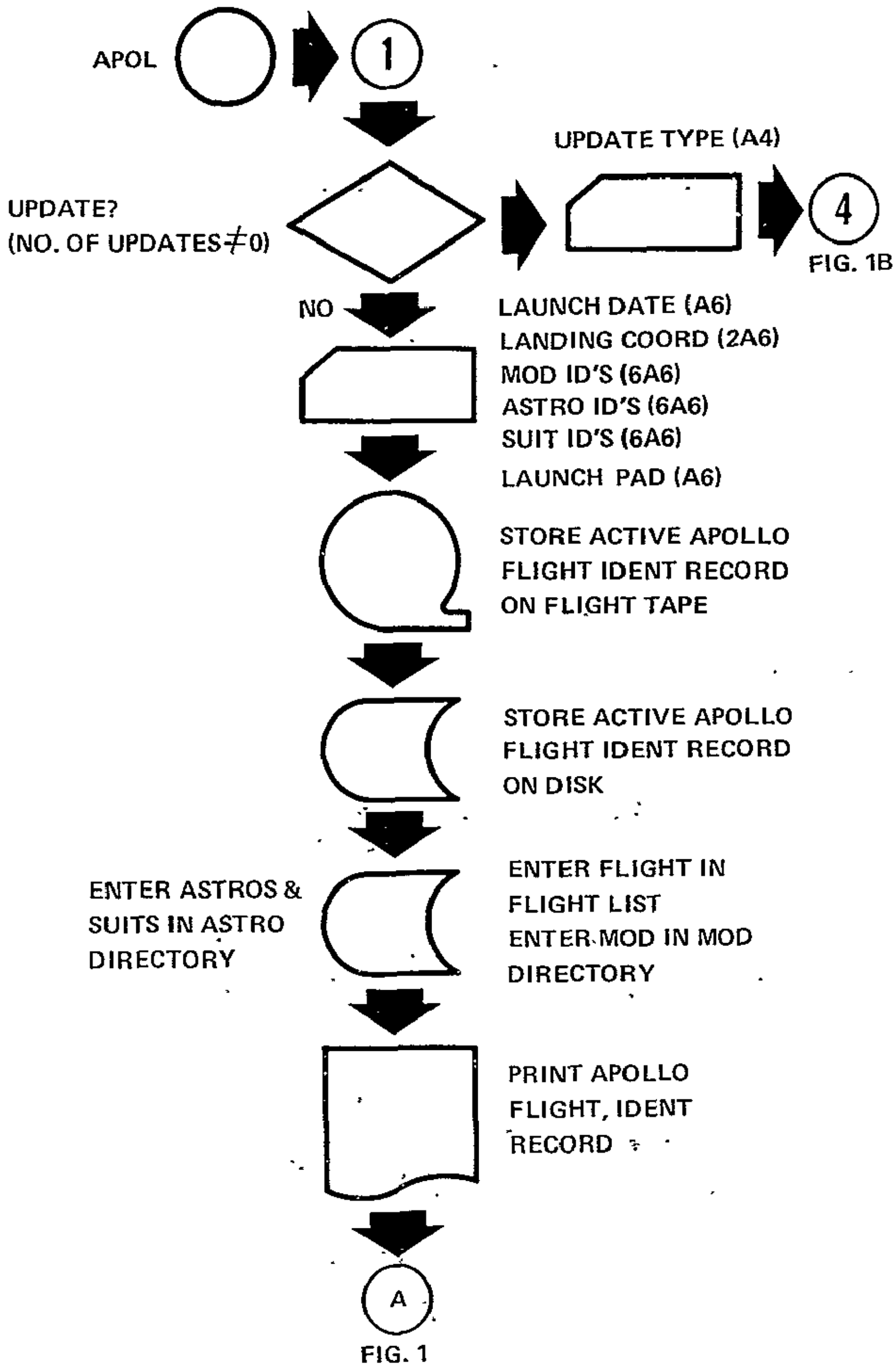


Figure 1A. File Preparation Routine (Continued)

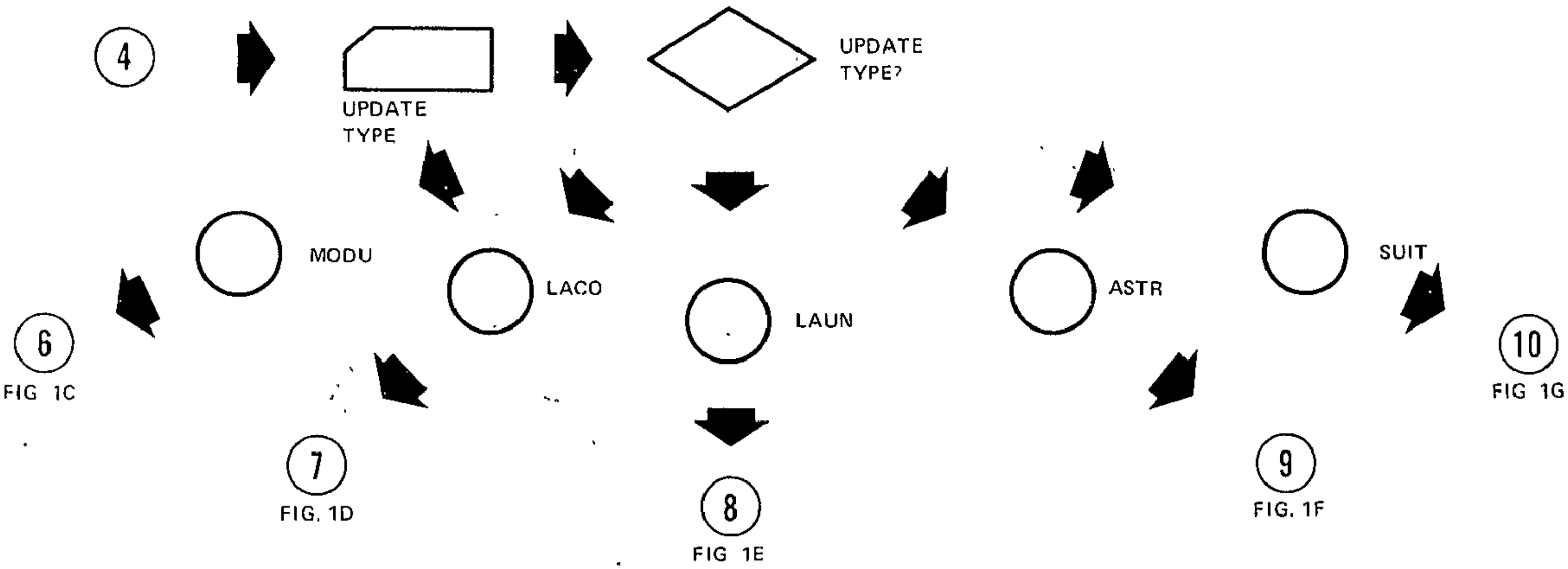


Figure 1B. File Preparation Routine (Continued)

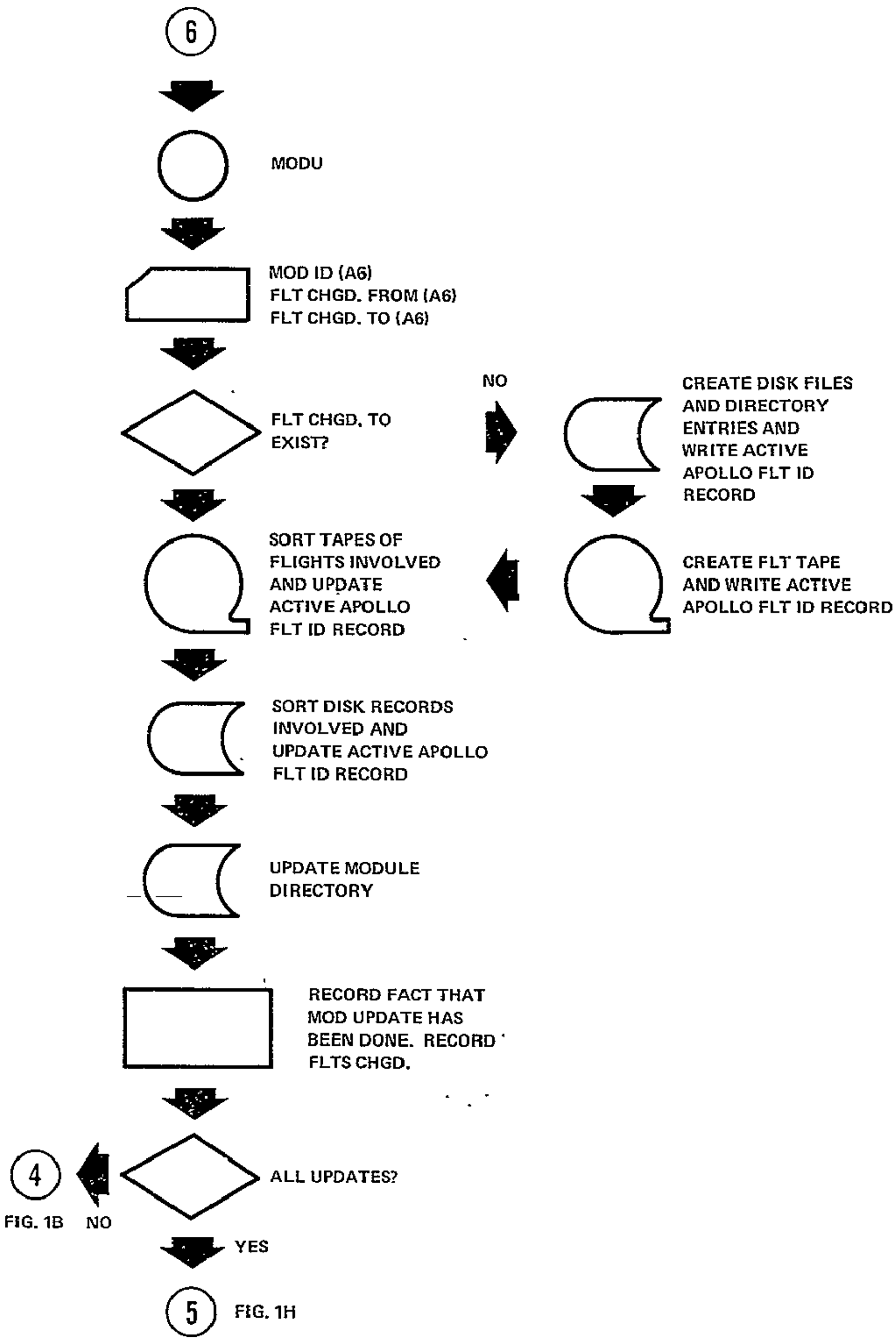


Figure 1C. File Preparation Routine (Continued)

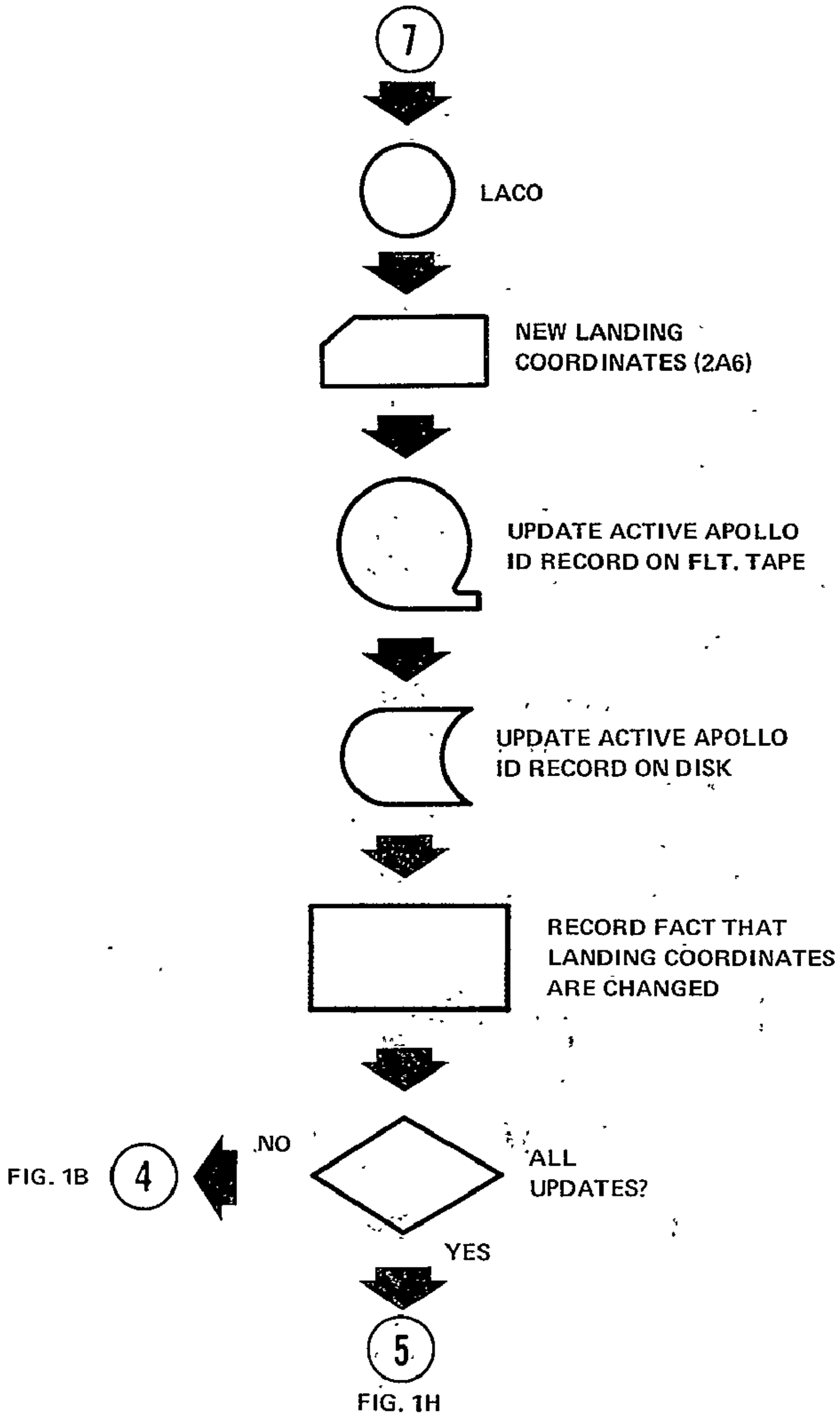


Figure 1D. File Preparation Routine (Continued)

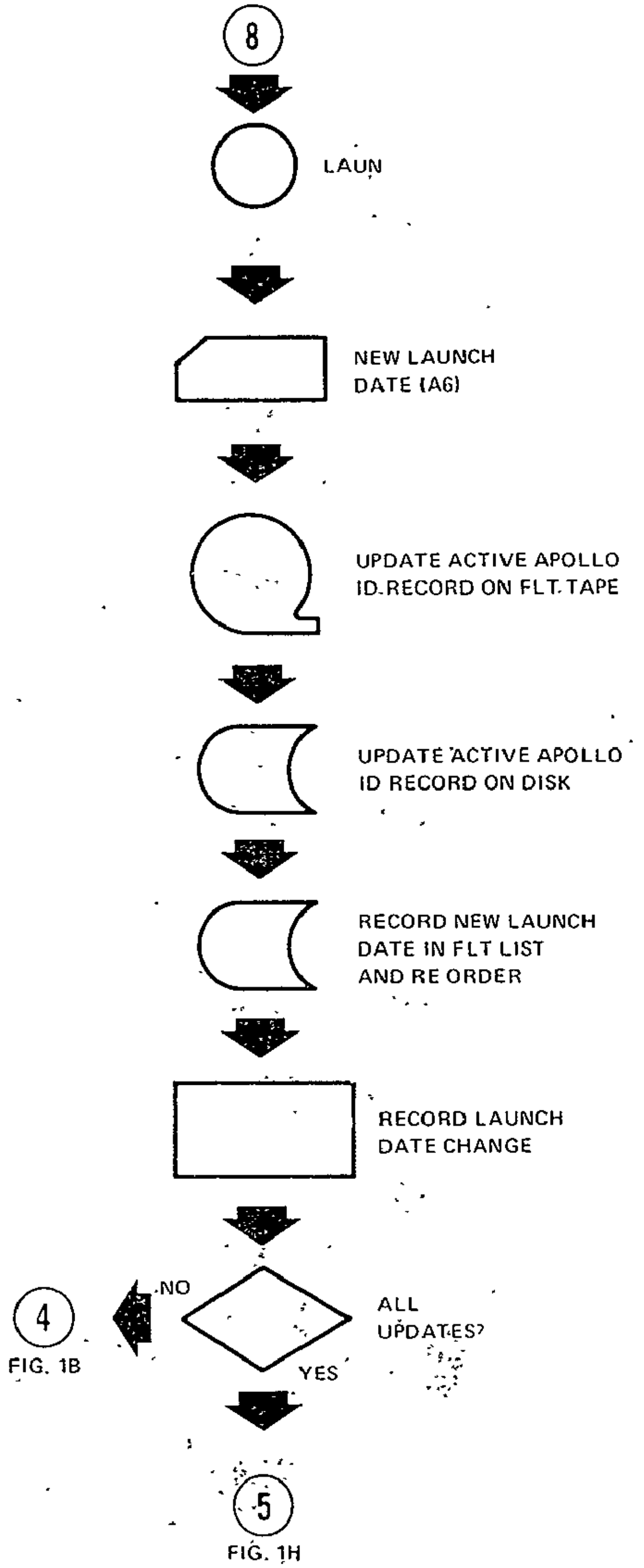


Figure 1E. File Preparation Routine (Continued)

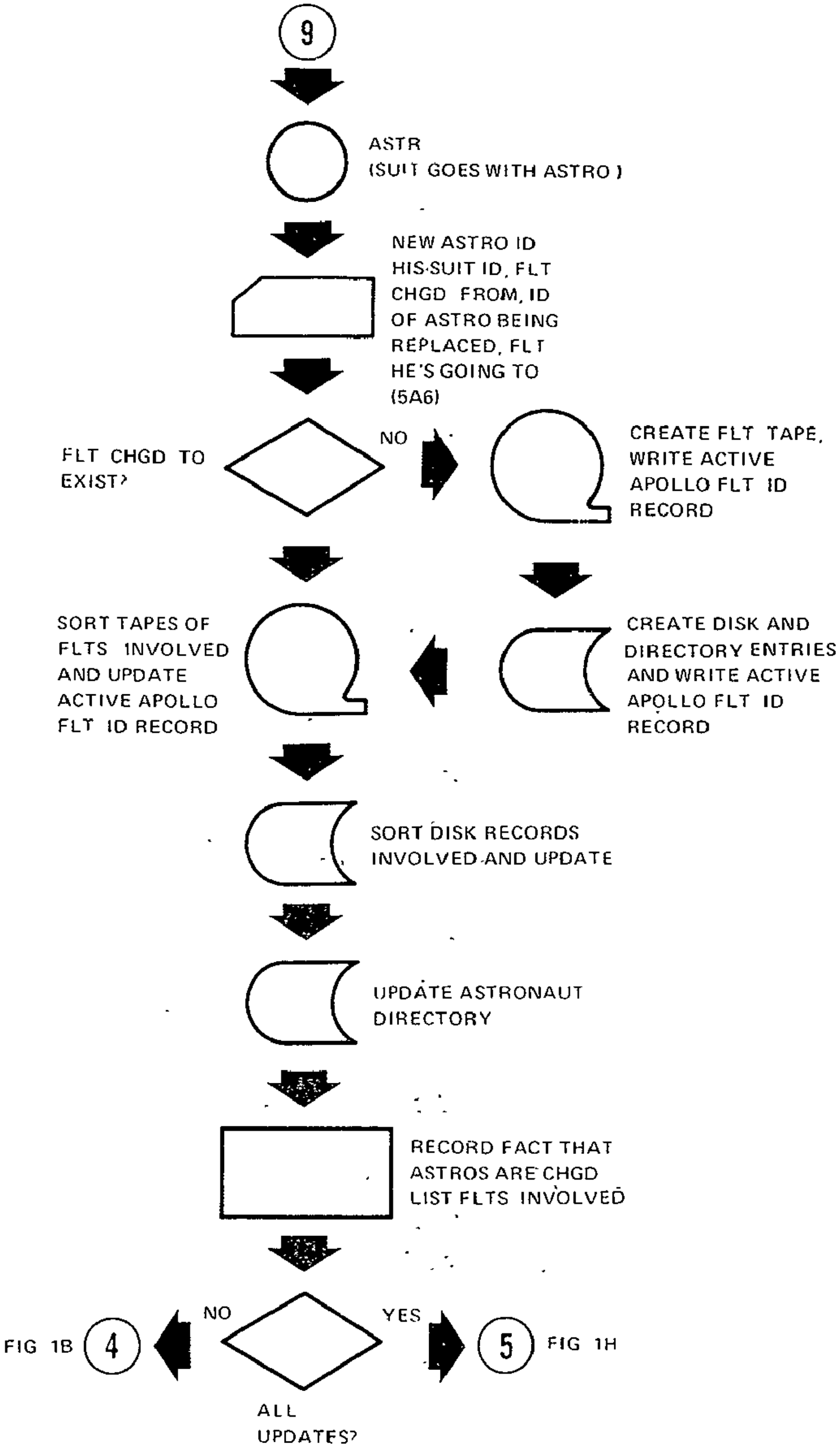


Figure 1F. File Preparation Routine (Continued)

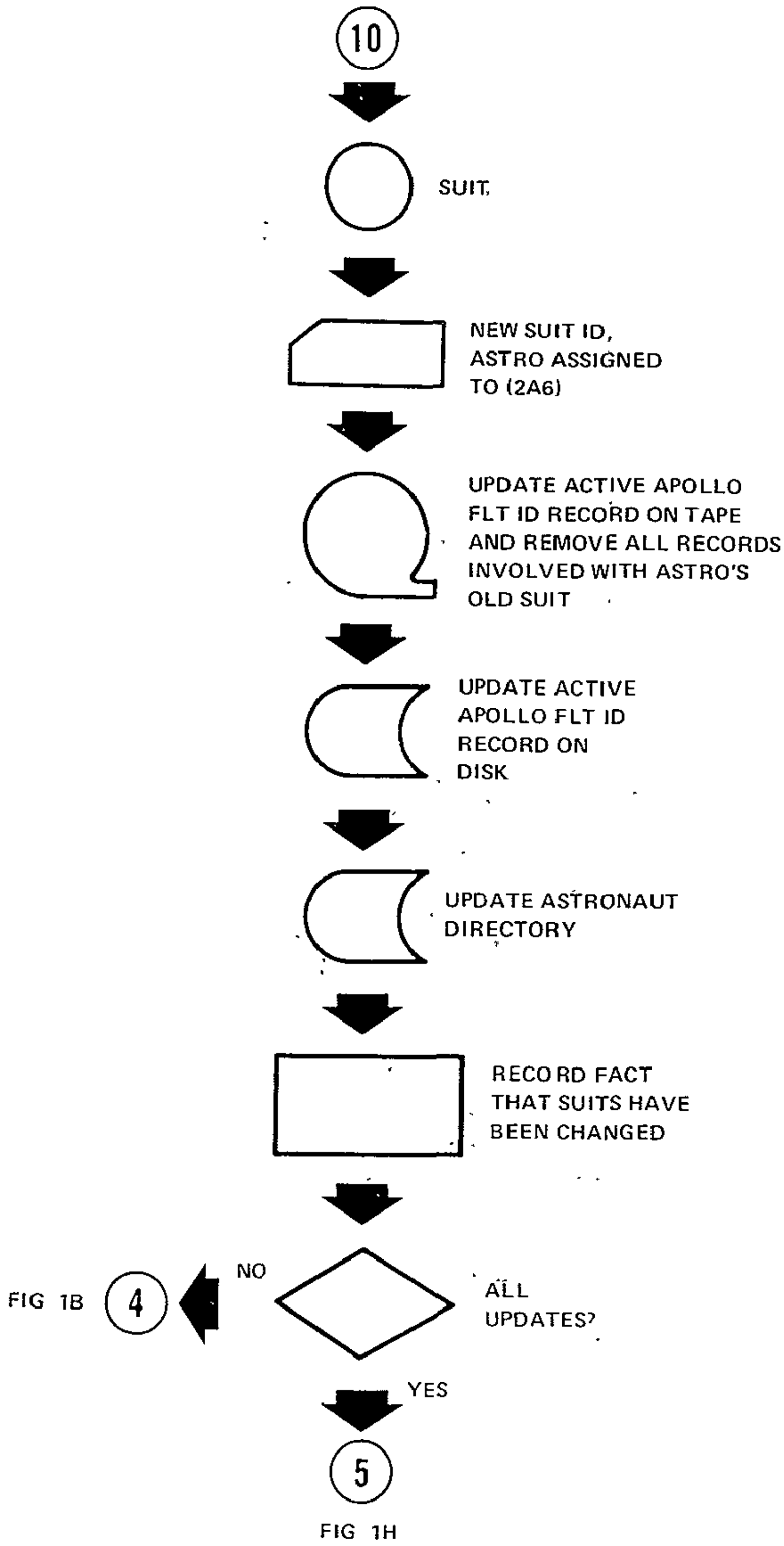


Figure 1G. File Preparation Routine (Continued)

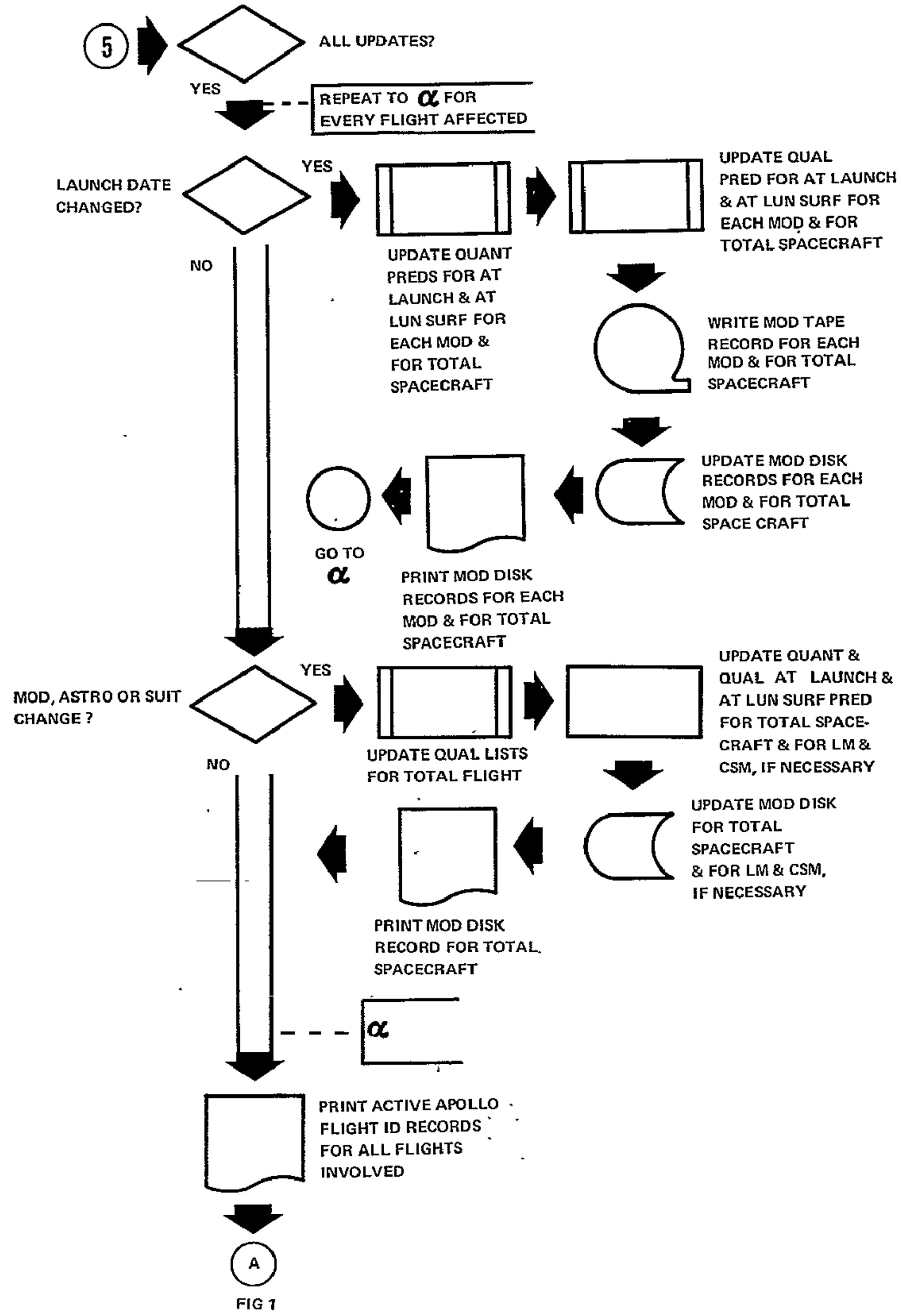


Figure 1H. File Preparation Routine (Continued)

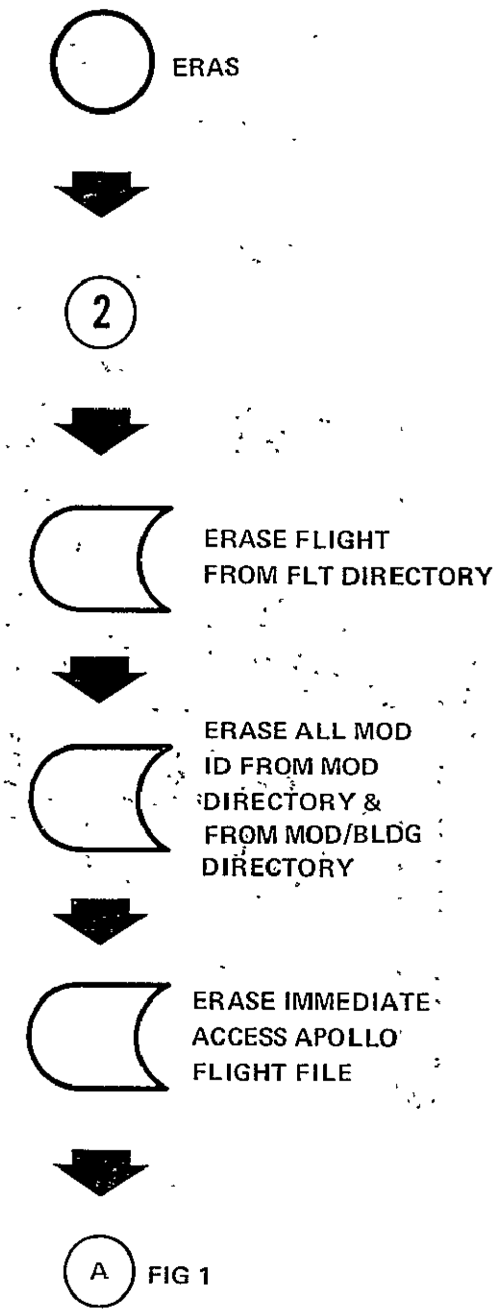


Figure 1I. File Preparation Routine (Continued)

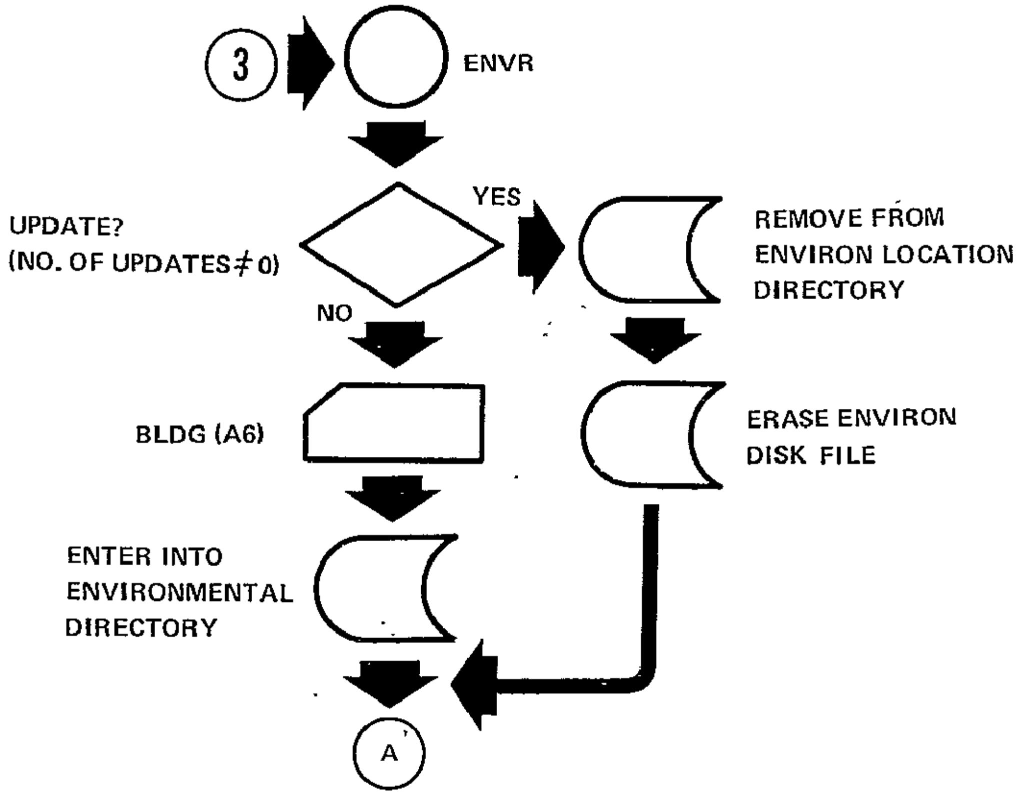


FIG. 1

Figure 1J. File Preparation Routine (Continued)

2.6.1 Details of File Preparation Routine

The File Preparation Routine must have the following capabilities:

a. It must be able to store information concerning module assignments, astronauts, and flight plans for each Apollo flight.

b. It must be able to update flight plans, astronauts, and module assignments when mission changes occur.

c. It must be able to store building identifications and remove these from directories when necessary.

The second of these capabilities is the most time consuming. This is because these pieces of information are stored on tapes and in disk records. Since they influence predictions of biological contamination on the spacecraft at various times, the program must have the ability to sort disk and tape records and to call the FORTRAN subroutines for the various models outlined in section 2.5.

2.6.1.1 The First Card. The first card read in by this program is an identification card which will guide the program to its various subsections. The parameters on this card, their formats, and meanings are:

a. File type (A4). There are three file types which will be recognized by this routine. Their neumonics and functions are:

APOL -- This file type will direct the File Preparation Routine to the section dealing with the active Apollo flights which are either being established for the first time or are being modified through changes to the physical configuration (i.e., modules), astronauts, or flight plans.

ERAS -- This file type will erase an Apollo flight from the flight list and will erase the immediate access Apollo flight file (see section 2.4.2) from the disk.

ENVR -- This file type will establish or erase an environmental sampling location.

We shall discuss each of these in more detail later.

b. File identification (A6). The file identification is one of two types. It is either an Apollo flight identification or environmental sampling location. The first of these is appropriate for either the APOL or ERAS file, while the latter is appropriate with the ENVR file.

c. Number of updates to be performed on this file (I5). The number of updates variable is an integer which tells the number of updates which are to be performed on the file identified by the file identification variable. If this variable is zero it will mean that we are establishing a new file with the name given by the file identification variable.

We are now prepared to discuss each section of the routine individually.

2.6.1.2 Establish an Apollo Flight File. In order to establish an Apollo flight file the first card would read:

File type -- APOL
File identification -- AP XXX
Number of updates -- 0

The remaining cards used in this branch of the routine contain the following information in the indicated formats:

a. Launch date (A6).
b. Landing coordinates (2A6).
c. Command module identification (A6).
d. Service module identification (A6).
e. Lunar module identification (A6).
f. Instrument unit identification (A6).
g. SIVB identification (A6).
h. SLA identification (A6).
i. Astronaut identification (6A6).
j. Suit identification (6A6).
k. Pad identification (A6).

The only additional comment concerning this information is that the nth suit in the list will be assumed to be assigned to the nth astronaut in the list of astronauts for n = 1, 2, 3, 4, 5, or 6. The convention to be used in the numbering of modules is (1) The number assigned to the command module in the command module identification, i.e., CM XXX, will be used as the identification of the command and service module assembly, i.e., CSM XXX; and (2) The number assigned to the lunar module in the lunar module identification, i.e., LM XXX, will be used as the identification of the lunar module ascent and descent stages, i.e., LMA XXX and LMD XXX.

The program is now prepared to output an active Apollo flight identification record (section 2.3.1) on tape, disk, and the on-line printer. It will also enter the flight in the flight list (section 2.2.1).

2.6.1.3 Update of Apollo Flight. Five types of updates can be done on an Apollo flight. The identification of type is read in after the first card and constitutes the first card in an update package. The number of update packages is indicated by the number-of-updates variable on the first card. The neumonics and explanation of the five types of updates are:

a. MODU -- A change has been made in the assignment of one of the modules.
b. LACO -- A change has been made in landing coordinates.
c. ASTR -- A change has been made in the astronauts assigned to the flight.
d. SUIT -- A suit has been changed.
e. LAUN -- A launch date has been changed.

We shall discuss each type of update in an Apollo flight individually. Before doing this we note that at the end of each branch concerning each type of update we record on a list that the update has been performed on that particular Apollo flight (saving the module, astronaut, or suit identification). After all the updates are done, we then perform the updates on the predictions by making use of the various models and writing the appropriate records on tape and disks. See Figure 1H for further information.

2.6.1.4 Module Update. To change a module from one flight to another the program reads a data card containing the following information:

a. New module identification (A6).
b. Flight from which the new module will be transferred (A6).
c. Flight to which the old module of the same type is being transferred (A6).

After reading this card, the program then checks the identification of the flight to which the old module of the same type is being transferred. If this flight identification is blank, the old module and the data associated with it remain as part of the data file to which it currently belongs. If the identification of the flight to which the old module of the same type is to be transferred is not blank, but the identification does not appear in the flight list (see section 2.2.1), then an active Apollo flight identification record (section 2.3.1) is generated for this flight. It contains all nines in every word of the record with the exception of the flight identification which is as specified. We now enter this record into a permanent Apollo flight file (section 2.4.1) and into an immediate access Apollo flight file (section 2.4.2). We enter this flight into the flight list and continue as if the flight were identified in the beginning.

One point should be emphasized. We allow a flight tape to contain records regarding more than one module of the same type. This multiple assignment is reflected in the module directory (section 2.2.2) and will be useful in cataloging unassigned modules. The active Apollo flight identification record will contain only one module identification for each type of module. This is the one (if it exists) which is assigned to the flight.

The program then sorts the disks and tape files involved. This means that it sorts the files for the flight from which the module is being transferred and rewrites all the records on the files concerning the flight to which the module is being transferred. It then records the fact that a module update has been performed on the flights affected and changes the module identification in the appropriate active Apollo flight identification records on tape and disk.

2.6.1.5 Astronaut Update. The procedure to update an astronaut is the same as that to update a module except that we must specify which astronaut he is replacing. The convention to be followed here is that an astronaut's suit always is transferred with him. Thus, we change both the astronaut and suit. The format of the card containing the update information is 6A6.

The fact that an astronaut update has been performed on the flights involved is then recorded and an update of the active Apollo flight identification records and the astronaut suit directory is performed.

2.6.1.6 Suit Update. The procedure to update a suit is also the same as updating a module except that the old suit is simply removed from the file rather than transferred to another flight. At the end of this branch, the fact that a suit update has been performed on the flight involved is recorded. We also update the appropriate active Apollo flight identification record and the astronaut suit directory. The format of the card containing the update information is 2A6.

2.6.1.7 Landing Coordinates or Launch Date Update. Only three things are done when either type of update is specified:

a. Update the active Apollo flight identification records on tape and disk.
b. Update the flight list directory (in the case of launch date change).
c. Record on the list the fact that a launch date or landing coordinates have been changed.

The card specifying a landing coordinate update has format 2A6 and the card specifying a launch date update has format A6.

2.6.1.8 Update of Predictions. After all storage files have been updated for the flight modifications that have been performed, the program then jumps to a section (see Figure 1H) which goes through the list of updates performed and modifies the predictions of the qualitative and quantitative bioburdens that require modification for the flight involved and stores the modified predictions in the appropriate records. The last function of this routine is to print the active Apollo flight identification record for each flight which has been modified.

2.6.1.9 Erase Flight from Flight Directory. The file type identification ERAS causes the File Preparation Routine to jump to a section (see Figure 1I) which removes the specified flight from the flight directory and erases the immediate access Apollo flight file (see section 2.4.2) from the disk. This will be used when a flight has been discontinued. The first card in this program would read:

File type -- ERAS
File identification -- AP XXX
Number of updates -- (blank)

2.6.1.10 Establish an Environmental Sampling Location. In order to establish an environmental sampling location the first card would read:

File type -- ENVR
File identification --XXXXXX
Number of updates -- 0

The only other data card needed has format A6 and contains the identification of the building where the sampling location is. This file identification and building would then be entered into the environmental directory and the program would request more data.

2.6.1.11 Update an Environmental Sampling Location. The only type of update which we foresee being performed on an environmental sampling location would be due to its inactivation. In this event the first card would read:

File type -- ENVR

File identification -- XXXXXX

Number of updates -- 1

The program would then remove this location from the environmental directory and erase the environmental disk record.

CHAPTER 3. DATA STORAGE ROUTINE

The Data Storage Routine is that part of the information system which stores (1) information concerning the Apollo flights that have not reached the lunar surface and (2) environmental information concerning the various buildings and areas at Cape Kennedy. This routine assures that the File Preparation Routine has already been used and that files and directory entries have been established for the particular module, flight, suit, astronaut, or environmental location under consideration.

In discussing this routine we shall follow the same format used in our discussion of the File Preparation Routine in chapter 2.

3.1 Parameter Formats

The parameter formats outlined in section 2.1 will hold for this routine.

3.2 Directory Formats

The directory formats outlined in section 2.2 will hold for this routine.

3.3 Storage Record Formats

The storage record formats given in section 2.3 will be used for the storage of data concerning Apollo flights. In addition, some tape and disk records will be needed for the storage of environmental information. The remainder of this section will be devoted to a discussion of these.

3.3.1 Environmental Tape Record

An environmental tape record will be stored on the environmental tape each time new data concerning an environmental location is read into the information system. The first three entries in this type of record will consist of:

a. Environmental location identification.
b. Date of samples.
c. Type of sample.

Item c will be a type real variable which is right-adjusted. There will be four types of samples which can be entered into the system. They are STRP, AIRS, AGAR, and ANDS. Each of these types will generate a different format for the remainder of the environmental tape record. As of now, the formats of the records generated by the AGAR (agar plate) and ANDS (Andersen) types of sample are not defined. We shall discuss the other two in detail.

3.3.1.1 STRP Sample. The STRP type of sample represents those data gathered on stainless steel fallout strips. The environmental tape record generated by this type of sample is:

a. Environmental location identification.
b. Date of samples.
c. Type of sample.
d. Number of strips.
e. Length of exposure.
f. Number of colonies for aerobic vegetative sample.
g. Number of colonies for anaerobic vegetative sample.
h. Number of colonies for aerobic spore former sample.
i. Number of colonies for anaerobic spore former sample.
j. Qualitative identification of each colony.

(Repeat items e through j for each strip)

k. Predicted fallout rate. There will be four parameters here, one for each category of microbe.
l. Predicted removal rate. There will be four parameters here, one for each category of microbe.

3.3.1.2 AIRS Sample. The AIRS type of sample represents those data gathered by Reynier or Casella air samplers. The environmental tape record generated by this type of sample is:

a. Environmental location identification.
b. Date of samples.
c. Type of sample.
d. Number of samples.
e. Type of sampler.
f. Length of time sampler used.
g. Number of colonies.
h. Qualitative identification of each colony.
i. Number of particles per cubic foot of air at environmental location.

3.3.2 Environmental Disk Record

There will be an environmental disk record for each sampling location. The format of this record is:

a. Environmental location identification.
b. Date of last update.
c. Predicted fallout rate.
d. Predicted removal rate.
e. Number of particles per cubic foot of air at this location.
f. Number of kinds of microbes found in area.
g. Qualitative identification of each kind of microbe.

3.4 Environmental Files

As noted earlier two kinds of environmental files will be kept. The first of these is a permanent file on tape. It is composed of environmental tape records for all of the sampling locations in random order.

The other type of file is an immediate access file on disk storage. There shall be one file for each building. It is composed of an environmental disk record for every sampling location.

3.5 Models

In this section we will discuss the estimative and predictive models needed in the Data Storage Routine. In addition to the models described in section 2.5 there are several others which will be needed. Again, they will be self-contained FORTRAN subroutines. We will discuss here what information they will need from the main program and what will be returned to the main program.

3.5.1 Quantitative Estimation Model

The quantitative estimation model uses the quantitative surface sampling data to estimate the total quantitative microbial load on the piece of hardware on the date the sample was taken. Access to the following data is needed by this routine.

a. Module or suit identification.
b. Date of samples.
c. Number of colonies found in each of the four categories in each sample.
d. Apollo flight tape.

The output from this routine is the forty-eight type real parameters which are to be stored on the Apollo flight tape for the load on the date of the sample, and the mean and 90 percent confidence quantities in each of the four categories for the load on the date of the sample.

3.5.2 Qualitative Identification Model

The details of the qualitative identification model[2] are in development at this time. This subroutine will identify qualitatively the microbes on the hardware. The input to the routine will be the results of a series of qualitative tests stored as an octal constant. The output will be an alphanumeric identification of the organism.

3.5.3 Qualitative Estimation Model

The qualitative estimation model is one of the few models which probably will not be a subroutine since it simply compiles a list of organisms which have been found in the surface samples taken from the hardware. This routine will combine the list which is already in the module disk record with those microorganisms which have just been found.

3.5.4 Qualitative Prediction Model

The qualitative prediction model uses the output of the qualitative surface sampling model (section 3.5.3) to predict the qualitative microbial load on the piece of hardware at a future date. Access to the following data is needed in this routine:

a. Module identification.

b. Date of samples.

c. Output from the qualitative estimation model.

d. Environmental tape and environmental disk records.

e. Physical location of the piece of hardware.

f. Apollo flight tape.

The output from this routine is the number of kinds of microbes predicted to be on the hardware at launch and at the lunar surface, together with a list of the qualitative identifications of each load.

3.5.5 Probability-of-Identifying-All Model

The probability-of-identifying-all model gives the probability that all microbes on the hardware have been identified. It needs access to the Apollo flight tape, and returns one real floating point parameter to the main program.

3.6 Data Storage Routine

Figures 2 through 2I present flow diagrams of the Data Storage Routine. The discussion following the figures covers some important aspects of the program.

3.6.1 Details of the Data Storage Routine

In order to fulfill its purposes, the Data-Storage Routine must have the following capabilities:

a. It must process and store swab and vacuum probe data taken from the surfaces of various modules and from suits.

b. It must process and store environmental information gathered with various types of air samplers, stainless steel fallout strips, and agar fallout plates. (This last type is included because it provides a way in which certain parameters needed in the predictive models can be obtained.)

c. It must process and store various types (as yet undefined) of data taken from astronauts.

d. It must process and store suit leakage data (as yet undefined).

e. It must call various subroutines which will perform calculations of predicted microbial loads on the various pieces of hardware and on the lunar surface.

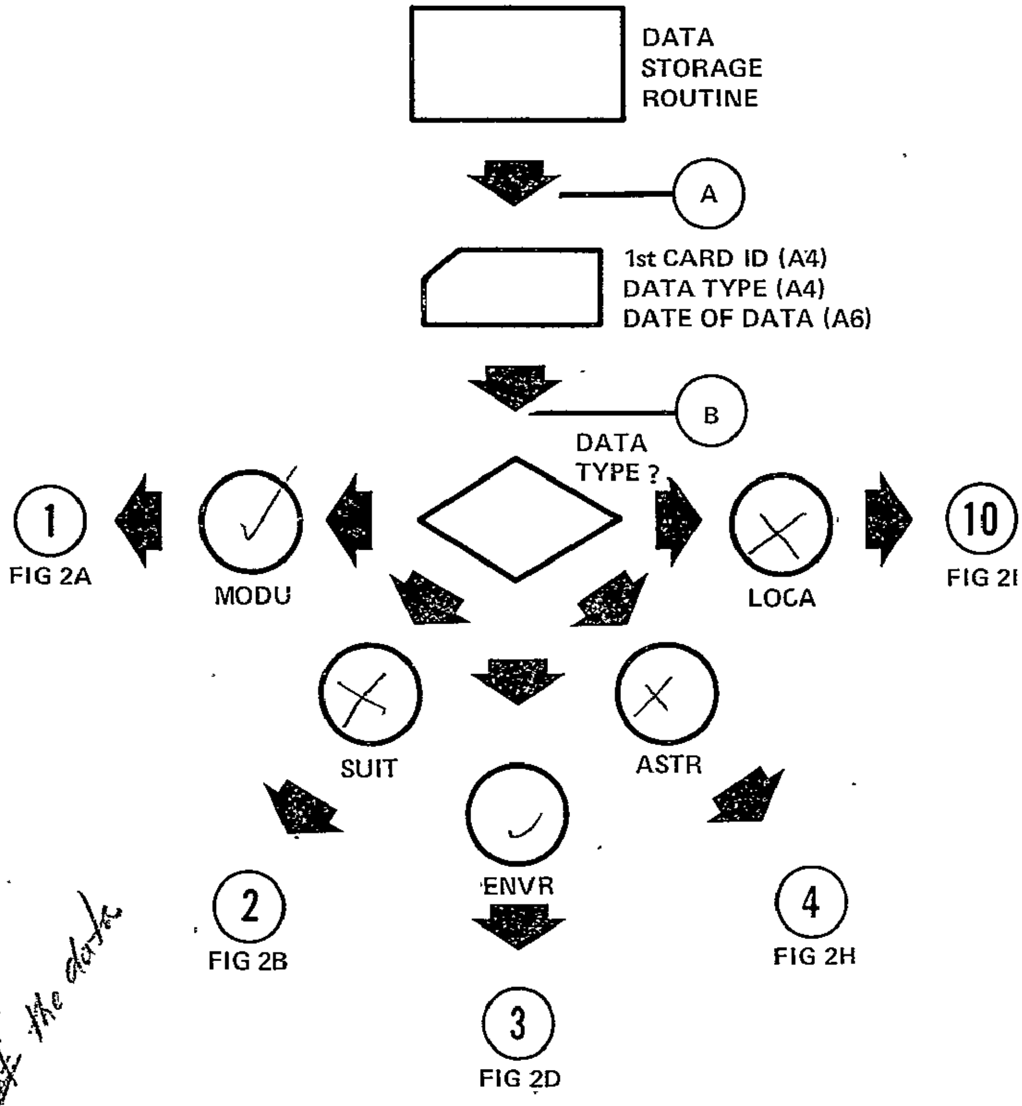


Figure 2. Data Storage Routine

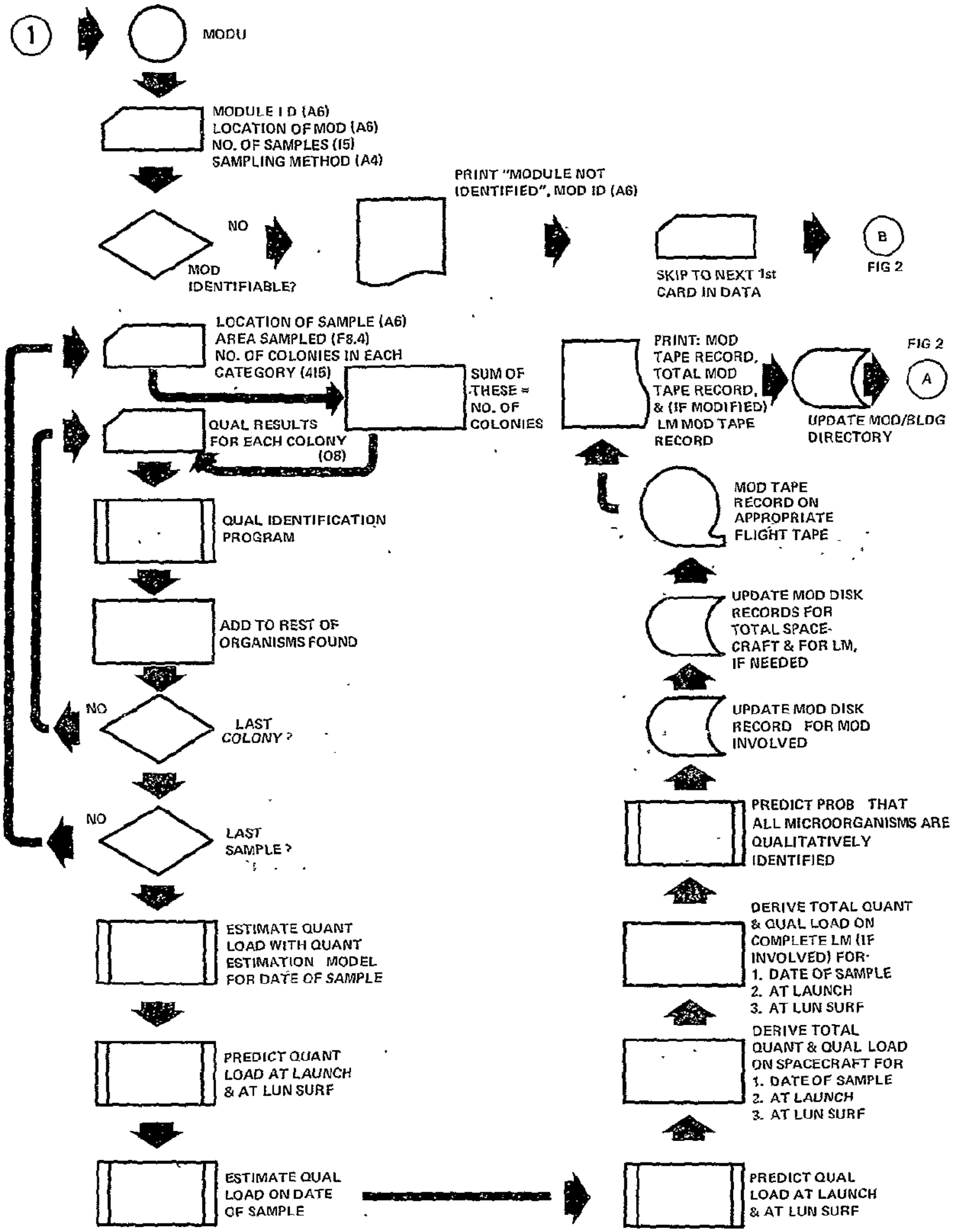


Figure 2A. Data Storage Routine (Continued)

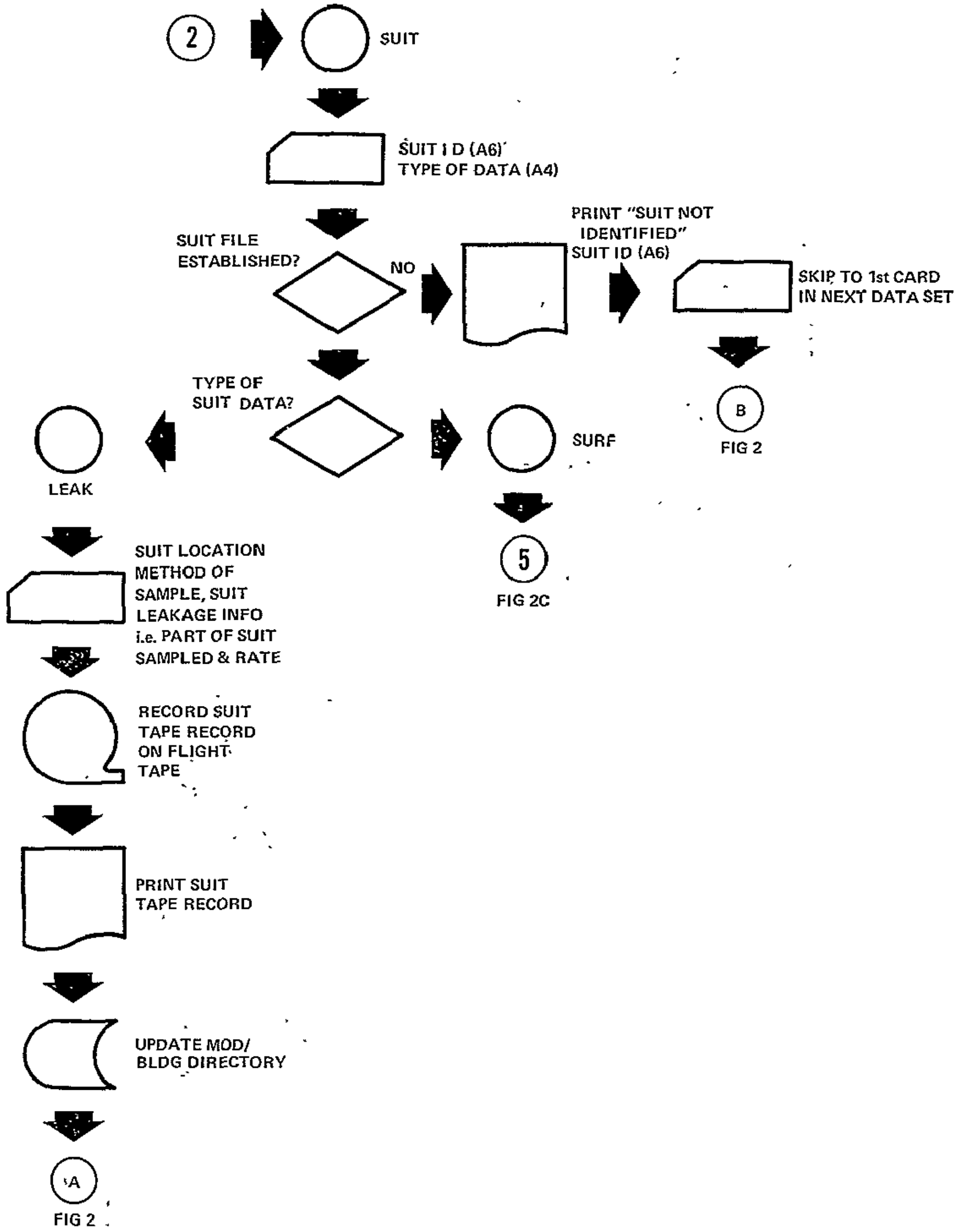


Figure 2B. Data Storage Routine (Continued)

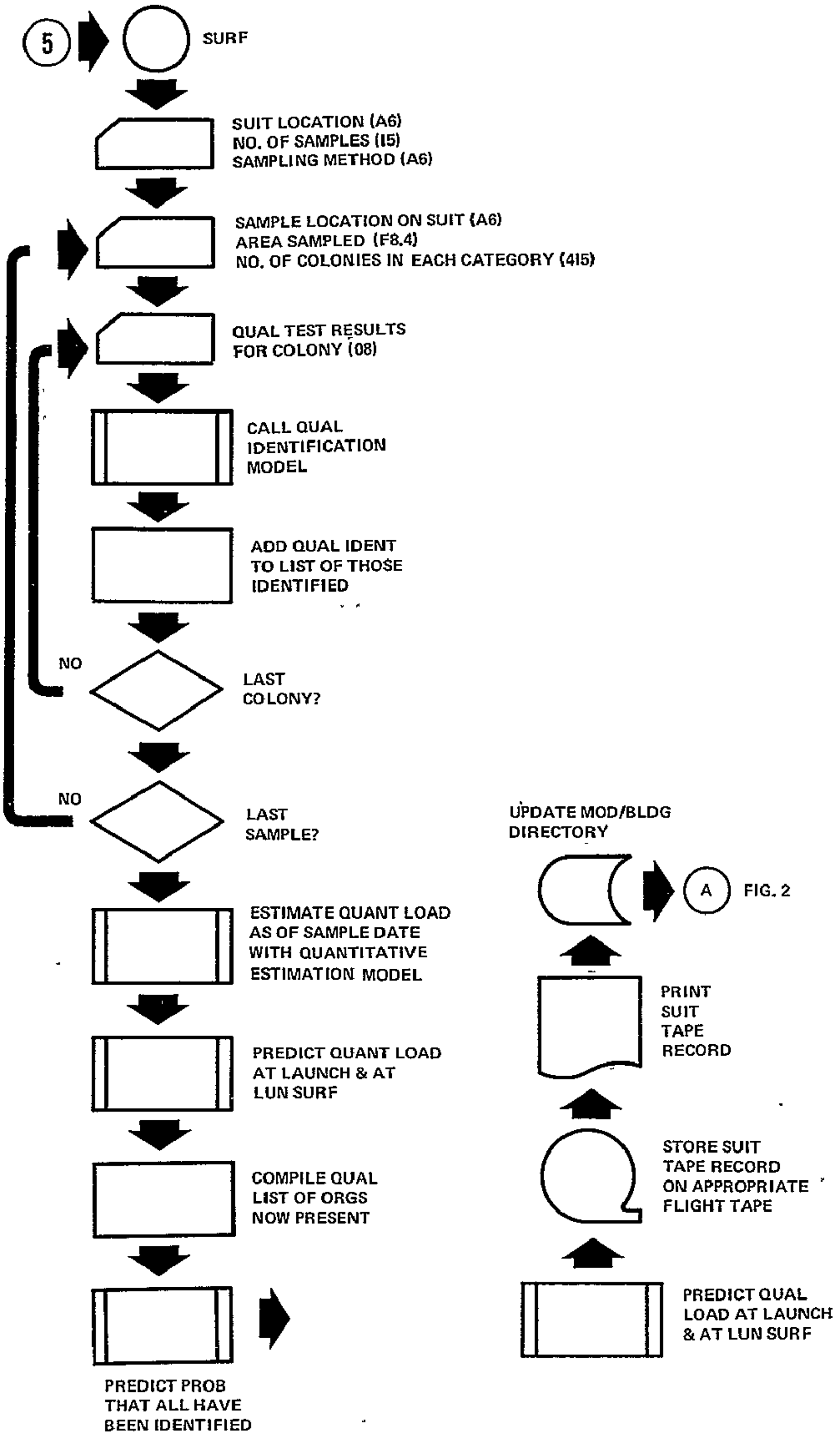


Figure 2C. Data Storage Routine (Continued)

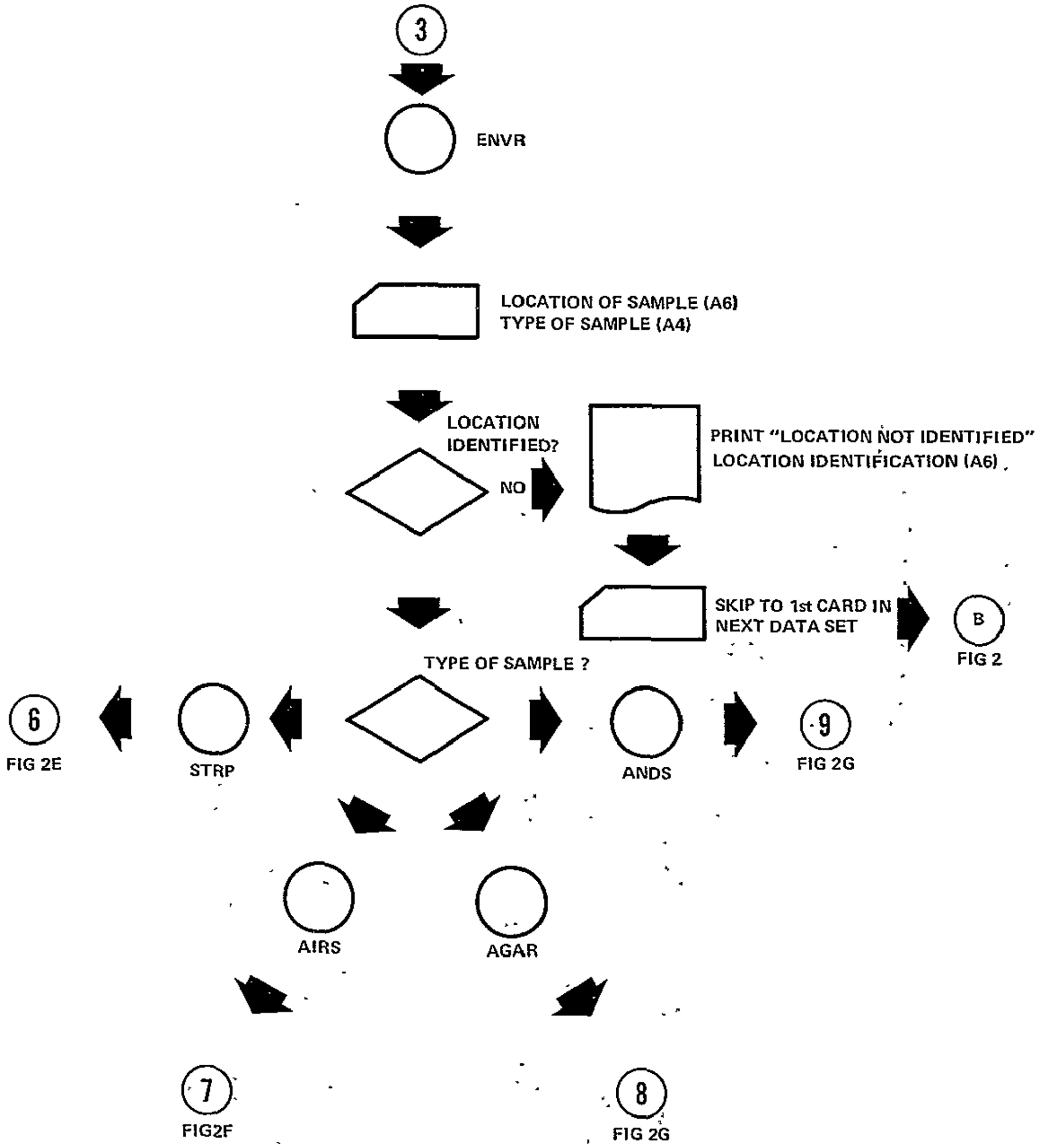


Figure 2D. Data Storage Routine (Continued)

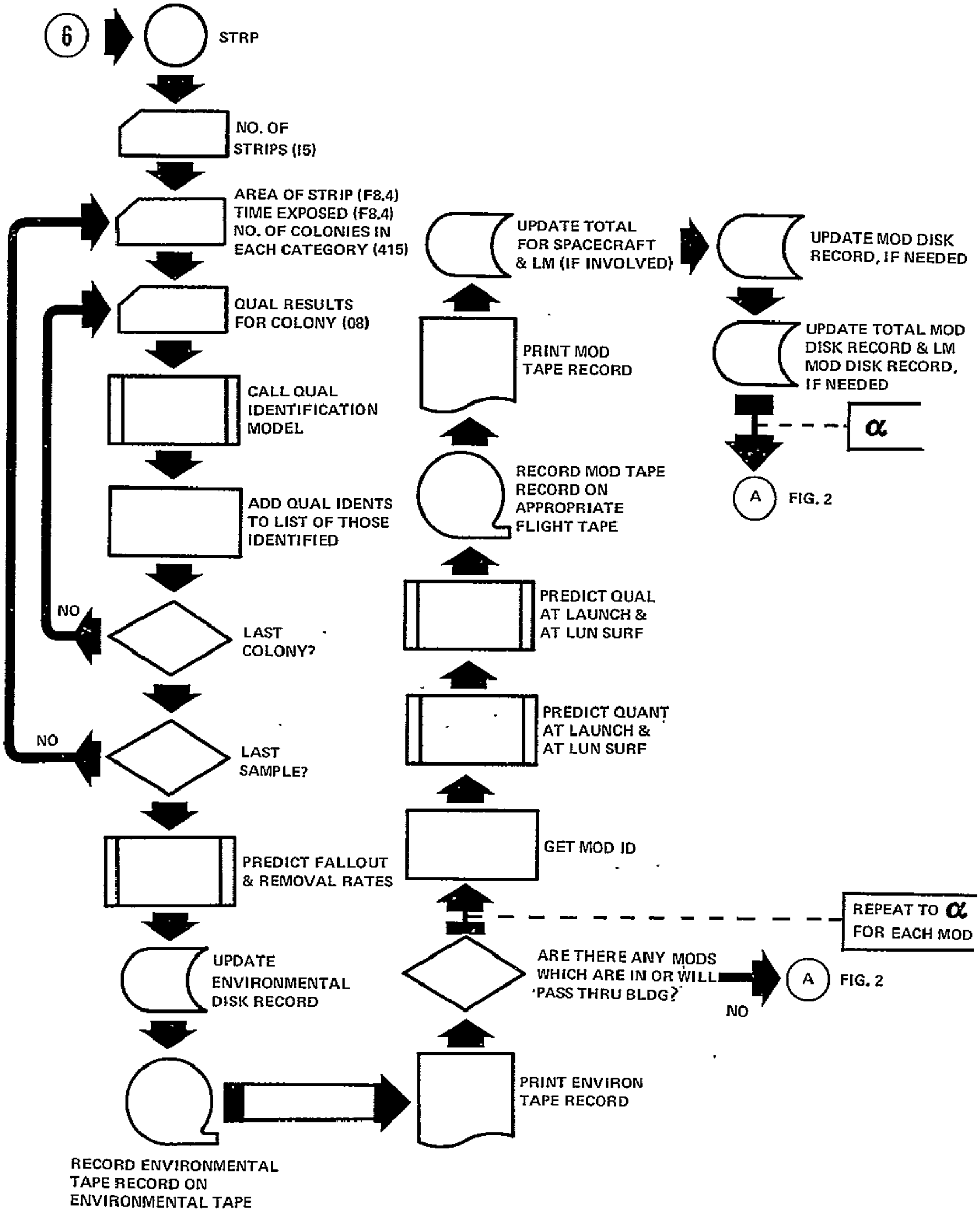
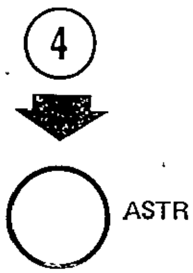


Figure 2E. Data Storage Routine (Continued)

THE DATA TO BE GATHERED BY THESE TWO METHODS WILL BE USED, PRIMARILY, FOR DETERMINING ASSAY MODEL PARAMETERS. THESE EXPERIMENTS ARE NOT YET DEFINED. THE DATA WILL BE STORED ROUTINELY, HOWEVER, TO BE AVAILABLE FOR ANY OTHER USES WHICH MAY DEVELOP.

Figure 2G. Data Storage Routine (Continued)



THE CONTENT, FORMAT AND SOURCES OF THIS DATA ARE NOT YET DEFINED. ITS AVAILABILITY, ETC. ARE AWAITING THE ESTABLISHMENT OF AGREEMENTS AND PROTOCOLS BETWEEN THE PLANETARY QUARANTINE OFFICE AND THE MANNED SPACECRAFT CENTER, HOUSTON.

Figure 2H. Data Storage Routine (Continued)

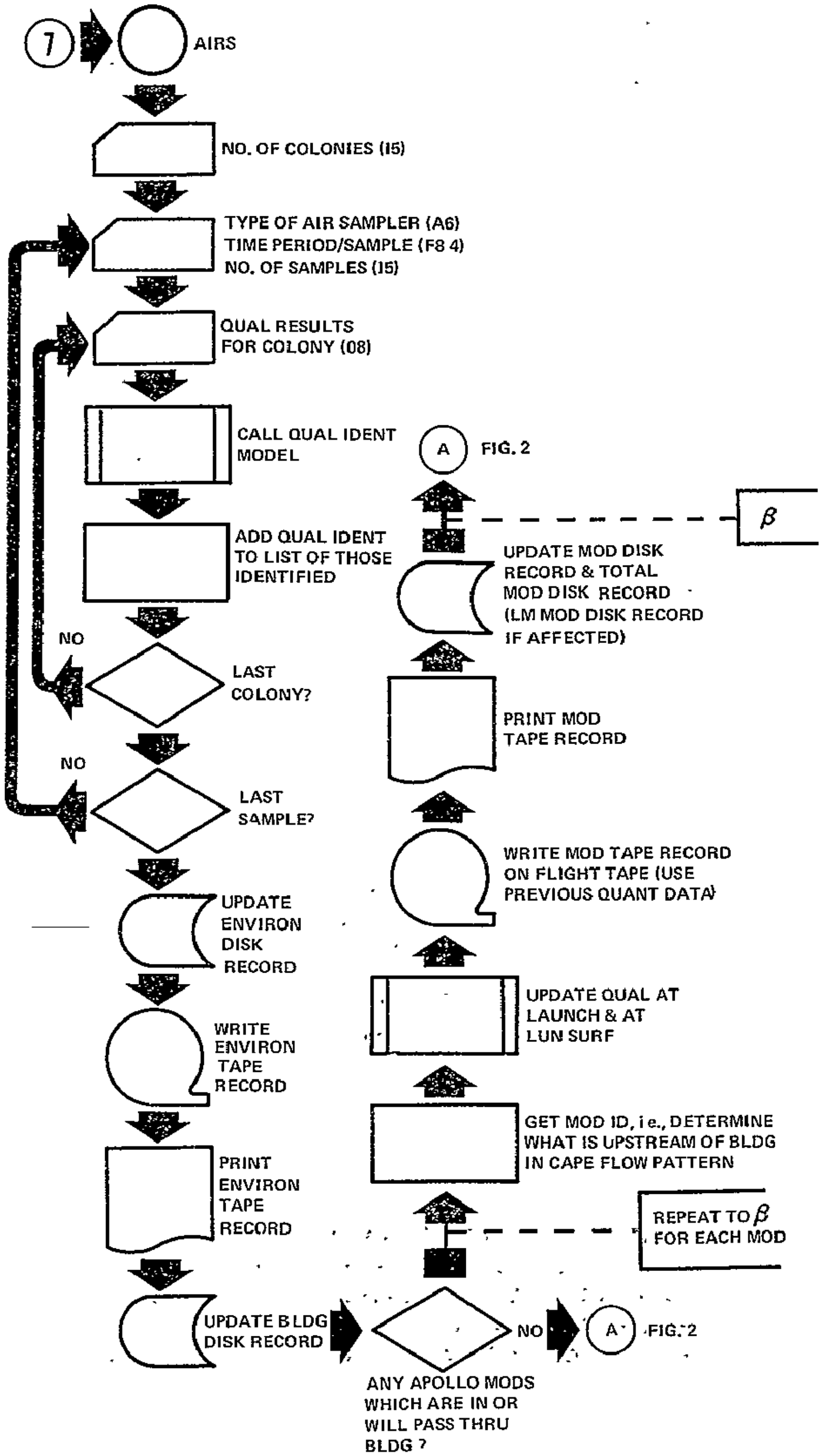


Figure 2F. Data Storage Routine (Continued)

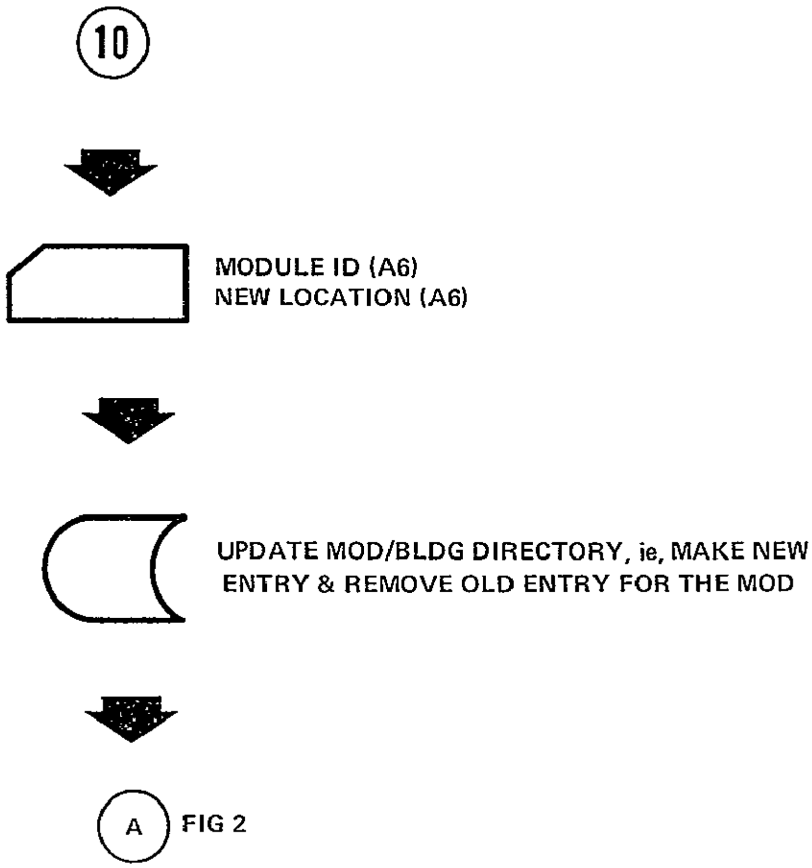


Figure 2I. Data Storage Routine (Continued)

The first card read in by this program is an identification card which will guide the program to its various subsections. The parameters on this card and their formats and meanings follow.

3.6.1.1 First Card Identification (A4). The first card identifies the start of a data deck. We suggest using "9999" or some similar combination of variables.

3.6.1.2 Data Type (A4). The data type variable directs the program to the appropriate subsection. The five available codes and their meanings are:

a. MODU -- Module Data. Data concerning surface samples taken from the pieces of hardware associated with an Apollo flight are to be entered into the system. This section of this program, diagrammed in Figure 2A, handles the storage and processing of data from swab or probe samples taken from an Apollo module. Since all the necessary details are given in the flow chart, we feel that no further discussion of this section of the program is needed, except for a description of the sampling method variable. This variable specifies what method was used to take the samples of microbial contamination on the surface of the specified module. Two types are allowed: (1) SWAB, the conventional cotton swab; and (2) PROB, the vacuum probe.[4] The sampling method determines the distribution of sampling error used in the quantitative estimation model (see section 3.5.1).

b. SUIT -- Suit Data. Data concerning a suit to be used by an Apollo astronaut are to be entered into the system. This section of the program, to process and store the data concerning an astronaut's suit, is diagrammed in Figures 2B and 2C. Two types of data are allowed: (1) LEAK, the code used to store and process data on the leakage rates for a suit; as yet the form of this type of data is not defined. And (2) SURF, the code used when data gathered by surface sampling are to be fed into the system; the cards following have the same form as those for module data. The type being entered is specified by the type of data variable.

c. LOCA -- Data concerning the new physical location of a module are to be read into the system.

d. ENVR -- Environmental Data. Data concerning samples taken at an environmental location are to be entered into the system. This subsection of the Data Storage Routine stores and processes data from samples taken at environmental locations identified in the File Preparation Routine. Figure 2D is the flow chart for this program. We allow the four types of samples defined in section 3.3.1. The details of these are given in Figures 2E to 2G.

We must point out two important things about our planning of this section of the routine: (1) The exact form of and the calculations to be performed on data taken by agar plates and by Andersen samplers are not, as of now, defined; and (2) Under the type of sample designated AIRS, we allow two types of air samplers. These are the Reynier air sampler (REYN), and the Casella air sampler (CASE).

e. ASTR -- Astronaut Data. Data concerning the bioflora of an astronaut are to be entered into the system. As of now, the data which will be available on astronaut bioflora are so vague that no attempt has been made to discuss them in this document.

3.6.1.3 Date of Data (A6). The date of data variable contains the month, day, and year on which the samples were taken about which data are being entered into the system.

CHAPTER 4. LUNAR INVENTORY ROUTINE

The Planetary Quarantine Officer is responsible for maintaining an inventory of the microbial loads taken to the lunar surface by manned and automated spacecraft. The Lunar Inventory Routine establishes and updates a file on disk which contains this information.

In this chapter we shall follow the same format used in Chapters 2 and 3.

4.1 Parameter Formats

The same parameter formats outlined in section 2.1 apply to this routine. The only new parameters needed are the type of impact and whether an Apollo flight was successful. Each is a type real variable which is right-adjusted.

4.2 Directory Formats

The same directory formats outlined in section 2.2 apply to this routine.

4.3 Storage Record Formats

The information needed in transferring an active Apollo flight to the lunar inventory (see section 4.6) makes use of the storage record formats described in section 2.3. There are three types of records which can be written on disk by this routine. We shall discuss each.

4.3.1 Standard Lunar Inventory Record

The majority of the records written by the Lunar Inventory Routine are standard lunar inventory records. These contain only the latest updates of information. The format of this record is:

a. Flight identification. This may be simply the identification for a lunar module ascent stage or an SIVB (see section 4.6 for details).
b. Date of last update.
c. Launch date.
d. Landing date.
e. Landing coordinates.
f. Predicted quantitative microbial load at launch.
g. Predicted quantitative microbial load reaching the lunar surface.
h. Predicted quantitative microbial load still alive as of last update, i.e., as of the date in item b.

4.3.2 Sample Contamination Record

The sample contamination record stores the information relating to the predicted contamination of samples to be collected on the next Apollo flight. The format of this record is:

a. Next Apollo flight identification.
b. Date of last update.
c. Landing coordinates.
d. Number of microorganisms predicted to be present at landing site.
e. Probability of sample contamination.

4.3.3 Total Lunar Contamination Record

The total lunar contamination record contains the information concerning the total number of microbes predicted to be alive on the lunar surface. This record contains only the date on which the predictions were made and the predicted number of microbes still alive as of that date.

4.4 Lunar Inventory File

The lunar inventory file is stored on disk and is available for immediate access by the Communication Routine. It consists of one total lunar contamination record, one sample contamination record, and one standard lunar inventory record for each piece of hardware which has reached the lunar surface.

4.5 Models

The Lunar Inventory Routine will use three models. They will all be FORTRAN subroutines. As of now, no details are available as to the specifics of the input or output from these models. We shall briefly discuss the purpose of each and supply the full details in section 4.6.1.

4.5.1 Death Model

The death model will predict the microbial die-off of the load on the lunar surface.

4.5.2 Lunar Inventory Model

The lunar inventory model is based on the work of Tierney.[1] It makes use of the predicted load at the lunar surface and the type of impact to predict the distribution of the microorganisms on the lunar surface. It also predicts the microbial load at the next Apollo landing site.

4.5.3 Sample Contamination Model

The sample contamination model will use the output of the lunar inventory model (section 4.5.2) to predict the probability that a sample collected by the crew of the next Apollo flight will be contaminated.

4.6 Lunar Inventory Routine

Figures 3 through 3E present flow diagrams of the Lunar Inventory Routine. The discussion following the figures points out some of the main features.

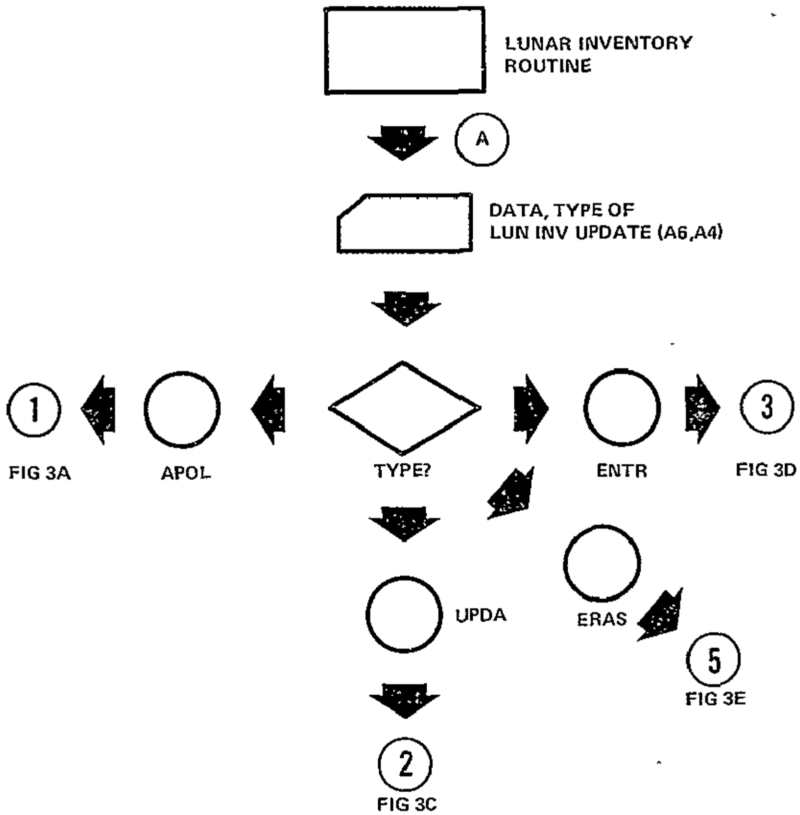


Figure 3. Lunar Inventory Routine

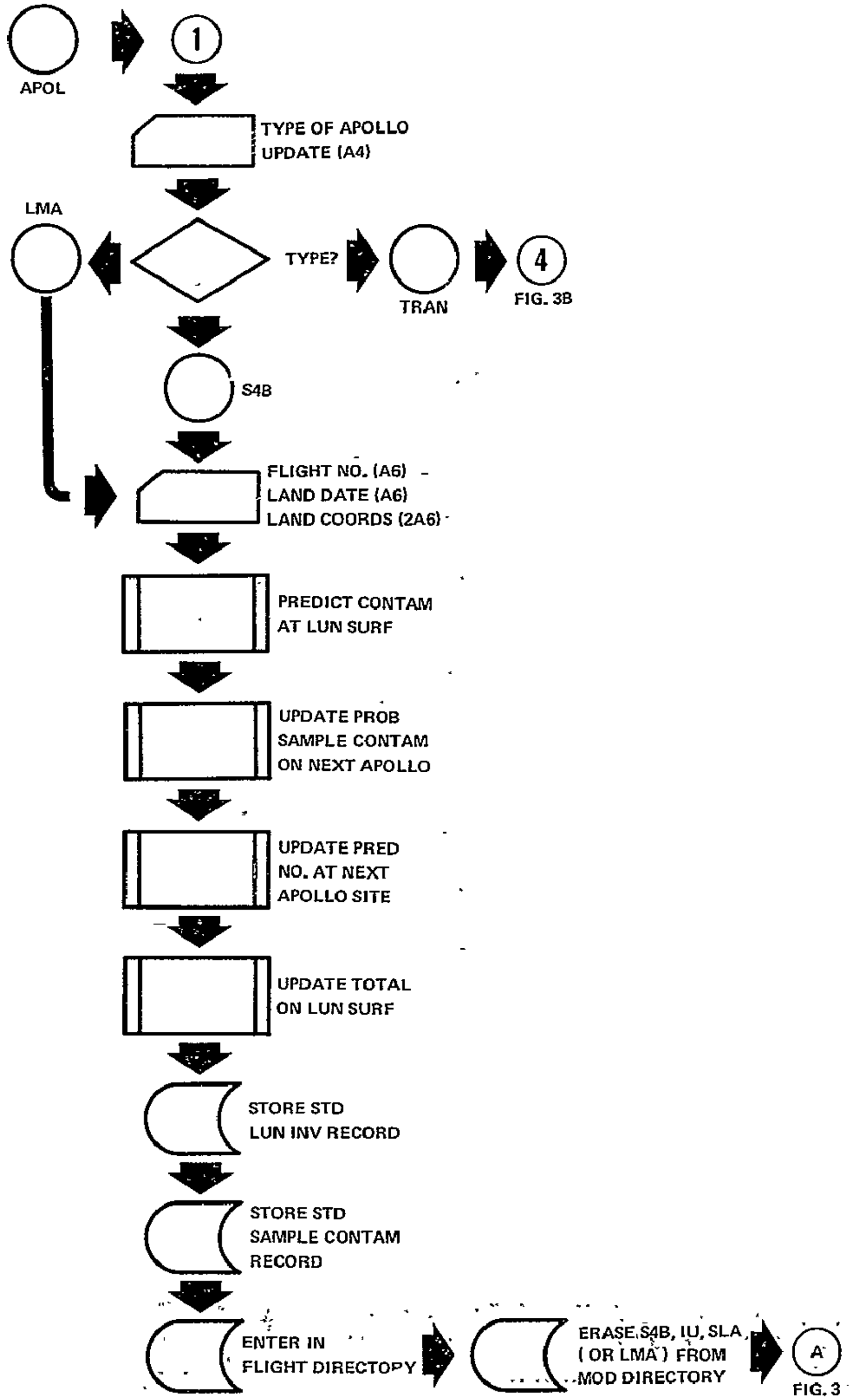


Figure 3A. Lunar Inventory Routine (Continued)

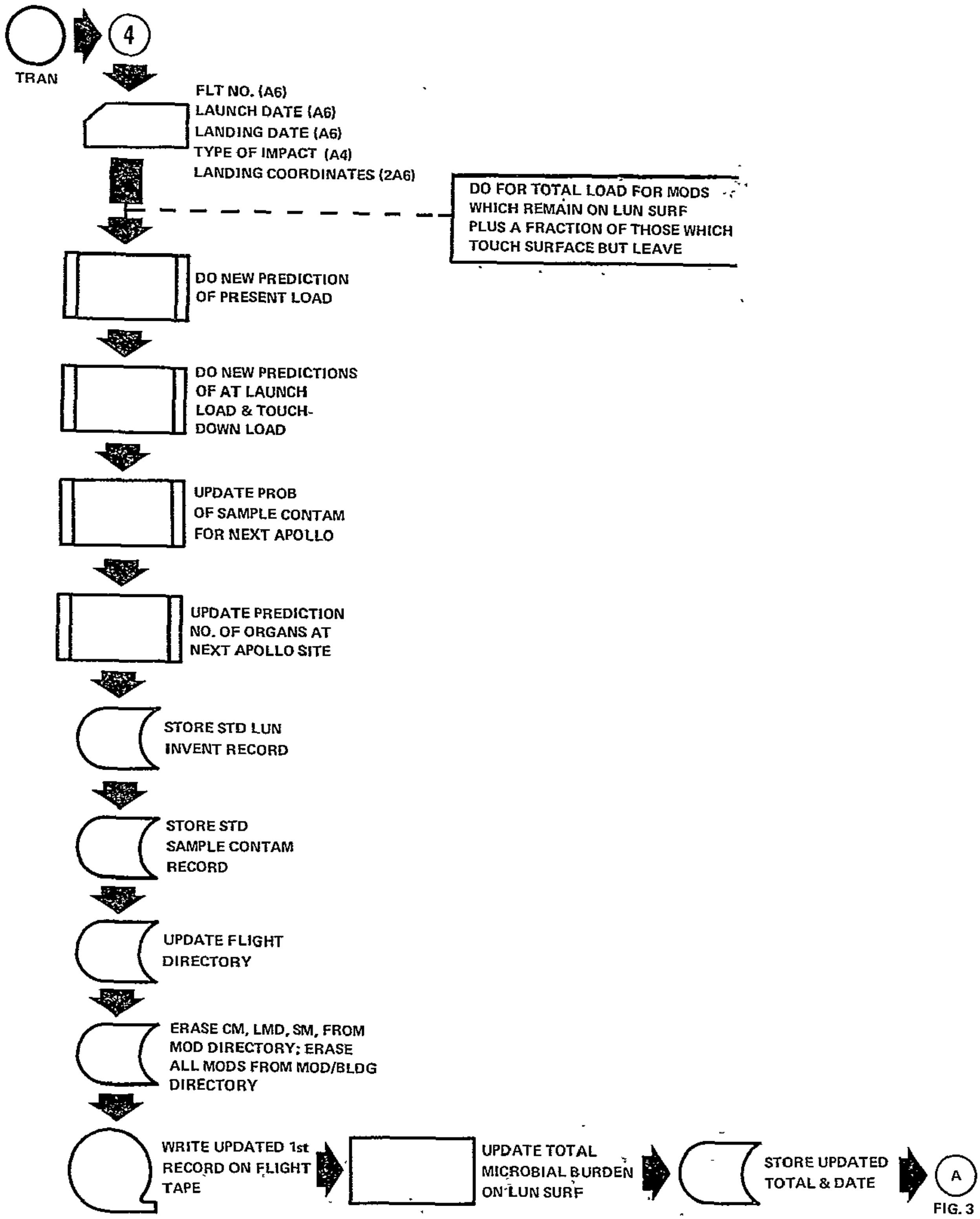


Figure 3B. Lunar Inventory Routine (Continued)

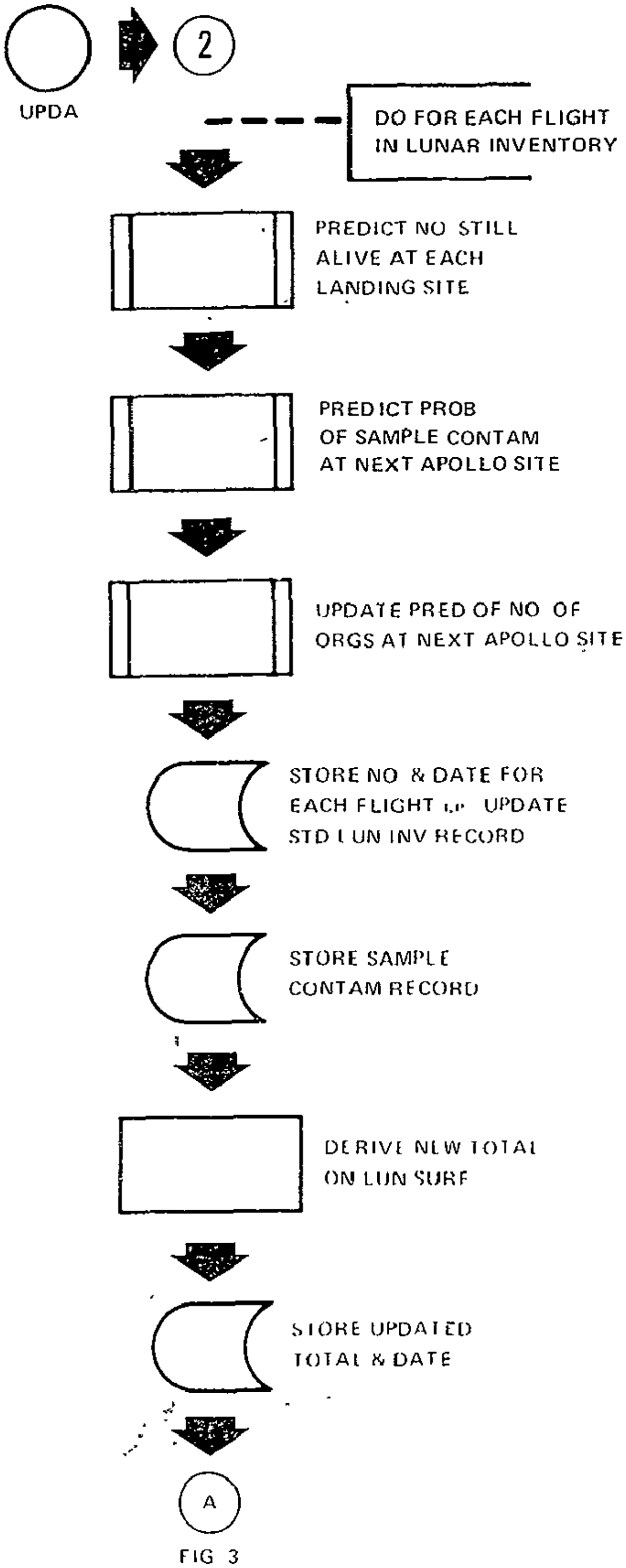


Figure 3C. Lunar Inventory Routine (Continued)

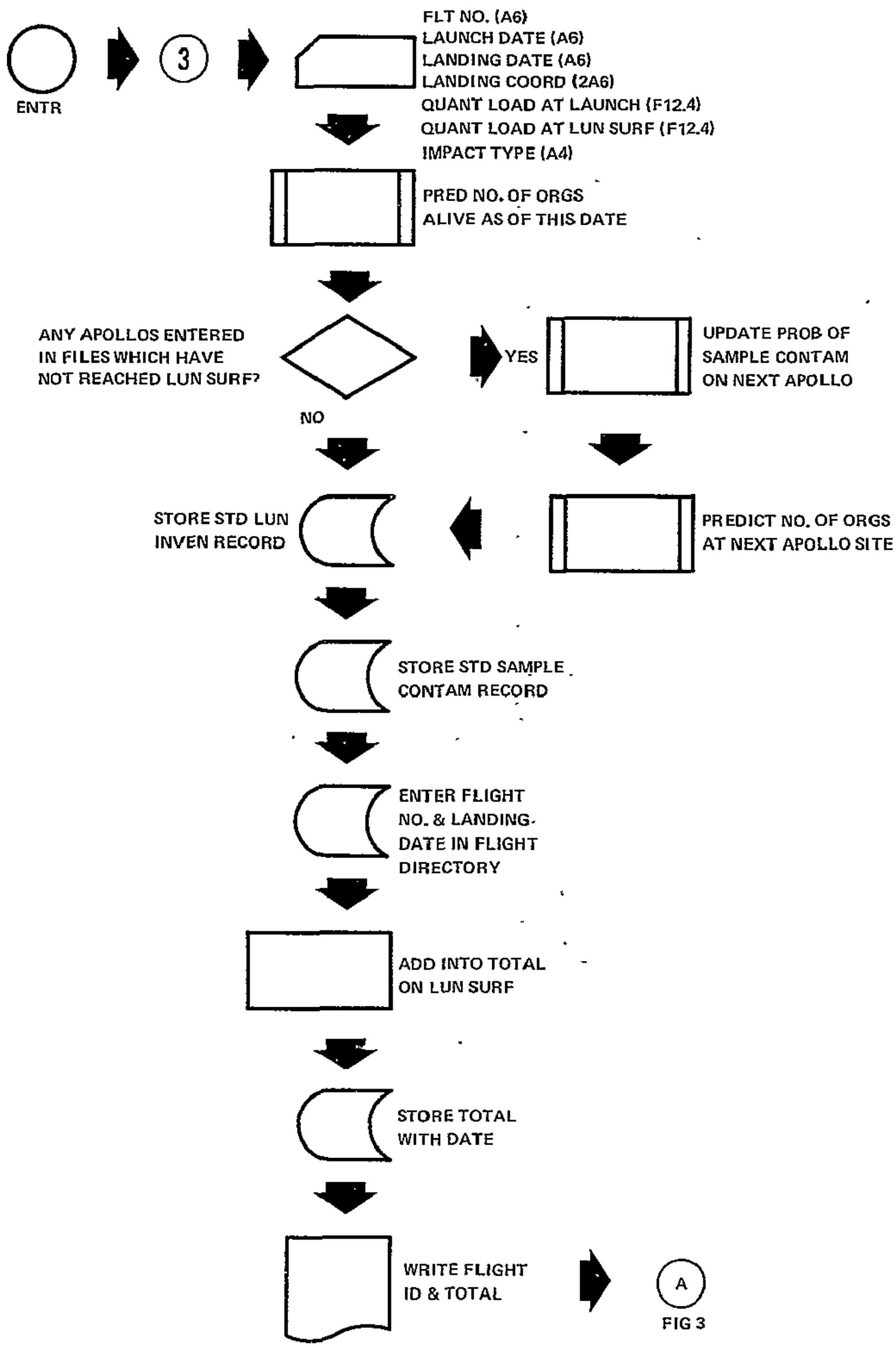


Figure 3D. Lunar Inventory Routine (Continued)

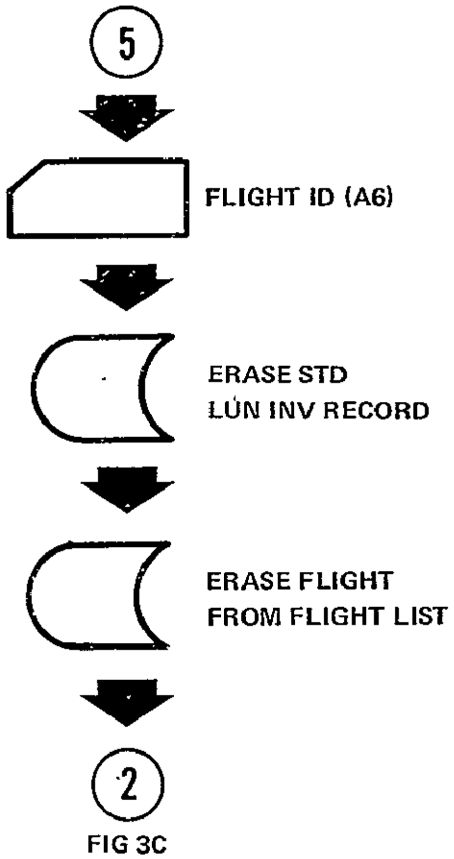


Figure 3E. Lunar Inventory Routine (Continued)

2

4.6.1 Details of Lunar Inventory Routine

For the Lunar Inventory Routine to maintain a standard lunar inventory record on each piece of hardware reaching the lunar surface, as well as a total lunar contamination record, it must have the following capabilities:

a. It must be able to transfer Apollo flights from the active status to the lunar inventory status.
b. It must be able to read and store information available on U.S. and U.S.S.R. unmanned flights and on U.S.S.R. manned flights.
c. It must be able to update predictions of the total bioburden on the lunar surface and to update predictions of the probability of sample contamination at the next Apollo landing site.
d. In case of an error, it must be able to remove information from the lunar inventory and to store corrected data.

This routine consists of four sections. The one which is to be used is specified by the "type of lunar inventory update" read in on the first card. We shall now discuss the possible codes and their meanings.

4.6.1.1 APOL. The code word APOL directs the routine to a section which handles the transfer of information concerning an Apollo flight from the active file to the lunar inventory file. This is done after an Apollo flight has reached the lunar surface. It performs its functions through subsections since there are three different encounters which are possible between a successful Apollo flight and the lunar surface. We shall describe each and the code word which directs the routine to the appropriate subsection.

a. TRAN -- This is used after the landing of the manned lunar module. It performs all calculations required to reflect inventory changes due to biocontamination from the astronauts and the lunar module.

b. S4B -- This is used in case the SIVB hits the lunar surface and is taken from the active files for the Apollo flight.

c. LMA -- The ascent stage of the lunar module is jettisoned in orbit about the moon. Its orbit will decay and it will return to the lunar surface. This LMA bioburden will add to the lunar inventory.

4.6.1.2 UPDA. The code word UPDA is used to update the predicted bioload still alive on the lunar surface at any given time (i.e., the date the program is run) and to update the probability of sample contamination at the next Apollo landing site.

4.6.1.3 ENTR. The code word ENTR is used to enter information about flights which are either not in the active files or for which the information in the active files is not correct. The flights for which we would foresee this being used are unmanned flights by the U.S.S.R. and U.S., manned flights by the U.S.S.R., and Apollo flights for which information is available that makes the information in the active files incorrect.

4.6.1.4 ERAS. The code word ERAS is used to remove a flight from the flight list and to erase its standard lunar inventory record. Its use will normally be followed by the section of the program designated by the code ENTR.

CHAPTER 5. COMMUNICATION ROUTINE

The Communication Routine is that part of the information system which communicates with the Planetary Quarantine Officer via the console in Washington. This portion of the system is designed to be a background routine which will be part of MSOS. All message formats and data are stored on disks. The data storage function is performed by the rest of the information system.

Our discussion of this part of the system will follow the outline used in previous chapters.

5.1 Parameter Formats

Except for message formats and the formats of data entered into the messages, we shall follow the same convention outlined in sections 2.1 and 4.1.

5.2 Directory Formats

The directories are used by the Communication Routine to find information quickly. The formats of these directories are the same as those outlined in section 2.2.

5.3 Storage Record Formats

The storage record formats are those outlined in sections 2.3, 3.3, and 4.3. The storage format for the messages is still undefined and will, in all probability, be left to the programmer.

5.4 File Formats

The files used by this routine are outlined in sections 2.4, 3.4, and 4.4.

5.5 Models

No models are used by this routine.

5.6 Communication Routine

As noted earlier, the console language associated with this routine is cast in a conversational format. The computer displays a question to the user. The user's choice among allowable responses to the question determines the information displayed next on the console. The user can, when desired, obtain a hard copy of the console display by activating the appropriate console control.

5.6.1 General Characteristics

The user calls the computer in accordance with the instructions in the operating manual. The computer, when it senses this interrupt, responds by displaying its first question to the user. The user, in turn, responds by typing in one of the responses appropriate to that question. That response triggers, in the initial steps, another question from the computer to the user. A short series of such questions and responses results in the display of the information desired by the user. Each response to a question narrows the area of interest. For instance, the first response might indicate an interest in Apollo. The second might specify a particular Apollo flight. The third might specify a particular spacecraft module associated with that flight. And the fourth might indicate an interest in the quantitative bioloading on that module.

After each display of data, a return dialog is initiated by the user and the computer asks if other information from the area being probed is desired. A "yes" response draws another question from the computer regarding what information is desired. For instance, the user might have obtained the quantitative loading on a spacecraft module and would now like the qualitative listing for the same module. Through this short reprobing dialog, the user can obtain consecutive displays of the various types of information available at the end of that particular information branch.

A "no" response at this level of the dialog draws a question from the computer regarding interest in a branch at the next higher level of generality. For instance, if the user has obtained all of the information he desires about one module associated with a particular Apollo flight, the computer offers the opportunity to examine another module immediately. A second "no" response brings a question regarding interest at the next higher level, i.e., interest in a different Apollo flight. A third "no" response draws a question regarding further interest in Apollo-connected information. And a fourth "no" brings the opportunity to indicate interest in a different routine altogether, i.e., Lunar Inventory. At any point in this dialog returning to higher levels of generality, the user can respond with a "yes" and proceed, through questions and responses, to the information at the end of another branch. He can move out and back in this manner until he has the information he wants.

The user can terminate his dialog with the computer by responding "off" to <u>any</u> question from the computer. This response causes the machine to break the communication connection.

Each time the computer receives a response to a question, it compares that response with a list of permissible responses to that question. If the response is not permitted, the computer displays the message "Sorry, your response is in error." This can indicate several kinds of error. It can mean that the user has asked the computer to proceed to information not available in the branch being investigated. It can mean that the user, although proceeding properly along a branch, has used a response which is out of sequence. For instance, he might respond "CM quan" before the machine has sequenced beyond the branch where "AP XXX" is an appropriate reply. Or it can simply mean that the operator has made a typographical error in his response. These error possibilities mean that the operator will need to familiarize himself with the information available at the end of each branch and become comfortable with the dialog sequence necessary to proceed out various branches. Hopefully, the language has been designed so the computer will logically cue the next appropriate response set. Also, it is hoped that the response codes to be typed by the user are phonetic enough to obviate any extensive memorization and short enough to minimize typing errors. Practice with the system should develop operator confidence quite rapidly. The operator should use the operating manual quite deliberately during initial uses.

5.6.2 Response Codes

The codes and abbreviations that denote meanings to the computer are given in Appendix I. Full words are typed in response to the computer's initial question, "What program would you like?" Abbreviation-type codes are used in response to all other branching questions. Some mission items carry three-figure numbers (for instance, the command modules have three-digit identifiers, e.g., command module 127), and other items have two-digit identifiers (the lunar modules are such an item). The code "XXX" as used in the definitions indicates any three-figure alphanumeric identification; consequently, "XXX" when identifying a lunar module signifies a zero followed by the two-digit identifier, i.e., lunar module 32 would appear as lunar module 032. This practice simply allows the computer to search standardized identifier formats.

5.6.3 Dialog, Programs, and Information Available

Figures 4 through 4R, which follow this section, present flow diagrams of the Communication Routine.

The computer's first question is "What program would you like?" (Figure 4). The user has two responses available to this question, "Apollo" or "Lunar inventory." The "Apollo" response starts the computer along the Apollo branch and leads, ultimately, to information concerning scheduled flights, items associated with those flights, their locations at the Cape, and their microbial contamination. This "Apollo" branch leads to information concerning only those Apollo flights which are at the Cape or enroute to the moon. Information concerning landed flights is different in character and resides in the Lunar Inventory Routine. A historic listing of data about a landed flight can be obtained from the permanent tape file concerning that flight, but in standard hard copy only and not over the console.

The "Lunar inventory" response directs the computer along that program branch and leads to information concerning the microbial loadings of past automated and manned shots, dates associated with the shots, landing coordinates, the effects of spacecraft loads on lunar microbial loads, and the effects of these lunar microbial loads at the next Apollo landing site and on the lunar samples from the next Apollo flight.

If the user selects the "Apollo" response, the computer then asks "What Apollo flight or module?" (Figure 4A). The user can now elect to examine a total scheduled Apollo flight and each of its modules and stages, or any specific module or stage in the files regardless of its Apollo flight association.

If he wants to examine a total flight, he responds with "Apollo XXX." The computer now asks "What Apollo XXX information do you want?" (Figure 4B). The user has a variety of options at this point. "Item list" gets him a display of the identifications of the astronauts and suits, and the CM, SM, CSM, LM, IU, SLA, and SIVB stage associated with Apollo XXX (Figure 4C). "Total quan" gets him the predictions of the mean total number of aerobic vegetative, anaerobic vegetative, aerobic spore former, and anaerobic spore former cells on all of the modules and stages associated with Apollo XXX for the date of the last samples, for the scheduled launch date, and for the scheduled lunar landing date (Figure 4D). This display also shows the 90 percent confidence level for each type of contamination. The "CM quan" (or other "module quan") response brings, for the module or stage only, the mean numbers and confidence limits for these four types of organisms and three dates (Figure 4E).

The "Total qual" response brings a merged list of the organisms found on all of the modules and stages associated with Apollo XXX as of the last sampling date (Figure 4F). The merging of individual lists means that an organism is listed only once regardless of the number of places in which it is found. This display also indicates the probability that all microorganisms present on the hardware have been detected. The "CM qual" (or other "module qual") response brings the microorganism list and detection probability for that specific Apollo XXX-associated module or stage (Figure 4G). The return dialog initiated by the user after each data display provides the opportunity to move from any one information area on the "AP XXX" spectrum to any other (Figure 4H).

The "CM XXX" (or other "module XXX") response brings the question, "What CM XXX information do you want?" (Figure 4I). Four responses are available: "Flight No." "Location," "CM quan," and "CM qual." The "Flight No." response brings a display identifying the Apollo flight to which CM XXX is assigned (Figure 4J). "Location" brings a display stating the current Cape location (MSOB, VABN, VABS, Pad 39A, or Pad 39B) of CM XXX (Figure 4K). "CM quan" brings the mean number and confidence limit display previously described under the "AP XXX" response, except that here it can be requested for a module or stage associated with any unlanded Apollo flight, not just Apollo XXX (Figure 4E). The "CM qual" response brings the list and detection probability described previously, but again it can relate to a module or stage from any Apollo flight (Figure 4G). Again, the return dialog permits the user to move from display to display for any or all "module XXX" (Figure 4L).

This completes the dialog available under the Apollo program branch.

We will now examine the "Lunar inventory" response to the computer's initial question, "What program would you like?" The machine's second question will be "What lunar inventory information do you want?" (Figure 4M). The user currently has ten responses available to this question. The response "MLAS" (quantitative microbial load at next Apollo site) calls for a display showing the landing coordinates of the next Apollo flight and the microorganisms per square meter alive at that location as of the last updating of the predictions (Figure 4N). A second response, "PCLS" (probability of contamination of next lunar samples), brings a display stating the probability of contaminating the next lunar surface samples with previously deposited microbes and with microbes predicted to arrive with the next Apollo flight (Figure 4O). The response "LLML" (total live lunar microbial load) generates a statement of the total number of live microbes remaining on the lunar surface from all previous lunar missions, manned and unmanned (Figure 4P). A series of seven basic response codes bring a display of the contamination remaining alive from any previous mission. These basic responses are MCAP XXX (microbial contamination from the Apollo XXX mission), and MCRA XXX, MCSV XXX, MCAI XXX, MCLO XXX, MCLU XXX, and MCRM XXX which denote the same information for Ranger (RA), Surveyor (SV), AIMP (AI), Lunar Orbiter (LO), Russian Luna (LU), and Russian manned (RM) sites (Figure 4Q). The display for each site contains the flight identification, launch date, landing date, landing coordinates, quantitative microbial load at launch, quantitative microbial load at impact, and the number of microbes still alive as of the date of last update. The return dialog available after each display provides an opportunity for examining consecutively any of the displays available from the Lunar Inventory Routine (Figure 4R).

This completes the dialog currently available from the remote console.

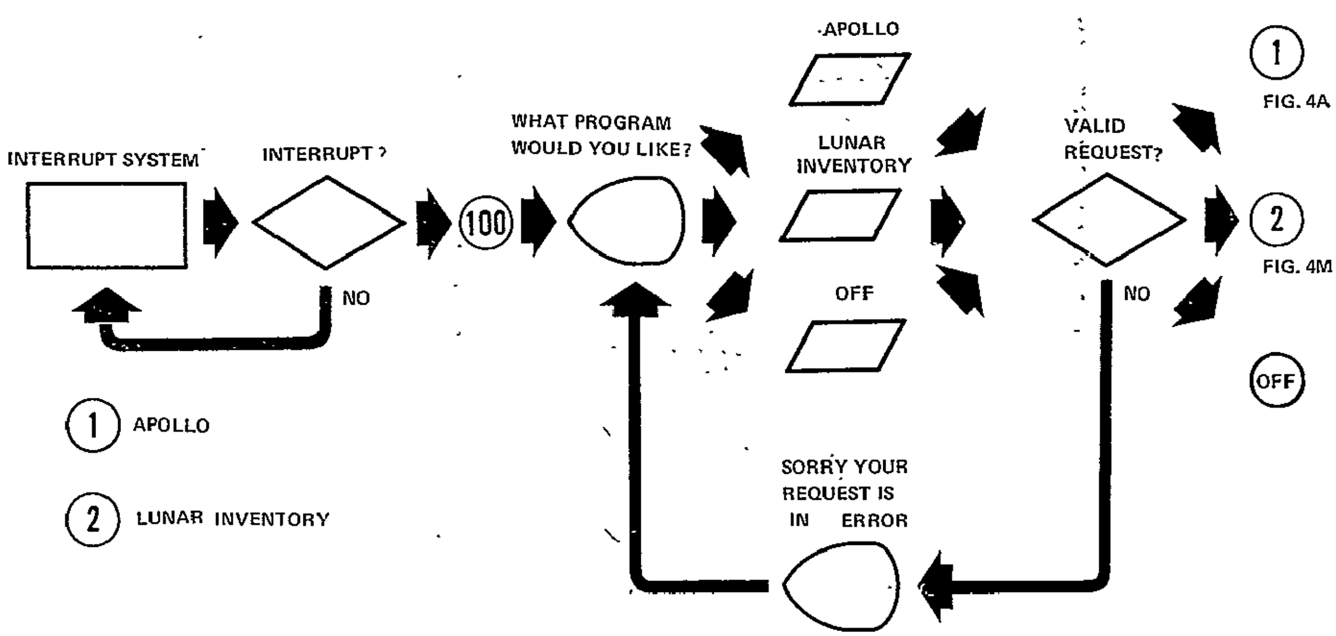


Figure 4. Background Program Console Language

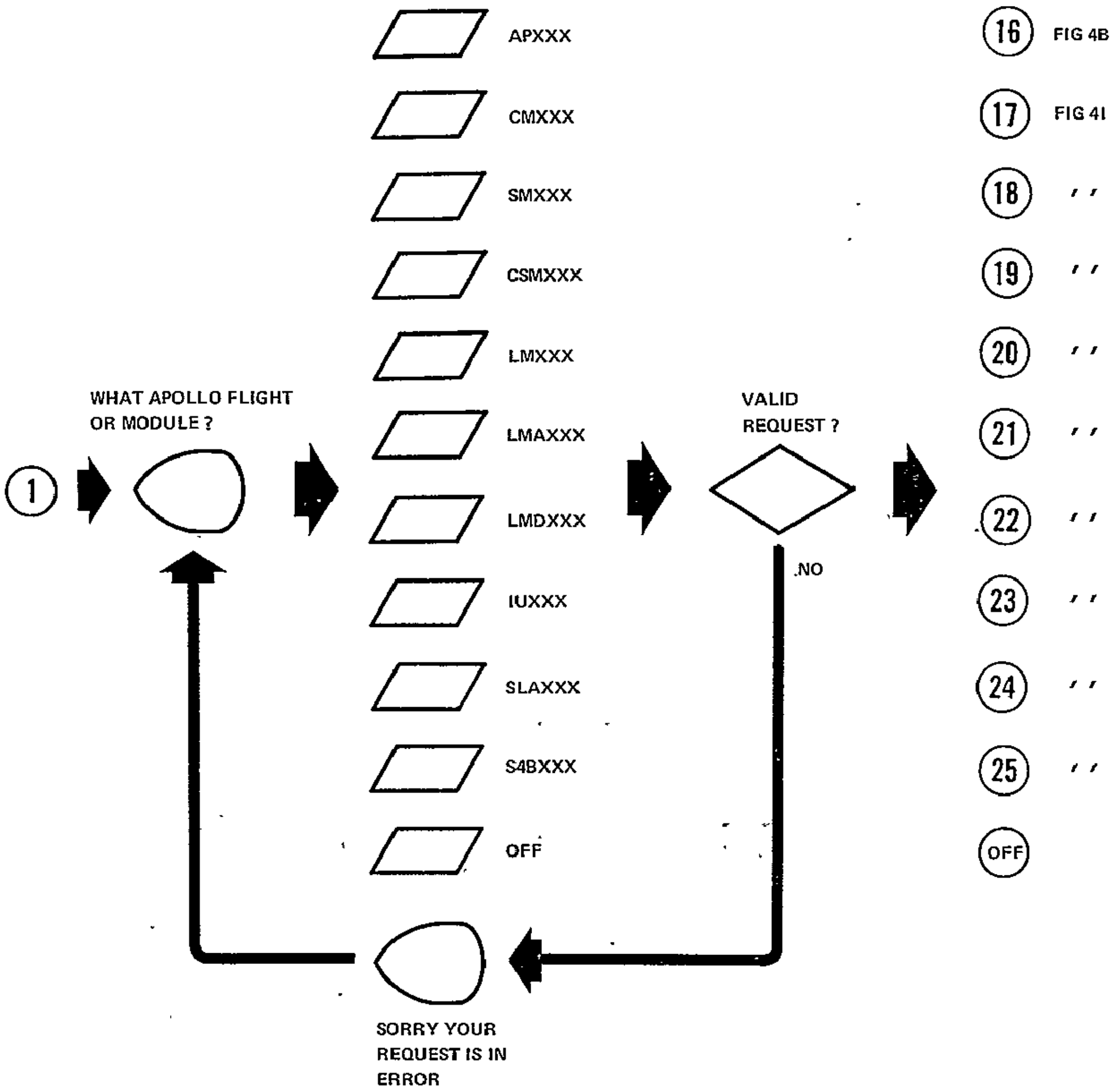


Figure 4A. Background Program Console Language, Apollo.

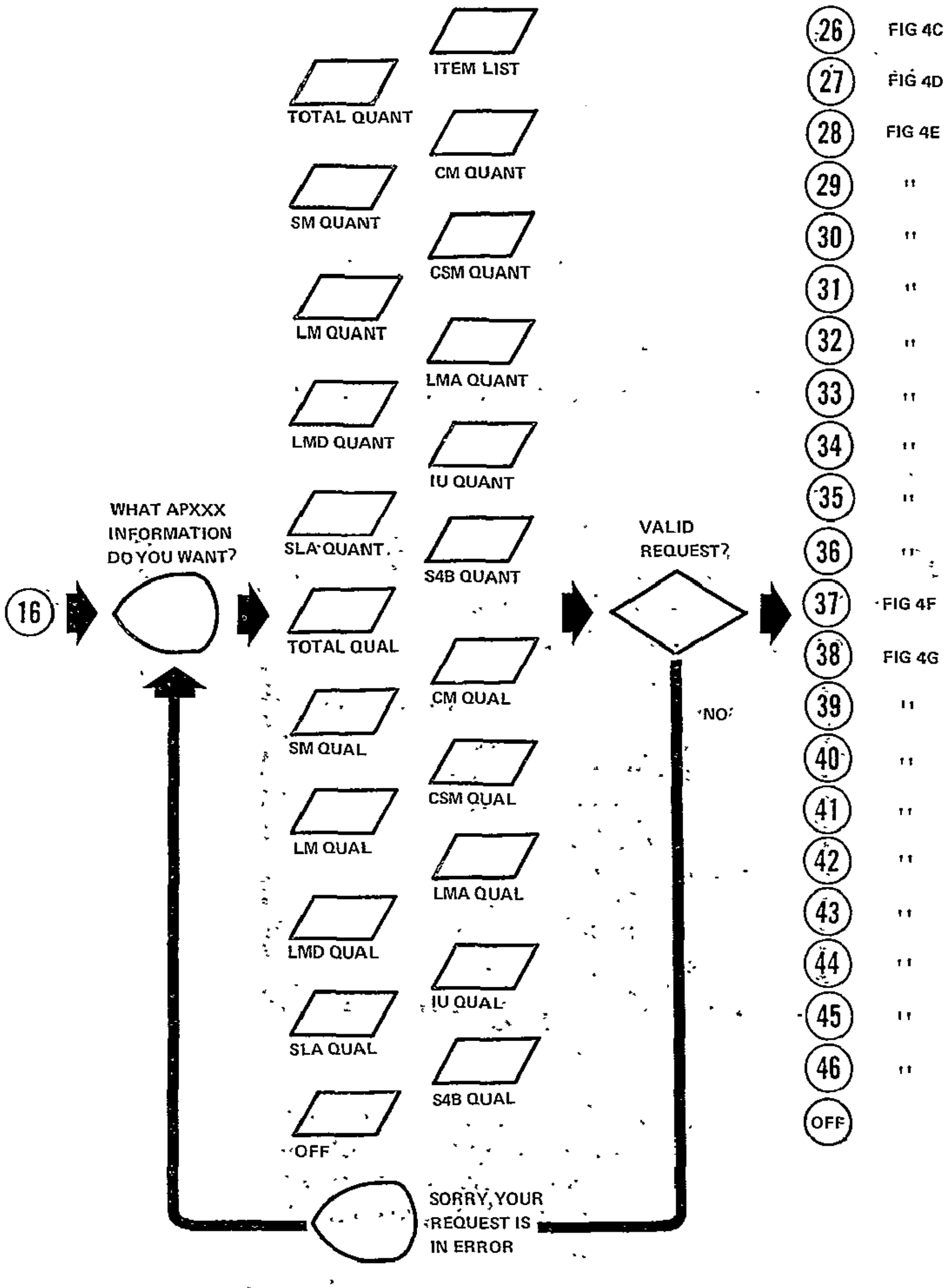


Figure 4B. Background Program Console Language, Apollo (Continued)

26

APOLLO XXX LIST

ASTRONAUT	SUIT ID
______	______
______	______
______	______
______	______
______	______
______	______

CMXXX SMXXX CSMXXX
LMXXX IUXXX SLAXXX
S4BXXX

47
FIG. 4H

Figure 4C. Background Program Console Language, Apollo (Continued)

27

TOTAL QUANTITATIVE LOAD ON
APXXX PREDICTED AS OF XX/XX/XX

MEAN NO.	AS OF LAST SAMPLE	AT LAUNCH	AT TOUCHDOWN
AER. VEG.	AEB	AEB	AEB
ANAER. VEG.	AEB	AEB	AEB
AER. S. F.	AEB	AEB	AEB
ANAER. S. F.	AEB	AEB	AEB
90% CONF LIMIT			
AER. VEG.	AEB	AEB	AEB
ANAER. VEG.	AEB	AEB	AEB
AER. S. F.	AEB	AEB	AEB
ANAER. S. F.	AEB	AEB	AEB

47
FIG. 4H

NOTE: TOTAL QUANTITATIVE LOAD ON APXXX IS THE SUM OF THE QUANTITATIVE LOADS ON ALL MODULES ASSOCIATED WITH APXXX

AEB = A x 10^B, WHERE A IS A NUMBER LESS THAN 1 AND B IS AN INTEGER

Figure 4D. Background Program Console Language, Apollo (Continued)

(28) → QUANTITATIVE LOAD ON CMXXX PREDICTED AS OF XX/XX/XX:

MEAN NO.	AS OF LAST SAMPLE	AT LAUNCH	AT TOUCHDOWN
AER VEG	AEB	AEB	AEB
ANAER VEG	AEB	AEB	AEB
AER S. F.	AEB	AEB	AEB
ANAER S. F.	AEB	AEB	AEB
90% CONF LIMIT			
AER VEG	AEB	AEB	AEB
ANAER VEG	AEB	AEB	AEB
AER S. F.	AEB	AEB	AEB
ANAER S. F.	AEB	AEB	AEB

→ (47) FIG 4H
(ONLY FOR INFO THRU APXXX BRANCH FOR INFO THRU ——XXX BRANCH
→ (50) FIG 4L)

(29) SM QUAN
(30) CSM QUAN
(31) LM QUAN
(32) LMA QUAN
(33) LMD QUAN
(34) IU QUAN
(35) SLA QUAN
(36) S4B QUAN

→ QUANTITATIVE LOAD ON — — — XXX PREDICTED AS OF XX/XX/XX.

MEAN NO.	AS OF LAST SAMPLE	AT LAUNCH	AT TOUCHDOWN
AER VEG	AEB	AEB	AEB
ANAER VEG	AEB	AEB	AEB
AER S. F.	AEB	AEB	AEB
ANAER S. F.	AEB	AEB	AEB
90% CONF LIMIT			
AER VEG	AEB	AEB	AEB
ANAER VEG	AEB	AEB	AEB
AER S. F.	AEB	AEB	AEB
ANAER S F.	AEB	AEB	AEB

→ (47) FIG 4H
(ONLY FOR INFO THRU APXXX BRANCH FOR INFO THRU ——XXX BRANCH
→ (50) FIG 4L)

Figure 4E. Background Program Console Language, Apollo (Continued)

(37) → QUALITATIVE LOAD ON APXXX PREDICTED AS OF XX/XX/XX:

DATE LAST SAMPLE (ORGANISM)	AT LAUNCH (ORGANISM)	AT TOUCHDOWN (ORGANISM)

(ROOM TO LIST UP TO 100 ORGANISMS)
PROBABILITY ALL ORGANISMS DETECTED = .XX

→ (47) FIG 4H

NOTE: TOTAL QUALITATIVE LOADING ON APXXX IS MERGING OF QUALITATIVE LOADINGS ON ALL ITEMS ASSOCIATED WITH APXXX
THIS LISTING WILL REQUIRE 6 DISPLAYS.
PROGRAM TO FOLLOW EACH OTHER FROM CONSOLE SIGNAL.

Figure 4F. Background Program Console Language, Apollo (Continued)

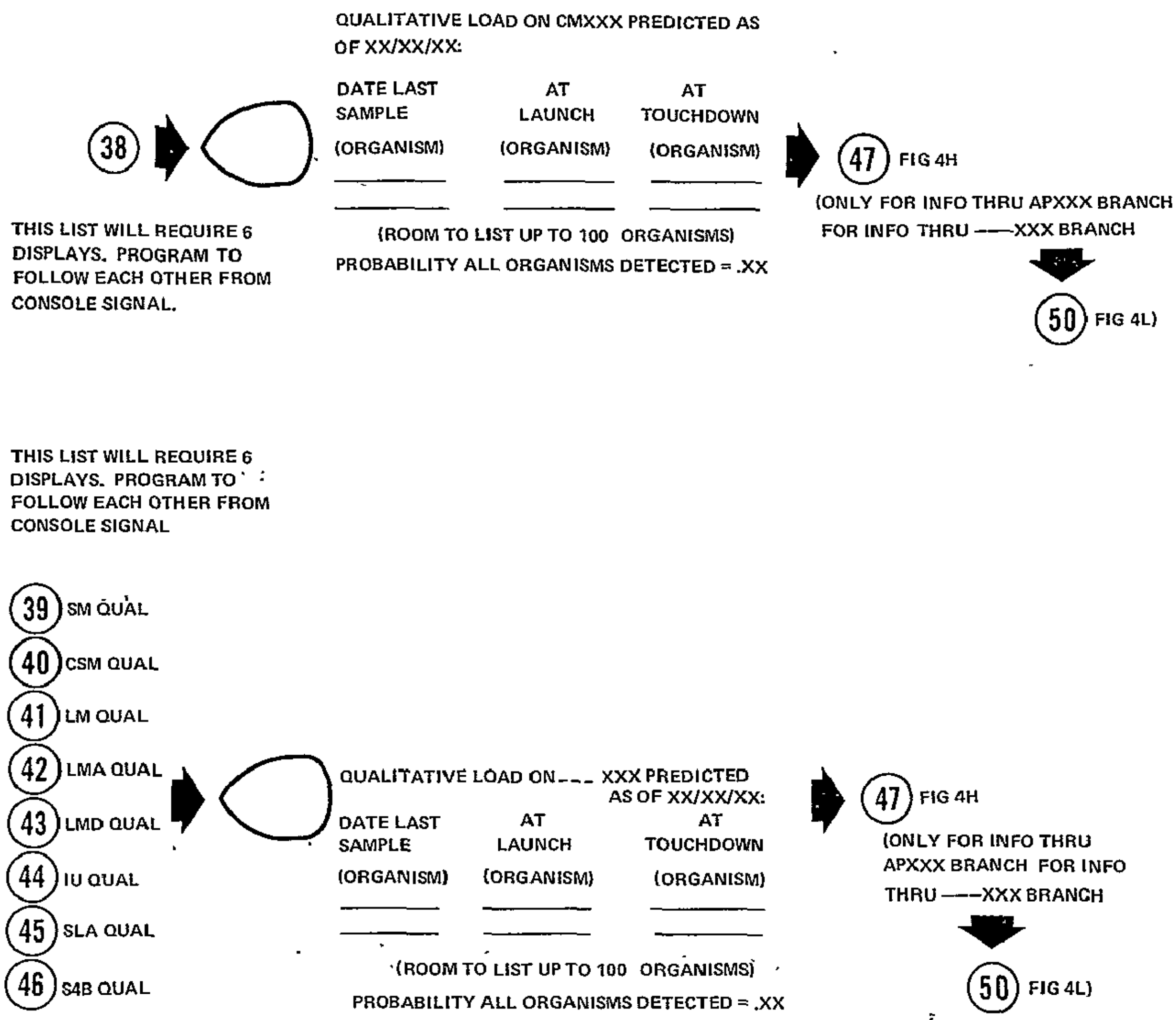


Figure 4G. Background Program Console Language, Apollo (Continued)

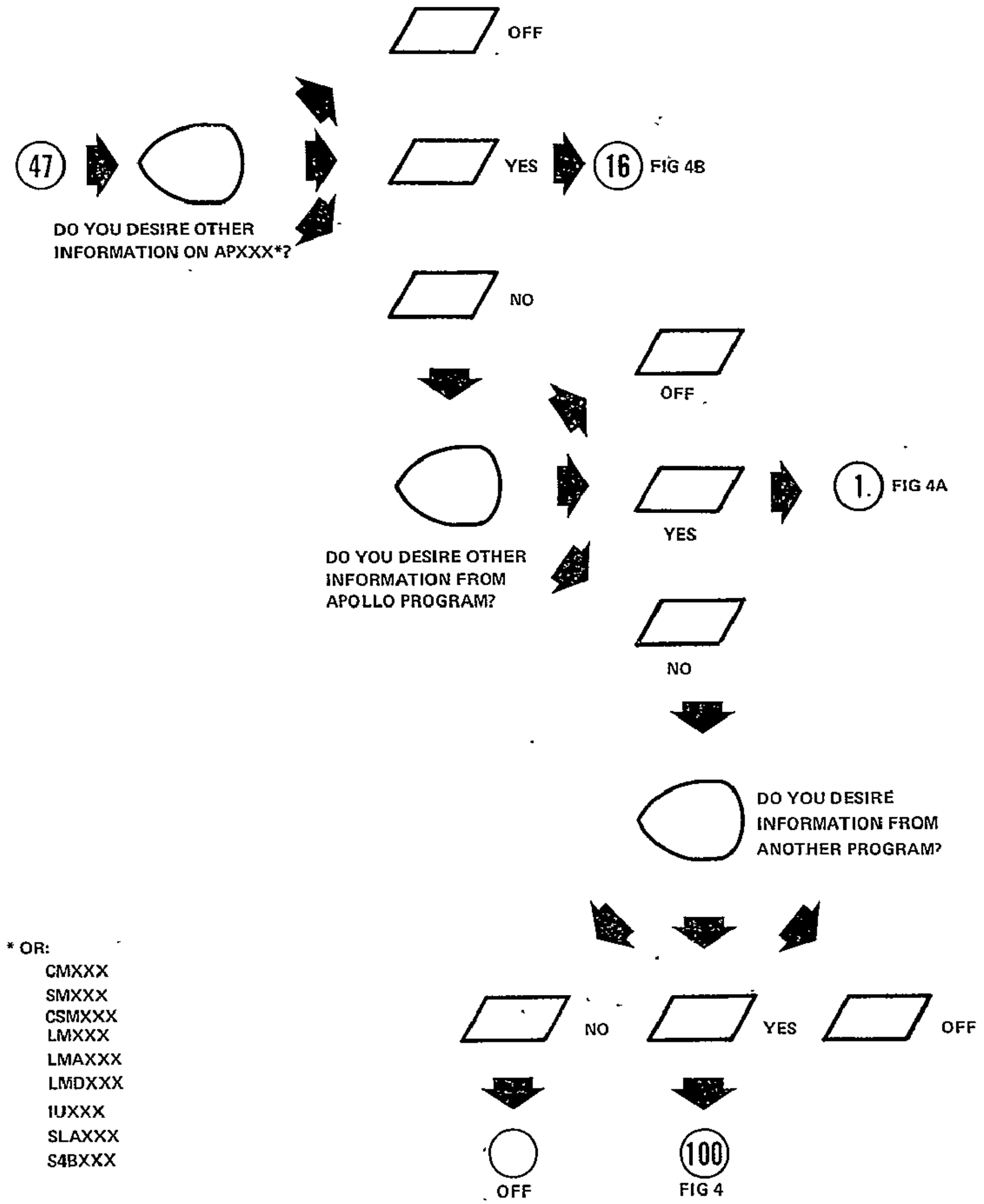


Figure 4H. Background Program Console Language, Apollo (Continued)

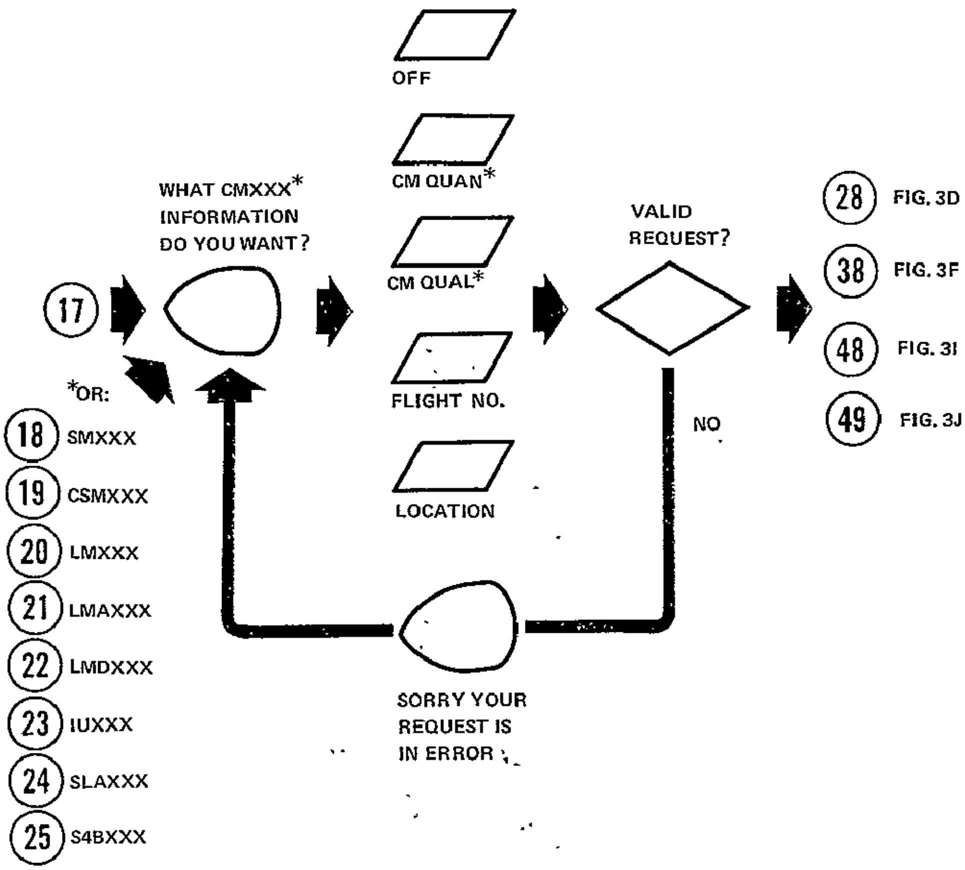


Figure 4I. Background Program Console Language, Apollo (Continued)

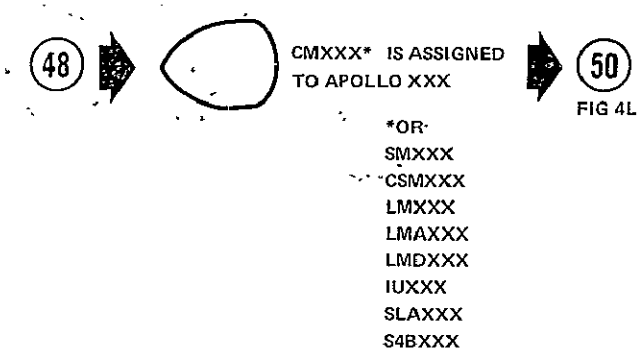


Figure 4J. Background Program Console Language, Apollo (Continued)

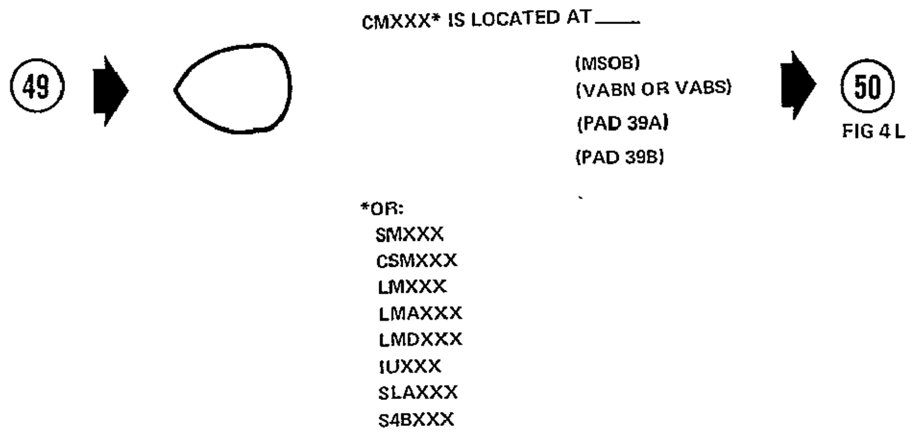


Figure 4K. Background Program Console Language, Apollo (Continued)

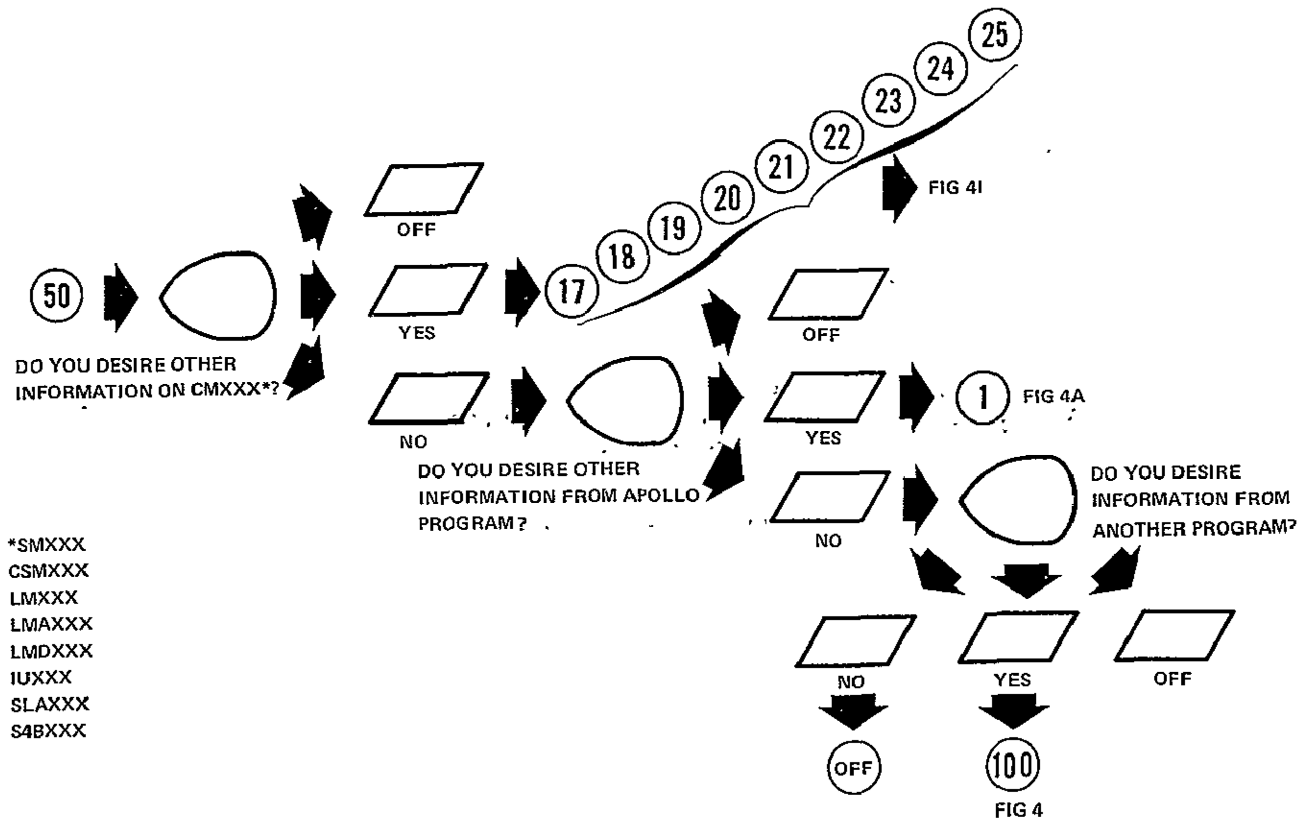


Figure 4L. Background Program Console Language, Apollo (Continued)

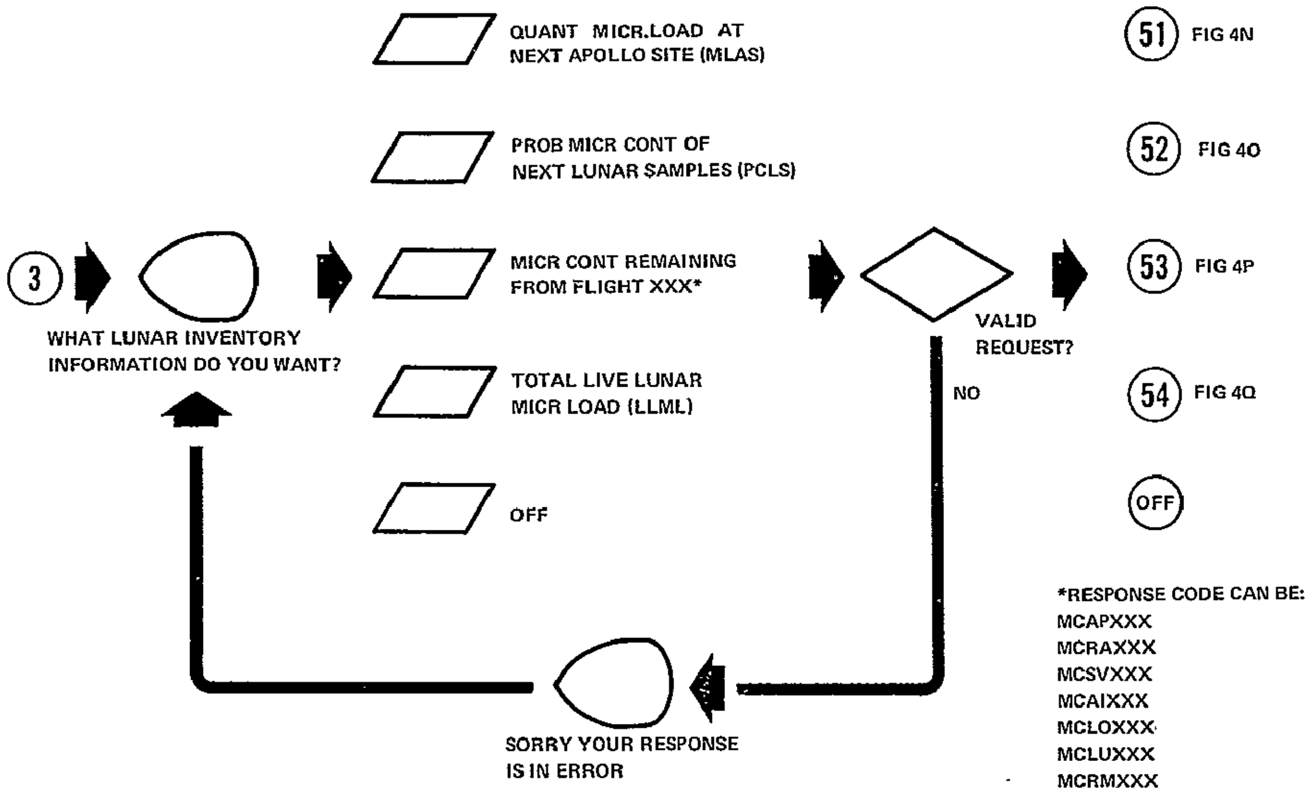


Figure 4M. Background Program Console Language, Lunar Inventory

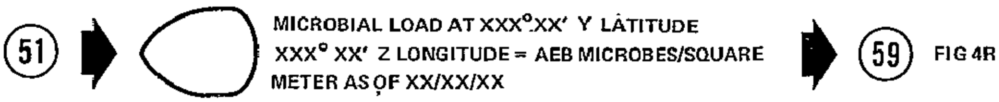


Figure 4N. Background Program Console Language, Lunar Inventory (Continued)

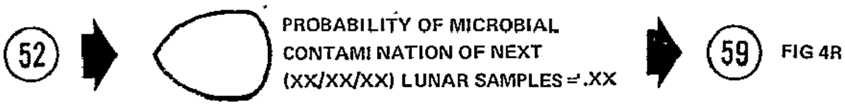


Figure 4O. Background Program Console Language, Lunar Inventory (Continued)

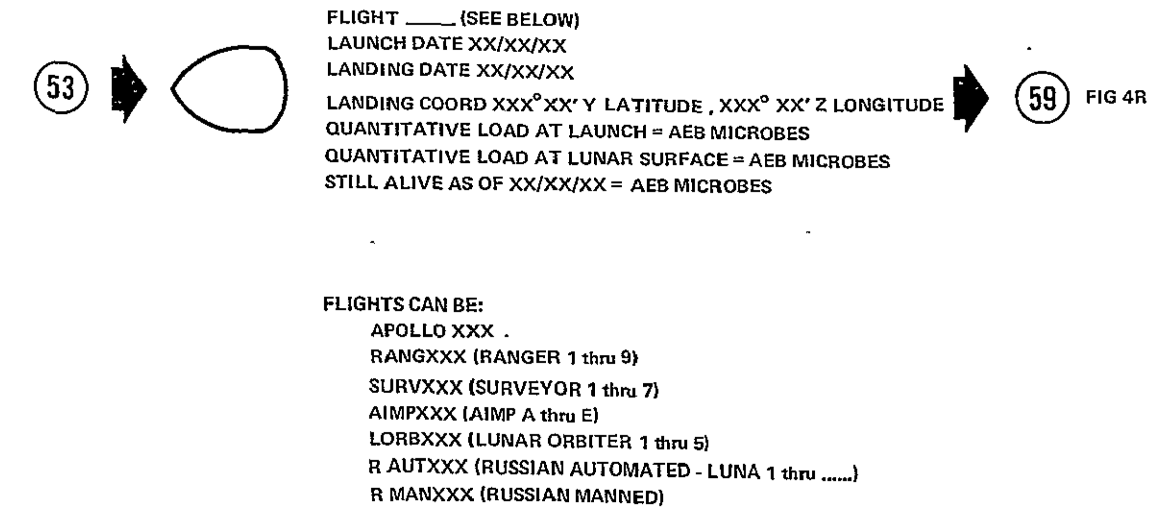


Figure 4P. Background Program Console Language, Lunar Inventory (Continued)

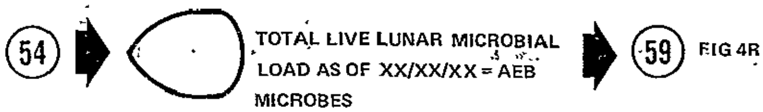


Figure 4Q. Background Program Console Language, Lunar Inventory (Continued)

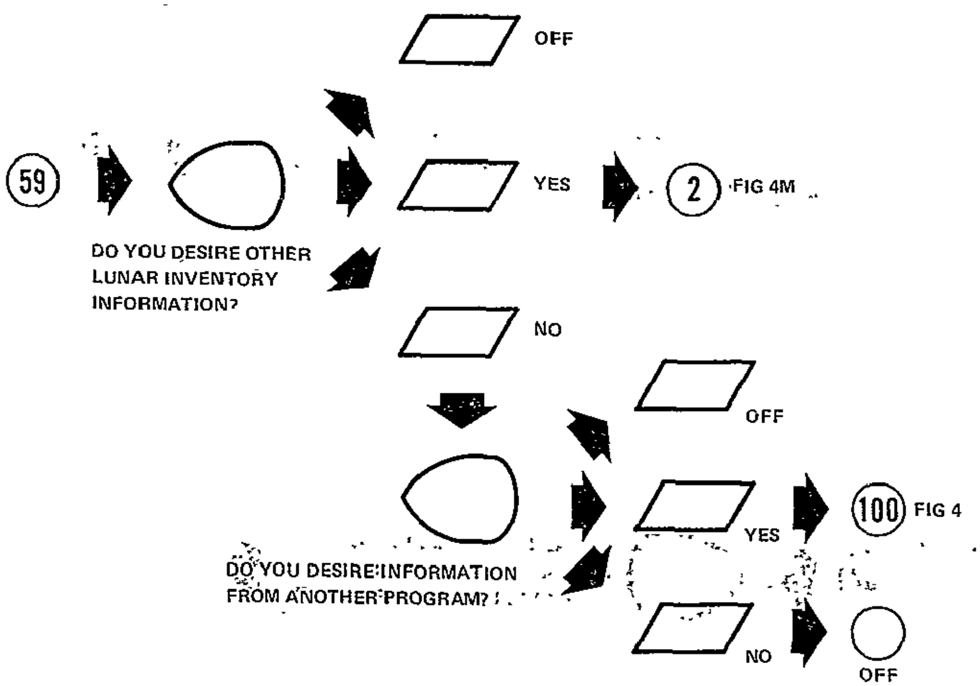


Figure 4R. Background Program Console Language, Lunar Inventory (Continued)

CHAPTER 6. OTHER DESIGN CONSIDERATIONS

The system features described in previous chapters are the result of a series of decisions regarding what should and should not be included in the system design. The material of this document to this point reflects decisions regarding what should be in the design. This chapter will address several features which are not in the design and will discuss the factors which led to the decision to exclude them.

6.1 Console Display of Astronaut Bioflora Data

Astronaut bioflora data, while expected to constitute an important parameter in biocontamination predictions, are not available as a console display. Two factors led to the exclusion decision. First, bioflora data are considered to be highly personal medical information, and not the kind of information the donor would ordinarily want examined on a casual basis by anyone other than those people responsible for his physical well-being. This ethical consideration was the basic one involved in the decision. Second, human bioflora information, unlike data gathered from hardware or working areas, cannot be modified or minimized, nor can criteria be established concerning it. The Planetary Quarantine Office has no policy responsibility with respect to it. Therefore, it is not a data item about which decisions can or must be made -- except, of course, for the medical fitness decisions made by the astronaut's physician. On this basis alone, it does not seem a necessary or useful item for console display.

6.2 Console Display of Suit Leakage Data

The leakage characteristics of the astronaut suits are expected to be a parameter in biocontamination estimates and predictions; however, they are not included as a console display item because suit leakage is expected to be a constant characteristic for a given suit under given conditions. It will probably be measured and will not be expected to change under ordinary circumstances. Therefore, it is not a factor to be monitored in any sense for changes or for control or certification purposes. Under this circumstance, its inclusion in the console displays was considered to be of minimal usefulness.

6.3 Console Display of Environmental Information

Information concerning the environments in which spacecraft are stored and worked on will, again, provide important parameters in biocontamination calculations. Considerable throught was given to what kinds of statements about buildings and other environments constitute meaningful information for use in the Planetary Quarantine Office. The most meaningful statements appear to be those concerning the levels of airborne biocontamination existing at various places and times. This conclusion implies statements about either the ambient concentrations (microbes/unit volume) in the air or the rates at which biocontaminants are falling onto and "leaving" surfaces in the environment. On large area/large volume bases, these statements, in turn, imply the gathering of air samples and/or deposition samples from a multitude of representative locations in the various environments experienced by spacecraft hardware.

The data gathering capabilities predicted for this system will not support this level of sampling. A relatively few air samples will be taken to establish and confirm model parameters, but air sampling will not be a routine method of gathering environmental information. Many deposition samples will be taken, but primarily in the immediate vicinity of the hardware to permit as direct as possible relation of the deposition information to hardware contamination.

Neither of these sampling efforts will support any meaningful statements concerning the general environment in the various buildings and other areas at the Cape. Therefore, the unavailability of appropriate environmental data precludes any console display of that type of information.

6.4 Data Error Correction Program

The data to be handled by this system consist of the results of bioassays from the various parts of the spacecraft/astronaut system. The other information handled by the system consists of identifications, dates, and landing coordinates. It is almost certain that errors will occur in the parameters during the anticipated inputs to the system. The correction of erroneous identification, date, and landing coordinate inputs can be handled straightforwardly as updates to the File Preparation Routine for active Apollo flights and through the ERAS and ENTR subsections of the Lunar Inventory Routine.

The correction of erroneous bioassay data inputs is not possible with this system design. The decision to omit an error correction routine was based on comparisons of the costs of programming and running such a routine with the effects of erroneous data on system calculations and outputs. Since bioassay data experience a leveling or averaging influence due to the combining of a number of assay results for a particular calculation, one erroneous datum will have relatively little influence on calculation results. On the other hand, the correction of a bioassay datum can become extremely expensive in terms of computer time. Every calculation involving that input must be redone. In the case of erroneous environmental data, the worst case, the changing of an assay input would require recalculation of every date of sample prediction, date of launch prediction, date of impact prediction, lunar inventory prediction, and sample contamination prediction relating to or affected by any module of any flight that had processed through the environmental area in question since the date of the erroneous input. The hours of recalculation time which could be involved would still not produce a noticeable change in the predictions due to the small effect of any one piece of data.

Admittedly, the notion of a known and uncorrectable error in a system is an uncomfortable one. To minimize this discomfort, Public Health Service personnel will subject input data to a verification examination prior to entering the data into the machine, and the machine will be programmed to store, but to delete from calculations, those data which lie outside defined limits.

The system itself provides a certain error detection capability. The Data Storage Routine expects a qualitative identification for every assay colony. Thus, the process of actually reading in the data has a check built into it. It will stop if the correct number of qualitative results is not present. If it is ever absolutely necessary to correct a bioassay input, it can be done by printing out a listing of all data entered into memory since the start of the program, erasing the memories, correcting and re-entering the data into memory, and redoing all calculations. We believe one would want to be absolutely certain of his motivation before electing this action.

LIST OF REFERENCES

1. Tierney, M. S., The Chances of Retrieval of Viable Microorganisms Deposited on the Moon by Unmanned Lunar Probes, to be published.

2. Roark, A. L., C. A. Trauth, Jr., and R. E. Woolsey, A Model for the Optimal Identification of Bacteria, to be published.

3. 3100/3200/3500 Computer Systems FORTRAN Reference Manual, Control Data Corporation, Pub. No. 60157600, Rev. A, April 1966.

4. Dugan, V. L., W. J. Whitfield, J. J. McDade, J. W. Beakley, and F. W. Oswalt, A New Approach to the Microbiological Sampling of Surfaces: The Vacuum Probe Sampler, Sandia Laboratories Publication SC-RR-67-114, February 1967.

APPENDIX I

COMMUNICATION ROUTINE ABBREVIATIONS

Abbreviations used in the Communication Routine have the following meanings:

Code	Meaning
CM	Command module.
SM	Service module.
CSM	Command and service module assembly.
LM	Lunar module.
LMA	Ascent stage of the lunar module.
LMD	Descent stage of the lunar module.
IU	Instrument unit.
SLA	Saturn to lunar module adaptor.
S4B	Saturn SIVB stage.
APOLLO	I want information from the Apollo computer program.
LUNAR INVENTORY	I want information from the Lunar Inventory Routine.
AP XXX	I want information concerning the specific Apollo flight XXX.
CM XXX	I want information concerning command module XXX.
SM XXX	I want information concerning service module XXX.
CSM XXX	I want information concerning command and service module XXX.
LM XXX	I want information concerning lunar module XXX.

Code	Meaning
LMA XXX	I want information concerning lunar module XXX ascent stage.
LMD XXX	I want information concerning lunar module XXX descent stage.
IU XXX	I want information concerning instrument unit XXX.
SLA XXX	I want information concerning Saturn to lunar module adaptor XXX.
S4B XXX	I want information concerning Saturn SIVB stage XXX.
ITEM LIST	I want a list of the astronauts, suits, and modules associated with Apollo XXX.
TOTAL QUAN	I want the sum of the quantitative microbial load predictions for the CM, SM, LM, IU, SLA, and SIVB stages assigned to Apollo XXX.
CM QUAN	1. I want the quantitative microbial load predictions for the CM associated with Apollo XXX (if the user has arrived at this response via the Apollo XXX dialog branch).
——	2. I want the quantitative microbial load predictions for CM XXX (if the user has arrived at this response via the CM XXX dialog branch).
SM QUAN CSM QUAN LM QUAN LMA QUAN LMD QUAN IU QUAN SLA QUAN S4B QUAN	The meanings are the same as for CM QUAN, except that the quantitative predictions for the different module or stage are wanted.

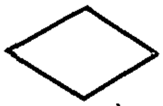
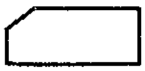
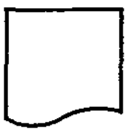
Code	Meaning
TOTAL QUAL	I want a merged list of the microorganisms which have been found on all of the modules associated with Apollo XXX.
CM QUAL	1. I want a list of the microorganisms which have been found on the CM associated with Apollo XXX (if the user has arrived at this response via the Apollo XXX dialog branch). 2. I want a list of the microorganisms found on CM XXX (if the user has arrived at this response via the CM XXX dialog branch).
SM QUAL CSM QUAL LM QUAL LMA QUAL LMD QUAL IU QUAL SLA QUAL S4B QUAL	The meanings are the same as for CM QUAL, except that the microorganism list is for the specific module or stage indicated in the code.
MLAS	I want the microbial load at the next Apollo site.
PCLS	I want the probability of contamination of next lunar samples.
MCAP XXX	I want the microbial contamination remaining from the Apollo XXX mission.
MCRA XXX	I want the microbial contamination remaining from the Ranger XXX mission (Ranger 001 through 009).
MCSV XXX	I want the microbial contamination remaining from the Surveyor XXX mission (Surveyor 001 through 007).
MCAI XXX	I want the microbial contamination remaining from the AIMP XXX mission (AIMP 00A through 00E).

Code	Meaning
MCLO XXX	I want the microbial contamination remaining from the lunar orbiter XXX mission (lunar orbiter 001 through 005).
MCLU XXX	I want the microbial contamination remaining from the Luna (Russian unmanned) mission (Luna 001 through . . . . .).
MCRM XXX	I want the microbial contamination remaining from the Russian manned XXX mission.
LLML	I want the total live lunar microbial load.

APPENDIX II

FLOW CHART SYMBOLS

The symbols used in the flow charts in this document are portrayed below, along with explanations of what their use implies in terms of program actions.

SYMBOL	USAGE	SYMBOL	USAGE
	1. MAJOR PROGRAM TITLES 2. PROCESSING INSTRUCTIONS		1. CATHODE-RAY TUBE DISPLAYS AT THE CONSOLE CONSISTING OF MESSAGES AND QUESTIONS TO THE CONSOLE OPERATOR
	1. PRE DEFINED PROCESSES, ie., SUBROUTINES		1. MAGNETIC TAPE STORAGE OF INFORMATION
	1. LOGIC QUESTIONS AND PROGRAM DECISION POINTS		1. DISK STORAGE OF INFORMATION
	1. PUNCHED-CARD DATA INPUT		1. TITLES OF SUBSECTIONS OF MAJOR PROGRAMS
	1. REPLIES BY CONSOLE OPERATOR TO COMPUTER		1. CONNECTOR WHICH CONTAINS CODE NUMBER TO NEXT STEP IN PROGRAM LOGIC
	1. AUTOMATIC, ROUTINE PRINTOUTS		

DISTRIBUTION:
L. B. Hall, NASA
Code SB
400 Maryland Avenue, S.W.
Washington, D.C. 20546 (25)

Martin S. Favero
Department of Health, Ed. and Welfare
CDC-Phoenix Field Station
4402 North 7th Street
Phoenix, Arizona 85014 (5)

J. R. Puleo
Spacecraft Bioassay Laboratory
Drawer Y
Cape Canaveral, Florida 32920 (4)

J. A. Hornbeck, 1
W. J. Howard, 1000
W. A. Gardner, 1500
H. E. Lenander, 1600
T. M. Burford, 1700
J. W. Weihe, 1710
D. P. Peterson, 1711
M. J. Norris, 1720
J. M. Worrell, Jr., 1721
H. D. Sivinski, 1740 (67)
D. R. Cotter, 1800
B. H. VanDomelen, 1830
R. T. Dillon, 1830
R. W. Henderson, 2000
L. D. Smith, 2300
J. W. Jones, 2350
L. J. Paddison, 2400
R. S. Claassen, 2600
J. R. Sublett, 2640
R. B. Powell, 3000
C. W. Campbell, 4000
S. J. Buchsbaum, 5000
S. J. Buchsbaum, 5200, Acting
C. F. Bild, 5400
D. W. Ballard, 5414
R. A. Bice, 7000
L. E. Lamkin, 7300
L. J. Heilman, 7400
C. R. Barncord, 7600
J. R. Meikle, 7640
G. A. Fowler, 9000
J. H. Scott, 9200
A. Y. Pope, 9300
J. W. McKiernan, 9521
W. F. Carstens, 3410
L. C. Baldwin, 3412
R. S. Gillespie, 3413
B. R. Allen, 3421
B. F. Hefley, 8232
C. H. Sproul, 3428-2 (10)