

The copyright © of this thesis belongs to its rightful author and/or other copyright owner. Copies can be accessed and downloaded for non-commercial or learning purposes without any charge and permission. The thesis cannot be reproduced or quoted as a whole without the permission from its rightful owner. No alteration or changes in format is allowed without permission from its rightful owner.



UUM
Universiti Utara Malaysia

**A FAMILY OF CLASSES IN NESTED CHAIN ABACUS AND
RELATED GENERATING FUNCTIONS**



EMAN F. MOHOMMED

UUM
Universiti Utara Malaysia

**DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2017**



Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

EMAN F. MOHOMMED

calon untuk Ijazah PhD
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"A FAMILY OF CLASSES IN NESTED CHAIN ABCUS AND RELATED GENERATING FUNCTIONS"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : **14 Disember 2017.**

That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: December 14, 2017.

Pengerusi Viva: (Chairman for VIVA)	Assoc. Prof. Dr. Maznah Mat Kasim	Tandatangan (Signature)
Pemeriksa Luar: (External Examiner)	Prof. Dr. Adem Kilicman	Tandatangan (Signature)
Pemeriksa Dalam: (Internal Examiner)	Dr. Sharmila Karim	Tandatangan (Signature)
Nama Penyelia/Penyelia-penyelia: (Name of Supervisor/Supervisors)	Prof. Dr. Haslinda Ibrahim	Tandatangan (Signature)
Nama Penyelia/Penyelia-penyelia: (Name of Supervisor/Supervisors)	Prof. Dr. Ammar Seddiq Mahmood	Tandatangan (Signature) _____
Nama Penyelia/Penyelia-penyelia: (Name of Supervisor/Supervisors)	Assoc. Prof. Dr. Nazihah Ahmad	Tandatangan (Signature)

Tarikh:
(Date) **December 14, 2017**

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:



Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Model abakus telah digunakan secara meluas untuk mewakili pemetaan bagi sebarang integer positif. Walau bagaimanapun, tiada kajian yang telah dilakukan untuk membangunkan manik abakus terkait dalam perwakilan bergraf bagi objek diskrit. Untuk mengatasi masalah keterkaitan, kajian ini tertumpu kepada pencirian n -objek terkait yang dikenali sebagai n -omino terkait, seterusnya menjana abakus rantai tersarang. Selanjutnya, sifat konsep teori bagi abakus rantai tersarang dibangunkan. Di samping itu, tiga jenis penjelmaan berbeza yang penting dalam pembinaan famili kelas turut dihasilkan. Fungsi penjana turut dirumuskan berdasarkan kelas ini dengan menggunakan pengangkaan objek kombinatorik (ECO). Dalam kaedah ECO, setiap objek diperoleh daripada objek yang lebih kecil dengan membuat pengembangan setempat. Pengembangan setempat ini dihuraikan dengan cara yang mudah melalui petua turutan. Kemudian petua turutan boleh diterjemahkan menjadi persamaan fungsian untuk fungsi penjana. Kesimpulannya, kajian ini berjaya menghasilkan perwakilan bergraf baru bagi abakus rantai tersarang yang dapat diaplikasikan dalam grid terhingga penjubinan.



UUM
Universiti Utara Malaysia

Abstract

Abacus model has been employed widely to represent partitions for any positive integer. However, no study has been carried out to develop connected beads of abacus in graphical representation for discrete objects. To resolve this connectedness problem this study is oriented in characterising n -connected objects known as n -connected dominoes, which then generate nested chain abacus. Furthermore, the theoretical conceptual properties for the nested chain abacus are being formulated. Along the construction, three different types of transformation are being created that are essential in building a family of classes. To enhance further, based on these classes, generating functions are also being formulated by employing enumeration of combinatorial objects (ECO). In ECO method, each object is obtained from smaller object by making some local expansions. These local expansions are described in a simple way by a succession rule which can be translated into a function equation for the generating function. In summary, this study has succeeded in producing novel graphical representation of nested chain abacus, which can be applied in tiling finite grid.

Keywords: abacus, partition, n -connected dominoes, generating function



UUM
Universiti Utara Malaysia

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful. Thank you Allah for it is from Your will and blessings that this research is able to be completed.

I would like to thank Professor Dr. Haslinda Ibrahim for her guidance in completing this thesis, along with Associate Professor Dr. Nazihah Ahmad who have provided clear and precise guidance on how to complete this thesis.

Lastly, I would also like to thank my parents and my son Anmar, as well as all my family members.



UUM
Universiti Utara Malaysia

Table of Contents

Permission to Use	i
Abstrak	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Tables.....	viii
List of Figures	ix
List of Appendices	xiii
List of Symbols	xiv
Declaration Associated With This Thesis	xv
CHAPTER ONE INTRODUCTION	1
1.1 Introduction	1
1.2 Graphical Representation of Partition	1
1.2.1 James Abacus	6
1.2.2 Beta Number	8
1.2.3 James Abacus Development	11
1.2.4 Advantages of James Abacus	18
1.3 n -Connected Ominoos	19
1.3.1 The Representation of n -Connected Ominoos	20
1.4 Research Motivation	25
1.5 Research Objectives	26
1.6 Scope of Study	27
1.7 Thesis Outline	27
CHAPTER TWO NESTED CHAIN ABACUS.....	30
2.1 Introduction.....	30
2.2 Definition and Terminologies	30
2.3 Nested Chain Abacus	33
2.4 The Connectedness of Beads in the Nested Chain Abacus	41

2.4.1	Connectedness of Beads with Respect to the Rows in Nested Chain Abacus	42
2.4.2	Connectedness of Beads with Respect to the Columns in Nested Chain Abacus.....	46
2.5	Design Structure of Nested Chain Abacus	50
2.5.1	Rectangular Nested Chain Abacus	53
2.5.2	Rectangle-Path Nested Chain Abacus	59
2.5.3	Singleton Nested Chain Abacus	63
2.6	Conclusion.....	70
 CHAPTER THREE NESTED CHAIN ABACUS TRANSFORMATION .		71
3.1	Introduction	71
3.2	Definition and Terminologies.....	71
3.3	Transformation in Chains.....	75
3.3.1	Transformation in Rectangle Chain	75
3.3.2	Transformation in Path Chain.....	88
3.3.3	Transformation in Singleton Chain.....	94
3.4	Nested Chain Abacus Transformation Algorithm.....	95
3.4.1	SNC2-Transformation	95
3.4.2	SNC-Transformation	97
3.4.3	MNC-Transformation.....	99
3.5	Conclusion.....	102
 CHAPTER FOUR CLASSES OF NESTED CHAIN ABACUS WITH RESPECT TO THE CHAINS		103
4.1	Introduction	103
4.2	Definitions for classes in nested chain abacus	103
4.3	Single Transformation Class	108
4.4	Stratum Transformation Class.....	112
4.5	Multi Transformation Classes	127
4.6	Generating Function with Respect to Chains.....	142
4.6.1	Succession Rule	142
4.6.2	Generating Function	145

4.7	Conclusion.....	148
CHAPTER FIVE CLASSES OF NESTED CHAIN ABACUS WITH RESPECT TO THE COLUMNS		150
5.1	Introduction.....	150
5.2	Definition and Related result.....	150
5.3	<i>e</i> -convex.....	152
5.4	ECO Method for the <i>e</i> -Convex Class.....	164
5.4.1	The Succession Rule Associated with ϑ	168
5.4.2	The Generating Function in Level N	169
5.4.3	The Generating Function of a Succession Rule.....	173
5.5	Spinal Design Approach.....	175
5.6	Conclusion.....	180
CHAPTER SIX TILING WITH NESTED CHAIN ABACUS.....		182
6.1	Introduction.....	182
6.2	Fundamental Definitions in Tiling.....	182
6.3	Tiling Algorithm.....	185
6.3.1	Algorithm with N_c	185
6.3.2	Algorithm with N_r	190
6.4	Theoretical Result.....	194
6.5	Conclusion.....	198
CHAPTER SEVEN CONCLUSION.....		199
7.1	Research contributions.....	199
7.2	Future Work.....	201
REFERENCES.....		202

List of Tables

Table 1.1	Hook length μ_{kj} of first column	5
Table 1.2	A summary of the James abacus digram development by applying several movement	17
Table 1.3	Family of n -connected ominoes and the numbers in each family for $n = 1, 2, \dots, 12$	20
Table 2.1	Placement of position numbers on the nested chain abacus with $e \times r$ positions	33
Table 2.2	Connectedness sequence of set-columns in the 16-connected beads .	44
Table 2.3	Sequence of set-columns in the 16-connected beads	48
Table 2.4	The number of chains for different values of $r = 1, 2, \dots, 12$ and $e = 1, 2, \dots, 8$	65
Table 3.1	Original and new location of bead positions where $e = 2$	87
Table 3.2	New location of the positions in the nested chain abacus by SNC2-Transformation	97
Table 3.3	New location of the positions in the nested chain abacus by SNC-Transformation	99
Table 3.4	New location of the positions in the nested chain abacus by MNC-Transformation	101
Table 3.5	New location of the positions in the nested chain abacus by MNC-Transformation	101
Table 6.1	Head column bead position of nested chain abacus with $e = 5, r = 4$ and $k = 3$	186
Table 6.2	Head column bead position of nested chain abacus with $e = 4, r = 4$ and $k = 3$	191
Table 6.3	Translation color R to color R^I	194

List of Figures

Figure 1.1	Partition $\mu = (5, 3, 3, 2, 1)$ in Ferrers diagram	2
Figure 1.2	Partition $\mu = (5, 3, 3, 2, 1)$ in Young diagram	2
Figure 1.3	The conjugate of the partition of $\mu = (5, 3, 3, 2, 1)$	3
Figure 1.4	Rim location in Young diagram where $\mu = (5, 3, 3, 2, 1)$	6
Figure 1.5	Transformation of the rim location to bead and empty bead positions in the Young diagram where $\mu = (5, 3, 3, 2, 1)$	7
Figure 1.6	James diagram when $\mu = (5, 3, 3, 2, 1)$, $e = 2$	7
Figure 1.7	Abacus	9
Figure 1.8	Abacus diagram	9
Figure 1.9	Abacus diagram for $e = 3$ and $e = 4$	9
Figure 1.10	James abacus for partitioned $\mu = (5, 3^2, 2, 1)$ when (a) $e = 2$ and (b) $e = 3$	10
Figure 1.11	Upside down	13
Figure 1.12	Right side-left	14
Figure 1.13	Direct rotation	15
Figure 1.14	Family of tetromino (4-connected ominoos)	21
Figure 1.15	A 5-connected square in different shapes	21
Figure 1.16	An 11-connected ominoos	22
Figure 1.17	An 11-connected ominoos	22
Figure 1.18	A 35-connected ominoos	23
Figure 1.19	A 26-connected ominoos with two hole	23
Figure 1.20	Twelve [3,1,2,1]-ominoes and their Gray codes	24
Figure 1.21	Young diagram of partition $\mu = (5, 4, 3^2, 1)$	25
Figure 1.22	Research Framework	29
Figure 2.1	A 7-connected ominoos	31
Figure 2.2	A 7-connected ominoos in a minimal frame	32
Figure 2.3	Direction of 7-connected ominoos	35
Figure 2.4	Nested chain abacus with 7-connected beads	36

Figure 2.5	Representation of the 4 shapes of family of tetromino (a) A 4-connected omiooes (b) A 4-connected omiooes w.r.t minimal frame (c) Nested chain abacus (d) Connected partition	40
Figure 2.6	(a) Connected partition with 4 columns and 4 rows (b) Nested chain abacus (c) 8-connected omenoos	41
Figure 2.7	Nested chain abacus of 16-connected beads with 6 columns and 3 rows.....	43
Figure 2.8	Nested chain abacus of 16-connected beads with 6 columns and 3 rows.....	45
Figure 2.9	Nested chain abacus of 16-connected beads with 6 columns and 3 rows.....	47
Figure 2.10	Conversion of the nested chain abacus in to matrix.....	52
Figure 2.11	(a) Nested chain abacus of $\mu^{(4,6)}=(8^2, 6^3, 5, 4^3, 3^2, 1^5)$ where $c = 2$ (b) Outer vertical rectangular chain (c) Inner vertical rectangular chain.....	55
Figure 2.12	(a) Nested chain abacus of $\mu^{(6,4)}=(8, 5, 7^5, 4^7, 2^3)$ where $c = 2$ (b) Outer horizontal rectangular chain (c) Inner horizontal rectangular chain.....	56
Figure 2.13	(a) The nested chain abacus where $c = 2$ (b) Outer vertical chain and (c) Vertical path chain	61
Figure 2.14	(a) Nested chain abacus where $c = 2$, (b) Outer chain and (c) Singleton chain.....	64
Figure 2.15	The structure of nested chain abacus.	70
Figure 3.1	Chain transformation' direction.....	72
Figure 3.2	(a) Full chain, (b) Arrow indicating one step full chain transformation and (c) The new location of the initial point after transformation	75
Figure 3.3	Single movement for $\mu^{(4,7)} = (10, 8^6, 5, 2, 0^6)$	77
Figure 3.4	(a) Nested chain abacus of $\mu^{(4,7)} = (10, 8^6, 5, 2, 0^6)$ with 2 rectangle chains, (b) Rectangle chain transformation applied to the outer rectangle chains and (c) Rectangle chain transformation applied to the inner rectangle chain	78
Figure 3.5	Elements for T_1	81

Figure 3.6	Elements for T_2	81
Figure 3.7	Elements for T_3	81
Figure 3.8	Elements for T_4	82
Figure 3.9	(a) Selected initial point in a rectangle chain with 9 bead positions and 3 empty bead positions, (b) Arrows indicating rectangle chain transformation and (c) The result of chain transformation	87
Figure 3.10	(a) Selected initial point in a path chain with 4 bead positions, (b) Arrows indicating x -steps path chain transformation for $x = 1$ and (c) The result of chain transformation for $x = 1$	89
Figure 3.11	(a) Nested chain abacus with one rectangle chain and one singleton chain, (b) Rectangle chain transformation applied to the outer rectangle chain and (c) Singleton chain transformation applied as the singleton chain	95
Figure 3.12	Nested chain abacus converted into matrix	96
Figure 3.13	Nested chain abacus with one chain in (a) and the result of applying Ch^1 in (b)	97
Figure 3.14	Nested chain abacus in (a) and convert into matrix in (b)	98
Figure 3.15	(a) Nested chain abacus of $\mu^{(5,4)}$ (b) Nested chain abacus of $\mu^{+3(5,4)}$	99
Figure 3.16	Nested chain abacus converted into matrix	100
Figure 3.17	(a) Nested chain abacus and (b) Nested chain abacus after employ Ch^9	102
Figure 3.18	Transformations in nested chain abacus	102
Figure 4.1	(a) Nested chain abacus for 15-connected beads and (b) The nested chain abacus for 15-connected beads embedded in square lattice .	104
Figure 4.2	(a) A D_{singl} nested chain abacus and (b) nested chain abacus but not D_{singl}	105
Figure 4.3	(a) A $D_{\text{outer}-1}$, (b) $D_{\text{outer}-2}$, (c) $D_{\text{outer}-3}$ and (d) not D_{outer}	106
Figure 4.4	Nested chain abacus of class D_{singl}	109
Figure 4.5	Fourteen of $D_{\text{outer}-2}^{(4,5)}$ for 17-connected beads where $b^I = 3$	113
Figure 4.6	Eight nested chain abacus, (a) four $D_{\text{outer}-3}^{(3,3)}$, (b) two $D_{\text{outer}-2}^{(2,3)}$ and (c) two $D_{\text{outer}-2}^{(3,2)}$	116
Figure 4.7	Eight $D_{\text{outer}-3}^{(3,4)}$ and two $D_{\text{outer}-2}^{(3,3)}$	121

Figure 4.8	Ten nested chain abacus.....	122
Figure 4.9	Ten $D_{\text{outer}-1}^{I(3,4)}$ for 8-connected beads	123
Figure 4.10	First levels of the generating tree of Ω if the first chain consists of one position.....	143
Figure 4.11	Number of nested chain abacus at level N where $N \diamond 0$	144
Figure 4.12	Nested chain abacus with respect to chains	149
Figure 5.1	(a) Nested chain abacus of e -core connected partition and (b) Nested chain abacus of connected partition	153
Figure 5.2	(a) A e -convex with 2 columns and 4 rows, (b) e -convex of $k = 0$, (c) e -convex of $k = 1$, (d) e -convex of $k = 2$, (e) e -convex of $k = 3$, (f) e -convex of $k = 4$ and (g) e -convex of $k = 5$	160
Figure 5.3	(a) A e^V -convex, (b) e^{PV} -convex, (c) e^{PH} -convex, (d) e^H -convex and (f) e^S -convex	165
Figure 5.4	The \mathfrak{S} operator applied to N nested chain abacus in class e^S -convex	167
Figure 5.5	\mathfrak{S} operator applied to e^{PH} -convex nested chain abacus	168
Figure 5.6	Generating tree of Ω	169
Figure 5.7	The 7 distinct forms of N^2 for 4-connected squares	176
Figure 5.8	177
Figure 5.9	Methods to development classes in nested chain abacus and generating function	181
Figure 6.1	(a) Nested chain abacus for 15-connected beads and (b) The nested chain abacus for 15-connected beads embedded in a finite grid . .	183
Figure 6.2	Head column beads and head row beads in nested chain abacus . .	184
Figure 6.3	(a) Nested chain abacus where $e = 4, r = 4, k = 3$ (b) Γ_1 where $s = 1$ (c) Γ_1 where $s = 1, 2$ (d) Γ_1 where $s = 1, 2$ and Γ_2 where $s = 0$ and $S^I = 1$ (e) Γ_1 where $s = 1, 2$ and Γ_2 where $s = 0, 1$ and $S^I = 1, 1$	189
Figure 6.4	(a) Nested chain abacus where $e = 4, r = 4, k = 3$ (b) τ_1 where $s = 1$ (c) τ_1 where $s = 1, 2$ (d) τ_2 where $s = 1, 2$ and τ_2 where $s = 0$ and $S^I = 1$ (e) τ_1 where $s = 1, 2$ and τ_2 where $s = 0, 1$ and $S^I = 1, 1$	193
Figure 6.5	Tiling with nested chain abacus	198

List of Appendices

Appendix A	Generating Function W.R.T Chains.....	206
Appendix B	Tiling Algorithm w.r.t Row	208
Appendix C	Tiling Algorithm w.r.t Column.....	215
Appendix D	Generating N_c and N_r Nested Chain Abacus	223
Appendix E	Chain Transformation.....	238
Appendix F	Generating Function	247



UUM
Universiti Utara Malaysia

List of Symbols

$\mu^{(e,r)}$	Connected Partition with e Columns and r Rows
\mathfrak{N}	Nested Chin Abacus
SR	Set-Row
SC	Set-Column
P_{ρ}^{Rec}	Sequence of Rectangular Nested Chain Abacus
P_{ρ}^{Rec-h}	Sequence of Rectangle Path Nested Chain Abacus
$SNC2$	Single Nested Chain Abacus Transformation
SNC	Singular Nested Chain Abacus Transformation
MNC	Multiple nested Chain Abacus
\mathcal{D}_{single}	Singular Transformation Class
\mathcal{D}_{outer}	Single Transformation Class, if $i = 1$
$\mathcal{D}_{inner-i}$	Single Transformation Class, if $i > 1$
\mathcal{D}_{inner}	Multi Transformation Classes
b_i	Number of Beads Positions in Chain i
b'_i	Number of Empty Bead Positions in Chain i
\mathfrak{S}^2	Classes of Nested Chain Abacus with two Columns
\mathfrak{N}_c	Class of Nested Chin Abacus w.r.t columns
\mathfrak{N}_r	Class of Nested Chin Abacus w.r.t rows

Declaration Associated With This Thesis

Journal

1. Mohommed, E. F., Ahmad, N., Ibrahim, H. (2016). Intersection of Main James Abacus Diagram for the Outer Chain Movement with Length $[1, 0, 0\dots]$. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(8), 51-56 (scopus).
2. Mohommed, E. F., Ahmad, N., Ibrahim, H., Mahmood, A. S. (2016). Nested Chain Movement of length 1 of Beta Number in James Abacus Diagram. Global Journal of Pure and Applied Mathematics, 12(4), 2953-2969 (scopus).
3. Mohommed, E. F., Ibrahim, H. Ahmad, N. (2017). Enumeration of n-connected objects inscribed in an abacus. Journal of Algebra, Number Theory Applications, (scopus).
4. Mohommed, E. F., Ahmad, N., Ibrahim, H. (2017). Enumeration Class of Polyminoes Defined by Two Column. IOSR Journal of Mathematics (IOSR-JM), 13(1), 44-49 .

proceeding

1. Mohommed, E. F., Ibrahim, H., Mahmood, A. S., Ahmad, N. (2015, December). Embedding chain movement in James diagram for partitioning beta number. In AIP Conference Proceedings (Vol. 1691, No. 1, p. 040019). AIP Publishing.
2. Mohommed, E. F., Ibrahim, H., Ahmad, N., Mahmood, A. (2016, August). Embedding the outer chain movement for main partition of β -number with length $[1, 0, 0, \dots]$. In AIP Conference Proceedings (Vol. 1761, No. 1, p. 020076). AIP Publishing.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The theory of partition is a fundamental area of number theory, it is concerning the representation of integer as sum of other integers. The theory of partition has been applied in many different areas such as combinatorics, statistical and particle physic. The partitions can be graphically represented with diagrams such as Ferrers diagram and Young diagram. Agraphical representation of partition is important in the partition theory because it can design and facilitate a visual structure of any shape in the form of discrete object. Henceforth, this thesis focuses on the use of graphical illustration of partition to develop a new design structure of connected ominoes. The beauty of this construction is further extended to be used in tiling fnite grid.

1.2 Graphical Representation of Partition

Diagrams are used to represent a partition of any positive integer. Since 1800s, the famous diagrams are the Ferrers diagram and the Young diagram (Benjamin & Quinn, 2003; Hardy & Wright, 1979). On the other hand, a James diagram or known as e -abacus uses a β -number to represent a sequence of non-decreasing integer numbers (Gyoja et al., 2010). Next, the concept of partition and graphical representation of the partition are reviewed.

Definition 1.2.1. (Andrews, 1998) *A partition of a positive integer, t , is a finite non-increasing sequence of non-negative integers $(\mu_1, \mu_2, \dots, \mu_n)$ such that $\sum_{i=1}^n \mu_i = t$ and n is the number of parts of any partition.*

Example 1.2.2. $(5, 3, 3, 2, 1), (5, 5, 2, 2), (6, 4, 2, 1, 1), \dots$ are partitions of $t = 14$.

If $\mu = (5, 3, 3, 2, 1)$, then $n = 5$.

The contents of
the thesis is for
internal user
only

REFERENCES

- Abramovich, S. (2012). Partitions of integers, ferrers-young diagrams, and representational efficacy of spreadsheet modeling. *Spreadsheets in Education (eJSiE)*, 5(2), 1-27.
- Andrews, G. (1998). *The theory of partitions* (No. 2). Cambridge University Press.
- Apostol, T. M. (2013). *Introduction to analytic number theory*. Springer Science & Business Media.
- Aval, C., DAdderio, M., Dukes, M., Hicks, A., & Le Borgne, Y. (2014). Statistics on parallelogram polyominoes and aq, t-analogue of the narayana numbers. *Journal of Combinatorial Theory, Series A*, 123(1), 271–286.
- Bacchelli, S., Ferrari, L., Pinzani, R., & Sprugnoli, R. (2010). Mixed succession rules: The commutative case. *Journal of Combinatorial Theory, Series A*, 117(5), 568–582.
- Barcucci, E., Bertoli, F., Del Lungo, A., & Pinzani, R. (1997). The average height of directed column-convex polyominoes having square, hexagonal and triangular cells. *Mathematical and Computer Modelling*, 26(8-10), 27–36.
- Barcucci, E., Frosini, A., & Rinaldi, S. (2005). On directed-convex polyominoes in a rectangle. *Discrete mathematics*, 298(1), 62–78.
- Barcucci, E., Lungo, A., Pergola, E., & Pinzani, R. (1999). Eco: a methodology for the enumeration of combinatorial objects. *Journal of Difference Equations and Applications*, 5(4-5), 435–490.
- Barcucci, E., Lungo, A., Pinzani, R., & Sprugnoli, R. (1996). Polyominoes defined by their vertical and horizontal projections. *Journal of Theoretical Computer Science*, 25(7), 129-136.
- Barequet, G., Rote, G., & Shalah, M. (2016). $\lambda > 4$: An improved lower bound on the growth constant of polyominoes. *Communications of the ACM*, 59(7), 88–95.
- Beauquier, D., & Nivat, M. (1990). Tiling the plane with one tile. In *Proceedings of the sixth annual symposium on computational geometry* (pp. 128–138).
- Beauquier, D., Nivat, M., Rémila, E., & Robson, M. (1995). Tiling figures of the plane with two bars. *Computational Geometry*, 5(1), 1–25.
- Bender, E. (1974). Convex n -ominoes. *Discrete Mathematics*, 8(3), 219–226.
- Benjamin, A., & Quinn, J. (2003). *Proofs that really count: the art of combinatorial proof*. Mathematical Association of America, Washington, D.C.
- Berlekamp, E., Conway, J., & Guy, R. (2003). *Winning ways for your mathematical plays* (Vol. 3). AK Peters Natick.
- Berstel, J. (1985). *Perrin, theory of codes*. Academic Press, New York.

- Carroll, L. (1867). *An elementary treatise on determinants: With their application to simultaneous linear equations and algebraical geometry*. Macmillan.
- Castiglione, G., Frosini, A., Restivo, A., & Rinaldi, S. (2005). Enumeration of l -convex polyominoes by rows and columns. *Theoretical Computer Science*, 347(1-2), 336–352.
- Castiglione, G., & Restivo, A. (2003). Reconstruction of l -convex polyominoes. *Electronic Notes in Discrete Mathematics*, 12, 290–301.
- Chow, S., & Ruskey, F. (2009). Gray codes for column-convex polyominoes and a new class of distributive lattices. *Discrete Mathematics*, 309(17), 5284–5297.
- Cipra, B. A. (1987). An introduction to the ising model. *American Mathematical Monthly*, 94(10), 937–959.
- De Hoyos, I. (1990). Points of continuity of the kronecker canonical form. *SIAM Journal on Matrix Analysis and Applications*, 11(2), 278–300.
- Del Lungo, A., Duchi, E., Frosini, A., & Rinaldi, S. (2004). On the generation and enumeration of some classes of convex polyominoes. *The Electronic Journal of Combinatorics*, 11(1), R60.
- Duchi, E. (2003). *Eco method and object grammars: two methods for the enumeration of combinatorial objects* (Unpublished doctoral dissertation). Universit' a Degli Studi di Firenze.
- Duchi, E., Rinaldi, S., & Schaeffer, G. (2008). The number of z -convex polyominoes. *Advances in Applied Mathematics*, 40(1), 54–72.
- Fayers, M. (2007). Another runner removal theorem for ν -decomposition numbers of iwahori–hecke algebras and q -schur algebras. *Journal of Algebra*, 310(1), 396–404.
- Fayers, M. (2008). Decomposition numbers for weight three blocks of symmetric groups and iwahori–hecke algebras. *Transactions of the American Mathematical Society*, 360(3), 1341–1376.
- Fayers, M. (2009). General runner removal and the mullineux map. *Journal of Algebra*, 322(12), 4331–4367.
- Fayers, M. (2010). On the irreducible representations of the alternating group which remain irreducible in characteristic. *Representation Theory of the American Mathematical Society*, 14(16), 601–626.
- Ferrari, L., Pergola, E., Pinzani, R., & Rinaldi, S. (2003). Jumping succession rules and their generating functions. *Discrete Mathematics*, 271(1), 29–50.
- Fulton, W. (1997). *Young tableaux: with applications to representation theory and geometry* (Vol. 35). Cambridge University Press.
- Gao, W., & Wang, W. (2014). Second atom-bond connectivity index of special chemical molecular structures. *Journal of Chemistry*, 2014, 2–8.

- Golomb, S. (1954). Checker boards and polyominoes. *The American Mathematical Monthly*, 61(10), 675–682.
- Goulden, I., & Jackson, D. (2004). *Combinatorial enumeration*. Courier Corporation.
- Goupil, A., Cloutier, H., & Nouboud, F. (2010). Enumeration of polyominoes inscribed in a rectangle. *Discrete Applied Mathematics*, 158(18), 2014–2023.
- Goupil, A., Cloutier, H., & Pellerin, M. (2013). Generating functions for inscribed polyominoes. *Discrete Applied Mathematics*, 161(1), 151–166.
- Guttmann, A., & Enting, I. (1988). The number of convex polygons on the square and honeycomb lattices. *Journal of Physics A: Mathematical and General*, 21(8), L467.
- Gyoja, A., Nakajima, H., Shinoda, K., Shoji, T., & Tanisaki, T. (2010). *Representation theory of algebraic groups and quantum groups* (Vol. 284). Springer Science & Business Media.
- Hardy, G., & Wright, E. (1979). *An introduction to the theory of numbers*. Oxford University Press.
- James, G. (1978). Some combinatorial results involving young diagrams. In *Mathematical proceedings of the cambridge philosophical society* (Vol. 83, pp. 1–10).
- James, G. (1987). *The representation theory of the symmetric groups*. Berlin.
- James, G., Lyle, S., & Mathas, A. (2006). Rouquier blocks. *Mathematische Zeitschrift*, 252(3), 511–531.
- King, O. (2014). *The representation theory of diagram algebras* (Unpublished doctoral dissertation). City University London.
- Klarner, A. (1966). *Enumeration involving sums over compositions* (Unpublished doctoral dissertation). University of Alberta, Edmonton.
- Littlewood, D. (1951). Modular representations of symmetric groups. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 209(1098), 333–353.
- Loehr, N. (2010). Abacus proofs of schur function identities. *SIAM Journal on Discrete Mathematics*, 24(4), 1356–1370.
- Loehr, N. (2011). *Bijjective combinatorics*. CRC Press.
- Mahmood, A. (2011). On the intersection of youngs diagrams core. *Journal of Education and Science (Mosul Univ.)*, 24(3), 143–159.
- Mahmood, A. (2013). Upside-down β -numbers. *Australian Journal of Basic & Applied Sciences*, 7(7), 36–46.
- Mahmood, A., & Ali, S. (2013a). Direct rotation β -numbers. *Advances in Mathematic Science*, 15(2), 642–649.
- Mahmood, A., & Ali, S. (2013b). Rightside-left β -numbers. *International Journal of Latest Research in Science and Technology*, 2(6), 124–127.

- Mahmood, A., & Ali, S. (2013c). Rightside-left o direct rotation β numbers. *International Journal of Modern Sciences and Engineering Technology*, 1(6), 36-46.
- Martínez, C., & Molinero, X. (2001). A generic approach for the unranking of labeled combinatorial classes. *Random Structures & Algorithms*, 19(3-4), 472–497.
- Mathas, A. (1999). *Iwahori-hecke algebras and schur algebras of the symmetric group* (Vol. 15). American Mathematical Soc.
- Mohammad, H. (2008). *Algorithms of the core of algebraic youngs tableaux* (Unpublished master's thesis). Collage of Education. University of Mosul.
- Pergola, E. (1999). *Eco : a methodology for enumerating combinatorial objects* (Unpublished doctoral dissertation). University of Florence.
- Rechnitzer, A. (2001). *Some problems in the counting of lattice animals, polyominoes, polygons and walks*. University of Melbourne, Department of Mathematics and Statistics.
- Redelmeier, H. (1981). Counting polyominoes: yet another attack. *Discrete Mathematics*, 36(2), 191–203.
- Sami, H. (2014). *On the main diagram of exchanging rows* (Unpublished master's thesis). Mosul University.
- Stanton, D., & White, D. (1986). *Constructive combinatorics*. Springer Science & Business Media.
- Surhone, L., Timpledon, M., & Marseken, S. (2010). *Polyomino: Square tiling, polyiamond, polyhex, tromino, tetromino, pentomino, hexomino, heptomino, nonomino*. Betascript Publishing. Retrieved from <https://books.google.com.my/books?id=Ffo2QwAACAAJ>
- Tingley, P. (2008). Three combinatorial models for sln crystals, with applications to cylindric plane partitions. *International Mathematics Research Notices*, 2008.
- Wildon, M. (2008). Counting partitions on the abacus. *The Ramanujan Journal*, 17(3), 355-367.
- Wildon, M. (2014). A short proof of a plethystic murnaghan–nakayama rule. *arXiv preprint arXiv:1408.3554*.
- Young, A. (1934). On quantitative substitutional analysis. *Proceedings of the London Mathematical Society*, 2(1), 304–368.

APPENDIX A

GENERATING FUNCTION W.R.T CHAINS

Input Rows and columns

$r = \text{input}(\text{'Input the number of rows: '});$

$e = \text{input}(\text{'Input the number of columns: '});$

* Classify cases

* Case 1 (If $r < e$ and r is odd, then $p_1 = e - r + 1$ $p_2 = 2p_1 + 6$)

if $r < e \text{ mod}(r, 2) == 1$

$P_1 = e - r + 1;$

$P_2 = 2 * P_1 + 6;$

end * Case 2 (If $e < r$ and e is odd, then $p_1 = r - e + 1$ and $p_2 = 2 p_1 + 6$)

if $e < r \text{ mod}(e, 2) == 1$

$P_1 = r - e + 1;$

$P_2 = 2 * P_1 + 6;$

end * Case 3 (If $e < r$ and e is even, then $p_1 = r - e + 1$ and $p_2 = 2 p_1 + 6$)

if $e == r \text{ mod}(r, 2) == 1 \text{ mod}(e, 2) == 1$

$P_1 = 1;$

$P_2 = 8;$

end

* Case 4 (If $r < e$ and r is even then $p_1 = 2r + 2e - 4 (2c - 1) = 2r - 2e + 4$, where $c = r/2$ and

$p_2 = p_1 + 8.$)

if $r < e \text{ mod}(r, 2) == 0$

$P_1 = 2 * r - 2 * e + 4;$

$P_2 = P_1 + 8;$

end

* Case 5 (If $e < r$ and e is even then $p_1 = 2r + 2e - 4 (2c - 1) = 2e - 2r + 4$, where $c = e/2$

and $p_2 = p_1 + 8$)

```
if e<=r mod(e,2)==0
```

```
P1=2*e-2*r+4
```

```
P2=P1+8
```

```
end
```

```
** Compute the generating function syms x
```

```
syms y
```

```
* f(x,y) fprintf('====f(x,y)====')
```

```
f(x,y)=-exp(1/(8*x*y8)) * int(yP2 - 9 * exp(-1/8 * x*y8), y)
```

```
· Generating function f(x)
```

```
f print f('==== Generating function f(x) =====
```

```
)
```

```
f(x, 1)
```

```
· PolynomialFrom
```

```
ff(1) = 1; ff(2) = P2;
```

```
forn = 3 : 10
```

```
ff(n) = ff(n-1) * (P2 + 8);
```

```
end
```

```
* Writethisresult
```

```
fori = 1 : 10
```

```
f print f('f(*d) = *di, i, ff(i)
```

```
end
```

APPENDIX B

TILLING ALGORITHM W.R.T ROW

```

thetac,ol = 25;
thetar,ow = 25;
H1 = 3;
r = 5;
e = 5;
Computetheupperofs, sI, sII
slim = (ceil(thetar,ow/r)) - 1;
s1lim = (ceil((thetac,ol - e)/H1)) + 3;
s2lim = ceil(thetac,ol/H1);
Setcolors
color2 = [255, 217, 102]/255;
color1 = [131, 59, 10]/255;
• Generatinginitialabacus
*temp = ceil(rand(1, e) * (r - numb,ead + 1));
temp = [2, 3, 2, 1, 2];
rect = zeros(thetar,ow, thetac,ol);
fori = 1 : r
rect(i, temp(i) : temp(i) + H1 - 1) = 1;
end
Drawshape(rect, thetar,ow, thetac,ol, color1, color2)
title('InitialRectangle')
H = max(temp);
L = temp(1, 1);
L1 = temp(1, end);
p = H1;

```

$p1 = L1 - L;$

*· If $L1 > L$, then apply the mapping 1 and mapping 2, mapping 3
if $L1 > L$*

form = 1 : r

for j = 1 : e

for s = 1 : sim

*if $rect(m, j) == 1$ and $(j + s * p1) > 0$ and $(j + s * p1) \leq theta_{col}$ and $mod(s, 2) == 1$*

*$rect(m + s * r, j + s * p1) = 0;$*

*else if $rect(m, j) == 0$ and $(j + s * p1) > 0$ and $(j + s * p1) \leq theta_{col}$ and*

$mod(s, 2) == 1$

*$rect(m + s * r, j + s * p1) = 1;$*

*else if $rect(m, j) == 1$ and $(j + s * p1) > 0$ and $(j + s * p1) \leq theta_{col}$ and*

$mod(s, 2) == 0$

*$rect(m + s * r, j + s * p1) = 1;$*

*else if $rect(m, j) == 0$ and $(j + s * p1) > 0$ and $(j + s * p1) \leq theta_{col}$ and*

$mod(s, 2) == 0$

*$rect(m + s * r, j + s * p1) = 0;$*

end

end

end

end

Draw_{shape}($rect, theta_{row}, theta_{col}, color1, color2$)

title(¹Rectangle after Mapping¹)

form = 1 : r

*$rect(m + s * r, j + s * p1 + s1 * p) = 1;$*

*else if $rect(m, j) == 0$ and $(m + s * p1 + s1 * p) > 0$ and*

*$(m + s * p1 + s1 * p) \leq theta_{col}$ and $mod(s, 2) == 1$ and $mod(s1, 2) == 1$*

```

rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
end

if L1 > L
form = 1 : r
for j = 1 : e
for s = 1 : sim
if rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 1
rect(m + s * r, j + s * p1) = 0;
elseif rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 1
rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 0;
end
end
end
end
end

```

end

end

Draw_shape(rect, theta_row, theta_col, color1, color2)

title('RectangleafterMapping1')

form = 1 : r

for j = H : e

for s = 0 : sim

for s1 = 1 : s1im

*if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col*

and mod(s, 2) == 0 and mod(s1, 2) == 1

*rect(m + s * r, j + s * p1 + s1 * p) = 0;*

*elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 1*

*rect(m + s * r, j + s * p1 + s1 * p) = 1;*

*elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 0*

*rect(m + s * r, j + s * p1 + s1 * p) = 1;*

*elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <*

0 and mod(s1, 2) == 0

*rect(m + s * r, j + s * p1 + s1 * p) = 0;*

*elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0*

*rect(m + s * r, j + s * p1 + s1 * p) = 0;*

*elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0*

*rect(m + s * r, j + s * p1 + s1 * p) = 1;*

*elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and*

```

(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
form = 1 : r
for j = 1 : H1
for s = 1 : s1_1im
for s2 = 1 : s2_1im
for j = H : e
for s = 0 : s_1im
for s1 = 1 : s1_1im
if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col
and mod(s, 2) == 0 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 0

```

```

rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <
1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 21')

form = 1 : r
for j = 1 : H1
for s = 1 : s1_im
for s2 = 1 : s2_im

```



```

function DrawShape(M, a, b, color1, color2)
figure
axis([0 b 0 a])
hold on
for i = 1 : a
    for j = 1 : b
        if M(i, j) == 1
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color1);
        else if M(i, j) == 0
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color2);
        end
    end
end
end
end
end
end

```



UUM
 Universiti Utara Malaysia

APPENDIX C

TILLING ALGORITHM W.R.T COLUMN

```

Initial conditions  $\theta_{col} = 30$ ;
 $\theta_{ow} = 30$ ;
 $d1 = 3$ ;
 $r = 5$ ;
 $e = 5$ ;
 $s1_{im} = \lceil \theta_{col}/e \rceil - 1$ ;
 $s_{im} = \lceil (r - \theta_{ow})/d1 \rceil$ ;
 $s2_{im} = \lceil \theta_{col}/e \rceil - 1$ ;
 $color1 = [56, 85, 34]/255$ ;
 $color2 = [156, 194, 228]/255$ ;
 $temp = [2, 1, 2, 3, 2]$ ;
 $rect = \text{zeros}(\theta_{ow}, \theta_{col})$ ;
for  $i = 1 : e$ 
 $rect(temp(i) : temp(i) + d1 - 1, i) = 1$ ;
end

Drawshape( $rect, \theta_{ow}, \theta_{col}, color1, color2$ )
title('InitialRectangle')

 $k = \max(temp)$ ;
 $k1 = temp(1, 1)$ ;
 $k2 = temp(1, end)$ ;

 $p = d1$ ;
 $p1 = k2 - k1$ ;
if  $k2 \leq k1$ 
 $form = 1 : r$ 
for  $j = 1 : e$ 

```

```

for s1 = 1 : s1_lim
    if rect(m, j) == 1 and (m + s1 * p1) > 0 and
        (m + s1 * p1) <= theta_ow and mod(s1, 2) == 1
        rect(m + s1 * p1, j + s1 * e) = 0;
    else if rect(m, j) == 0 and (m + s1 * p1) > 0
        and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 1
        rect(m + s1 * p1, j + s1 * e) = 1;
    else if rect(m, j) == 1 and (m + s1 * p1) > 0
        and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0
        rect(m + s1 * p1, j + s1 * e) = 1;
    else if rect(m, j) == 0 and (m + s1 * p1) > 0
        and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0
        rect(m + s1 * p1, j + s1 * e) = 0;
    end
end
end
end
end

```

Draw_shape(rect, theta_ow, theta_col, color1, color2)

```

form = r - p + 1 : r
for j = 1 : e
    for s = 0 : 6
        for s1 = 1 : 12
            if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0
                and (m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and
                    mod(s1, 2) == 1
                rect(m + s * p1 + s1 * p, j + s * e) = 0;
            else if rect(m, j) == 0 and

```

$(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 0 \text{ and } \text{mod}(s1, 2) == 1$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$
 $\text{else if } \text{rect}(m, j) == 1 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 0 \text{ and } \text{mod}(s1, 2) == 0$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$
 $\text{else if } \text{rect}(m, j) == 0 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 0 \text{ and } \text{mod}(s1, 2) == 0$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 0;$
 $\text{else if } \text{rect}(m, j) == 1 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 1 \text{ and } \text{mod}(s1, 2) == 0$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 0;$
 $\text{else if } \text{rect}(m, j) == 0 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 1 \text{ and } \text{mod}(s1, 2) == 0$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$
 $\text{else if } \text{rect}(m, j) == 1 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 1 \text{ and } \text{mod}(s1, 2) == 1$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$
 $\text{else if } \text{rect}(m, j) == 0 \text{ and}$
 $(m + s * p1 + s1 * p) > 0 \text{ and } (m + s * p1 + s1 * p) \leq \theta$
 $\text{mod}(s, 2) == 1 \text{ and } \text{mod}(s1, 2) == 1$
 $\text{rect}(m + s * p1 + s1 * p, j + s * e) = 0;$

end

end

end

end

end

Draw_shape(rect, theta, ow, theta_col, color1, color2)

title('Rectangle after Mapping 2')

end

if k2 > k1

form = 1 : r

for j = 1 : e

for s1 = 1 : s1_lim

*if rect(m, j) == 1 and (m + s1 * p1) > 0 and (m + s1 * p1) <= theta_ow*

and mod(s1, 2) == 1

*rect(m + s1 * p1, j + s1 * e) = 0;*

*elseif rect(m, j) == 0 and (m + s1 * p1) > 0*

*and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 1*

*rect(m + s1 * p1, j + s1 * e) = 1;*

*elseif rect(m, j) == 1 and (m + s1 * p1) > 0*

*and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0*

*rect(m + s1 * p1, j + s1 * e) = 1;*

*elseif rect(m, j) == 0 and (m + s1 * p1) > 0*

*and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0*

*rect(m + s1 * p1, j + s1 * e) = 0;*

end

end

end

end

Draw_sshape(rect, theta_row, theta_col, color1, color2)

form = r - p + 1 : r

for j = 1 : e

fors = 0 : 6

fors1 = 1 : 12

*if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and*

mod(s1, 2) == 1

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 0 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 1

*rect(m + s * p1 + s1 * p, j + s * e) = 1;*

elseif rect(m, j) == 1 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 1;*

elseif rect(m, j) == 0 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 1 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 1 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 0 and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{col}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 0$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$

else if $\text{rect}(m, j) == 1$ and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{col}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 1$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$

else if $\text{rect}(m, j) == 0$ and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{col}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 1$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 0;$

end

end

end

end

end

$\text{Draw}_{\text{shape}}(\text{rect}, \theta_{row}, \theta_{col}, \text{color1}, \text{color2})$

$\text{title}(\text{Rectangle after Mapping } 2^J)$

$\text{form} = 1 : d1$

for $j = 1 : e$

for $s = 0 : 6$

for $s2 = 1 : 12$

if $\text{rect}(m, j) == 1$ and $(m + s * p1 - s2 * p) > 0$ and $(m + s * p1 - s2 * p) \leq \theta_{col}$ and

$\text{mod}(s, 2) == 0$ and $\text{mod}(s2, 2) == 1$ $\text{rect}(m + s * p1 - s2 * p, j + s * e) = 0;$

else if $\text{rect}(m, j) == 0$ and $(m + s * p1 - s2 * p) > 0$ and

$(m + s * p1 - s2 * p) \leq \theta_{col}$ and $\text{mod}(s, 2) == 0$ and $\text{mod}(s2, 2) == 1$

$\text{rect}(m + s * p1 - s2 * p, j + s * e) = 1;$



```

else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 0 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 0 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 0;
else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 0;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 1
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 1
rect(m + s * p1 - s2 * p, j + s * e) = 0;
end
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
end

function Draw_shape(M, a, b, color1, color2)

```



```

figure
axis([0b0a])
hold on

for i = 1 : a
    for j = 1 : b
        if M(i, j) == 1
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color1);
        else if M(i, j) == 0
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color2);
        end
    end
end
end
end
end
end

```



UUM
 Universiti Utara Malaysia

APPENDIX D

GENERATING N_C AND N_R NESTED CHAIN ABACUS

Code A

```
r=input('Enter a number of rows:');
e=input('Enter a number of columns:');

theta_ow = input('Enter a number of columnsof onutputrectangle(r < theta_ow) :');
theta_col = input('Enter a number of columnsof onutputrectangle(e < theta_col) :');
d1 = input('Enter the same number of bead position :');
con = Find_connected_abacus(e, r-d1+1, r, e, d1, 'col'); num_initialrect = size(con, 1);
if num_initialrect == 0
error('Error : There is no connected abacus for these parameters. ');
end
temp = con(randi([1, num_initialrect], 1), :)
rect = zeros(theta_ow, theta_col);
for i = 1 : e
rect(temp(i)/2 + 1 : temp(i)/2 + d1, i) = 1;
end
s_lim = ceil(theta_ow/r) - 1;
s1_lim = ceil(theta_col/e);
s2_lim = ceil(theta_col/e);
s_color1 = [56, 85, 34]/255;
s_color2 = [156, 194, 228]/255;
k = min(temp);
k1 = temp(1, 1)
k2 = temp(1, end)
p = d1;
p1 = k2 - k1;
```

```

rect = zeros(theta_ow, theta_ol);
for i = 1 : e
rect(temp(i) : temp(i) + d1 - 1, i) = 1;
end

Draw_shape(rect, theta_ow, theta_ol, color1, color2)
title('InitialRectangle')

```

```

if k2 > k1
for m = 1 : r
for j = 1 : e
for s1 = 1 : s1_max
if rect(m, j) == 1 and (m + s1 * p1) > 0 and (m + s1 * p1) <= theta_ow
and mod(s1, 2) == 1
rect(m + s1 * p1, j + s1 * e) = 0;
else if rect(m, j) == 0 and (m + s1 * p1) > 0
and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 1
rect(m + s1 * p1, j + s1 * e) = 1;
else if rect(m, j) == 1 and (m + s1 * p1) > 0
and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0
rect(m + s1 * p1, j + s1 * e) = 1;
else if rect(m, j) == 0 and (m + s1 * p1) > 0
and (m + s1 * p1) <= theta_ow and mod(s1, 2) == 0
rect(m + s1 * p1, j + s1 * e) = 0;
end
end
end

```

end

Draw_sshape(rect, theta_row, theta_col, color1, color2)

form = r - p + 1 : r

for j = 1 : e

for s = 0 : 6

for s1 = 1 : 12

*if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and*

*(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and*

mod(s1, 2) == 1

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 0 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 1

*rect(m + s * p1 + s1 * p, j + s * e) = 1;*

elseif rect(m, j) == 1 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 1;*

elseif rect(m, j) == 0 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 0 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 1 and

*(m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col and*

mod(s, 2) == 1 and mod(s1, 2) == 0

*rect(m + s * p1 + s1 * p, j + s * e) = 0;*

elseif rect(m, j) == 0 and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{ol}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 0$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$

elseif $\text{rect}(m, j) == 1$ and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{ol}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 1$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 1;$

elseif $\text{rect}(m, j) == 0$ and

$(m + s * p1 + s1 * p) > 0$ and $(m + s * p1 + s1 * p) \leq \theta_{ol}$ and

$\text{mod}(s, 2) == 1$ and $\text{mod}(s1, 2) == 1$

$\text{rect}(m + s * p1 + s1 * p, j + s * e) = 0;$

end

end

end

end

end

$\text{Draw}_{\text{shape}}(\text{rect}, \theta_{ow}, \theta_{ol}, \text{color1}, \text{color2})$

$\text{title}(\text{'Rectangle after Mapping 2'})$

$\text{form} = 1 : d1$

for $j = 1 : e$

for $s = 0 : 6$

for $s2 = 1 : 12$

if $\text{rect}(m, j) == 1$ and $(m + s * p1 - s2 * p) > 0$ and $(m + s * p1 - s2 * p) \leq \theta_{ol}$ and

$\text{mod}(s, 2) == 0$ and $\text{mod}(s2, 2) == 1$ $\text{rect}(m + s * p1 - s2 * p, j + s * e) = 0;$

elseif $\text{rect}(m, j) == 0$ and $(m + s * p1 - s2 * p) > 0$ and

$(m + s * p1 - s2 * p) \leq \theta_{ol}$ and $\text{mod}(s, 2) == 0$ and $\text{mod}(s2, 2) == 1$

$\text{rect}(m + s * p1 - s2 * p, j + s * e) = 1;$



```

else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 0 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 0 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 0;
else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 0;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 0
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 1 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 1
rect(m + s * p1 - s2 * p, j + s * e) = 1;
else if rect(m, j) == 0 and (m + s * p1 - s2 * p) > 0 and
(m + s * p1 - s2 * p) <= theta_col and mod(s, 2) == 1 and mod(s2, 2) == 1
rect(m + s * p1 - s2 * p, j + s * e) = 0;
end
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
end

function Draw_shape(M, a, b, color1, color2)

```

```

figure
axis([0b0a])
hold on

for i = 1 : a
    for j = 1 : b
        if M(i, j) == 1
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color1);
        elseif M(i, j) == 0
            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color2);
        end
    end
end
end
end
end
end

```



UUM
Universiti Utara Malaysia

Code B

```

r=input('Enter a number of rows:');
e=input('Enter a number of columns:');

theta_ow = input('Enter a number of columnsof outputrectangle(r < theta_ow) :');
theta_ol = input('Enter a number of columnsof outputrectangle(e < theta_ol) :');
H1 = input('Enter the same number of bead position :');
con = Find_connected_bacus(r, e - H1 + 1, r, e, H1, 'row');
num_initialrect = size(con, 1);
if num_initialrect == 0
    error('Error : There is no connected abacus for these parameters.')
end

```

```

temp = con(randi([1, num, nitialrect], 1), :);
rect = zeros(theta_ow, theta_ol);
for i = 1 : r
rect(i, temp(i)/2 + 1 : temp(i)/2 + H1) = 1;
end

s1im = ceil(theta_ow/r) - 1;
s1im = ceil(theta_ol/e);
s2im = ceil(theta_ol/e);

s_color1 = [131, 59, 10]/255;
s_color2 = [255, 217, 102]/255;

H = min(temp);
L = temp(1, 1);
L1 = temp(1, end);
p = e - H + 1;
p1 = L - L1;
figure
Draw_shape(rect, s_color1, s_color2)
title('Initial Rectangle')

• If L1 <= L, applicatethemapping1 and mpping1
if L1 > L

form = 1 : r
for j = 1 : e
for s = 1 : s1im
if rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 1 rect(m + s * r, j + s * p1) = 0;
else if rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 1

```



```

rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_col and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 0;
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 1')
form = 1 : r
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
end
if L1 > L

```

```

form = 1 : r
for j = 1 : e
for s = 1 : sim
if rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 1
rect(m + s * r, j + s * p1) = 0;
elseif rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 1
rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 1 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 1;
elseif rect(m, j) == 0 and (j + s * p1) > 0 and (j + s * p1) <= theta_ol and
mod(s, 2) == 0
rect(m + s * r, j + s * p1) = 0;
end
end
end
end

Draw_shape(rect, theta_ow, theta_ol, color1, color2)
title('Rectangle after Mapping 1')

form = 1 : r
for j = H : e
for s = 0 : sim
for s1 = 1 : s1_sim
if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_ol
and mod(s, 2) == 0 and mod(s1, 2) == 1

```

```

rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 0 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 0 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 0 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c0 and mod(s, 2) == 1 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end
end

```

```

Drawshape(rect,theta,row,theta_col,color1,color2)
title('Rectangle after Mapping2')

form = 1 : r
for j = 1 : H1
fors = 1 : s1/im
fors2 = 1 : s2/im
for j = H : e
fors = 0 : s/im
fors1 = 1 : s1/im
if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <= theta_col
and mod(s, 2) == 0 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 0;
else if rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 1
rect(m + s * r, j + s * p1 + s1 * p) = 1;
else if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 1;
else if rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 0 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
else if rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 0;
else if rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_col and mod(s, 2) == 1 and mod(s1, 2) == 0
rect(m + s * r, j + s * p1 + s1 * p) = 1;

```

```

elseif rect(m, j) == 1 and (m + s * p1 + s1 * p) > 0 and (m + s * p1 + s1 * p) <=
1 and mod(s1, 2) == 1
    rect(m + s * r, j + s * p1 + s1 * p) = 1;
elseif rect(m, j) == 0 and (m + s * p1 + s1 * p) > 0 and
(m + s * p1 + s1 * p) <= theta_c and mod(s, 2) == 1 and mod(s1, 2) == 1
    rect(m + s * r, j + s * p1 + s1 * p) = 0;
end
end
end
end
end
end

Draw_shape(rect, theta_row, theta_col, color1, color2)
title('Rectangle after Mapping 2')
form = 1 : r
for j = 1 : H1
    fors = 1 : s1 : im
        fors2 = 1 : s2 : im
            function Draw_shape(M, a, b, color1, color2)
                figure
                axis([0 b 0 a])
                hold on
                for i = 1 : a
                    for j = 1 : b
                        if M(i, j) == 1
                            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color1);
                        elseif M(i, j) == 0
                            rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color2);
                        end
                    end
                end
            end
        end
    end
end
end

```

end

end

end

end

Code C

```
function Draw_shape(M, a, b, color1, color2)
```

```
    figure
```

```
    axis([0b0a])
```

```
    hold on
```

```
    for i = 1 : a
```

```
        for j = 1 : b
```

```
            if M(i, j) == 1
```

```
                rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color1);
```

```
            elseif M(i, j) == 0
```

```
                rectangle('Position', [j - 1, a - i, 1, 1], 'FaceColor', color2);
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
end
```

** Find all Initial rect

```

function com=create_ombination(n, k)
for i = 1 : n
tmp = [];
for j = 1 : k
tmp = [tmp; ones(k, n-i), 1] * j;
end
rr = [];
for j = 1 : k(i-1)
rr = [rr; tmp];
end
com(:, i) = rr;
end
com = com - 1
end

```



UUM
Universiti Utara Malaysia

Code D

**** Find the connected Abacus**

```

function con=Find_connected_abacus(n, k, r, e, H, str)
con = [];
· Find all possible initial rects
all = create_ombination(n, k);
· * Find the connected abacus
· Algorithm for column
if strcmp(str, I_col^I)
for nn = 1 : size(all, 1)
temp = all(nn, :);

```

```

flag = 1;
rectisconnectedabacus)
rect = zeros(r, e);
for i = 1 : e
    rect(temp(i)+ 1 : temp(i)+ H, i) = 1;
end

· *Testtheconnection

t = sum(rectT);
for i = 1 : r
    flag = flag* (t(1, i) >= 1);
end

for i = 1 : e - 1
    A = temp(i) : temp(i)+ H - 1;
    B = temp(i + 1) : temp(i + 1)+ H - 1;
    C = intersect(A, B);
    flag = flag* ( isempty(C));
end

if flag == 1
    con = [con; temp];
end

```


APPENDIX E

CHAIN TRANSFORMATION

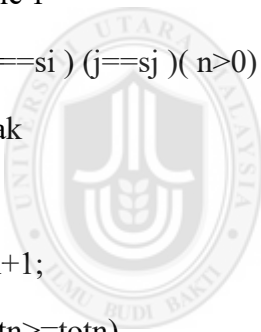
File Number one

```
clc
kkk=0;
clear;
nt=0;
v=1;
global Tmat;
r = input(' numbers of rows ')
c = input(' numbers of columns ')
mat = ones(r,c);
Tmat=mat;
r,c
= size(mat) ;
ch=21;
path=0;
tn=0;
fl=[r,c] ;
tmpv=min(fl)/2;
tpn1=ceil(tmpv)
while path <tpn1
path=path+1
v=v-.15
si=path;sj=path;
```

```

i=si;j=sj;
vv=path;
n=0;
sti=2;
stj=2;
str=('r-si-1');
nr=r-si;
nc=c-sj;
vs=1/ch;
vs=0;
nt=nt+1;
while 1
if (i==si) (j==sj) (n>0)
break
end
n=n+1;
if ( tn>=totn)
break
end
mat(i,j)=v;
tn=tn+1;
pt(tn,4)=j;
pt(tn,3)=i;
pt(tn,1)=tn;
pt(tn,2)=path;
if ((i<nr+1)(sti>0))
i=i+1;

```



```

else if((j<nc+1)(stj>0));
j=j+1;
sti=-1;
else
if (i>si)
i=i-1;
stj=-1;
else if (j>sj)
j=j-1;
if ((i==si)(j==sj)) sti=1;stj=1;
end
end
end
end
end
i;
j;
A1 = path;
A2 = n;
end
end
s=size(pt);
c=s(1);
cp=1
i=1;
clc
k=1;

```

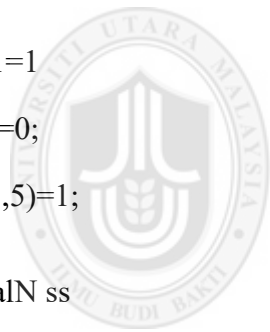


UUM
Universiti Utara Malaysia

```

pathStart(1)=1;
for i=1:s
if (i>1)
if pt(i,2) ==pt(i-1,2) k=k+1;
pathStart(k)=i;
end
end
end
k=k+1;
pathStart(k)=s(1)
k(1:tpn1)=0;
clc
p01=1
No=0;
pt(:,5)=1;
totalN ss
=size(pt)
tpn=max(pt(:,2))
for i=1:tpn
i
pn(i)=input('ÚÏÏ ÇáÚäÇÕÑ ááãÓÇÑ ');
end
for p=1:length(pn)
for i=1:totalN
if(pt(i,2)==p)(k(p)<pn(p)) pt(i,5)=11;
k(p)=k(p)+1;
end

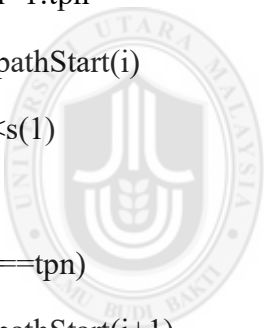
```



```

ii=pt(i,3);
jj=pt(i,4);
mat(ii,jj)= pt(i,5);
end
end
Ax=0
ii=0;
jj=0;
**ccc=length(x1)
Ax(1:pathStart(2)-1,1:5,1:3)=-1
k=0
for i=1:tpn
i1=pathStart(i)
if i<s(1)
end
if (i==tpn)
i2=pathStart(i+1)
else
i2=pathStart(i+1)-1
end
* x1=pt(pathStart(1):pathStart(2)-1,:);
x=pt(i1:i2,:);
k=k+1;
rrr=size(x)
pL(k)=rrr(1)
Ax(1:pL(k),:,i)=x;
* Ax(1:length(x),:,i)=x;

```



UUM
Universiti Utara Malaysia

```

end
Ax(1:pL(i),:,i)
clc
** x1=Ax(1:pL(1),:,1);
** x2=Ax(1:pL(2),:,2);
** x3=Ax(1:pL(3),:,3);
s=0
N=0
mTem=0
i=1
global Nx1
Nx1=0;
LoopF(i,Ax,pL,mat)
F3
File Number Two

```



```

global Nx1
global Tmat
for t=1:sv
ii=xt(t,3);
jj=xt(t,4);
tt=xt(t,5)
mat(ii,jj)= tt;
end

```

```

mat

r,c

= size(mat) ;

if (isequal(mTem,mat))

dlmwrite('Rtxt',mat,'-append','delimiter',' ','roffset',1);

*****

imagesc((1:c)+0.5,(1:r)+0.5,mat);

colormap(winter);

axis equal ;

N=N+1 ;

set(gca,'XTick',1:(c),'YTick',1:(r),...

'XLim',[1 c+1],'YLim',[1 r+1],...

'GridLineStyle','-','XGrid','on','YGrid','on');

rddd1 = 1
rddd2 = 1
Nx1=Nx1+1
Tmat(:,Nx1)=mat;

s=sprintf('000

saveas(gcf,s);

***** end

mTem=mat;

```

File Number Three

```

a b w

=size(Tmat)

Tmat2=Tmat(:,,:)

```

```

Tmat3=Tmat(:,:,1)
k=0;
m1=Tmat(:,:,1);
m2=Tmat2(:,:,1);
kk=1
for i=1:w
t=1
t=0
for j=i+1:w
kk=kk+1
m1=Tmat(:,:,i);
m2=Tmat2(:,:,j);
if Tmat(:,:,i)~= Tmat2(:,:,j)
t=1;
end
if k==150
nnn=2
end
end
if (t==0)
k=k+1
Tmat3(:,:,k)= Tmat(:,:,i)

```

File Number Four


```

function it=LoopF(i,Ax,pL,mat)
global Nx1;
Nx1=Nx1*1
LpL=length(pL)
if i>LpL
return;
end
pLt=pL(i);
xt=Ax(1:pLt,:,i);
sx(i)=size(xt,1);
sxv=sx(i)
mTem=0;
N=0;
clc
for j=1:sxv
sh=1;
Y1 = circshift(xt(:,5),sh);
xt(:,5)=Y1;
F2
Ax(1:pL(i),5,i)=Y1;
***** tt=i+1
Ax2=Ax;pL2=pL;mat2=mat;
LoopF(tt,Ax2,pL2,mat2)
end
end

```

APPENDIX F

GENERATING FUNCTION

```
clc;clear all;close all;

key=3;

tmp=key;

fprintf('=====')

for n=1:7

tmp1=[];

for i=1:size(tmp,2)

tmp1=[tmp1,ones(1,tmp(i)-1)*tmp(i),tmp(i)+2];

end

tmp=[];tmp=tmp1;

fn=size(tmp,2);

fprintf('f

(end
```

