Foundations2 Assignment 2020 Turing Machine multiplier

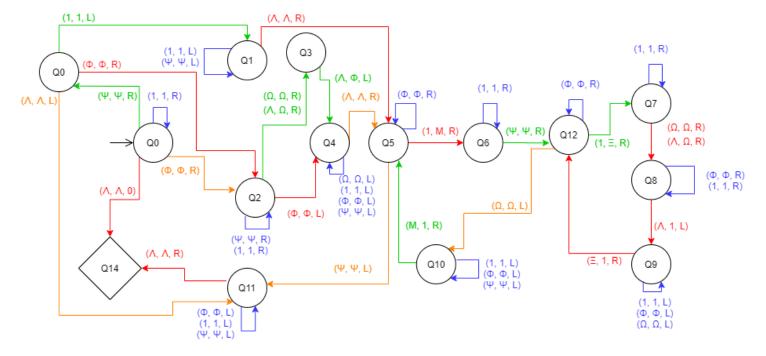
Sam Fay-Hunt sf52@hw.ac.uk

Contents

1	Turing machine multiplication	2
	1.1 Graph	2
	1.2 Formal definition	
2	Discussion of graph	3
	2.1 Logic of graph	3
	2.2 States & Symbols	3
3	TM functionality	3
4	Implementation of Turing machine	3
5	Tests	4
	5.1 Quick summary	
	5.2 2 x 2	5
	5.3 -1 x 5	16
6	Efficiency of program	35
7	Power of 3 machine	35

1 Turing machine multiplication

1.1 Graph



1.2 Formal definition

The following is the formal definition of the multiplier machine with negative numbers. $States = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}\}$

 $Symbols = \{ \land, \Phi, \Psi, \Omega, 1, M, \Xi \}$

$M_{mult}(q_0, \wedge) = (q_{12}, \wedge, 0)$ $M_{mult}(q_0, \Psi) = (q_1, \Psi, R)$	$M_{mult}(q_0, 1) = (q_0, 1, R)$ $M_{mult}(q_0, \Phi) = (q_2, \Phi, R)$
$M_{mult}(q_1, 1) = (q_1, 1, L)$	$M_{mult}(q_1, \Psi) = (q_1, \Psi, L)$
$M_{mult}(q_1, \wedge) = (q_5, \wedge, R)$	$M_{mult}(q_1, \Phi) = (q_2, \Phi, R)$
$M_{mult}(q_2, 1) = (q_2, 1, R)$	$M_{mult}(q_2, \Psi) = (q_2, \Psi, R)$
$M_{mult}(q_2, \wedge) = (q_3, \Omega, R)$	$M_{mult}(q_2,\Omega) = (q_3,\Omega,R)$
$M_{mult}(q_2, \Phi) = (q_4, \Phi, L)$	$M_{mult}(q_3, \wedge) = (q_4, \Phi, L)$
$M_{mult}(q_4,\Omega) = (q_4,\Omega,L)$	$M_{mult}(q_4, 1) = (q_4, 1, L)$
$M_{mult}(q_4, \Phi) = (q_4, \Phi, L)$	$M_{mult}(q_4, \Psi) = (q_4, \Psi, L)$
$M_{mult}(q_4, \wedge) = (q_5, \wedge, R)$	$M_{mult}(q_5, \Phi) = (q_5, \Phi, R)$
$M_{mult}(q_5, 1) = (q_6, M, R)$	$M_{mult}(q_5, \Psi) = (q_{11}, \Psi, L)$
$M_{mult}(q_6, \Phi) = (q_6, \Phi, R)$	$M_{mult}(q_6, \Psi) = (q_6, \Psi, R)$
$M_{mult}(q_6,1) = (q_7,\Xi,R)$	$M_{mult}(q_6,\Omega) = (q_{10},\Omega,L)$
$M_{mult}(q_7,\Omega) = (q_8,\Omega,R)$	$M_{mult}(q_7, \wedge) = (q_8, \Omega, R)$
$M_{mult}(q_7, 1) = (q_7, 1, R)$	$M_{mult}(q_8, \wedge) = (q_9, 1, L)$
$M_{mult}(q_8, 1) = (q_8, 1, R)$	$M_{mult}(q_8, \Phi) = (q_8, \Phi, R)$
$M_{mult}(q_9, 1) = (q_9, 1, L)$	$M_{mult}(q_9, \Phi) = (q_9, \Phi, L)$
$M_{mult}(q_9,\Omega) = (q_9,\Omega,L)$	$M_{mult}(q_9,\Xi) = (q_6,1,R)$
$M_{mult}(q_{10}, 1) = (q_{10}, 1, L)$	$M_{mult}(q_{10},\Phi) = (q_{10},\Phi,L)$
$M_{mult}(q_{10}, \Psi) = (q_{10}, \Psi, L)$	$M_{mult}(q_{10}, M) = (q_5, 1, R)$
$M_{mult}(q_{11}, 1) = (q_{11}, 1, L)$	$M_{mult}(q_{11},\Phi) = (q_{11},\Phi,L)$
$M_{mult}(q_{11}, \wedge) = (q_{12}, \wedge, R)$	

2 Discussion of graph

2.1 Logic of graph

describe the logic of the graph, why I chose this logical method. consider the correctness.

Initially the Graph determines if either, or both, of the pair of numbers on the existing tape are negative.

2.2 States & Symbols

Why I chose these states and symbols blah cut down version here

3 TM functionality

How the graph works, where to start, where you end and dealing with 0, and negative numbers

4 Implementation of Turing machine

Language choice

5 Tests

Tests, inlcuding count of tapes printed, time to complete test

5.1 Quick summary

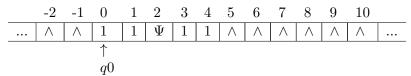
I have included a few tests on the following pages to demonstrate, basic functionality and corner cases.

Tests with tapes included in this document,

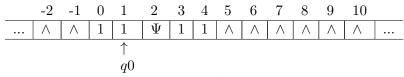
Test name	Time(ms)	Tapes
2×2	3	65
-1×5	4	111

$5.2 \quad 2 \times 2$

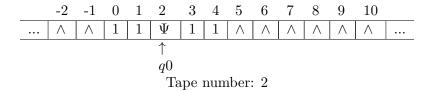
Number of tapes: 65 Tape execution time: 3ms



Tape number: 0



Tape number: 1

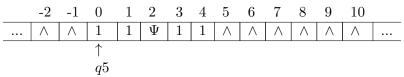


Tape number: 5

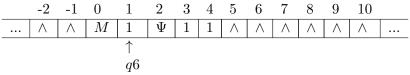


Tape number: 6

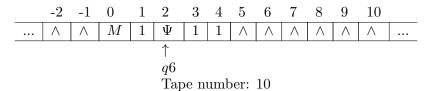
Tape number: 7

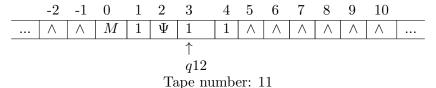


Tape number: 8



Tape number: 9

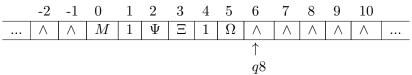




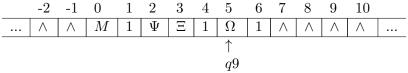
-

.....

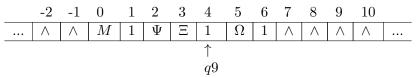
Tape number: 13



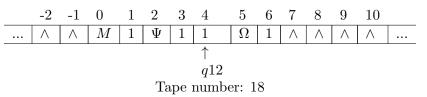
Tape number: 14

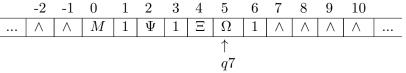


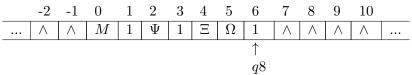
Tape number: 15



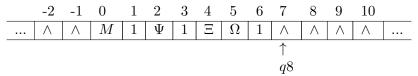
Tape number: 16



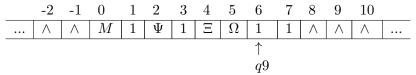




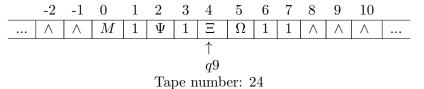
Tape number: 20



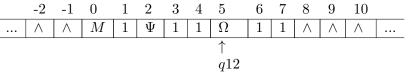
Tape number: 21



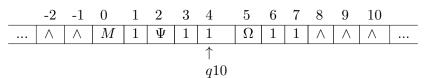
Tape number: 22



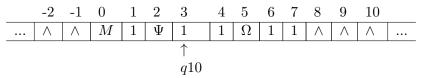
*



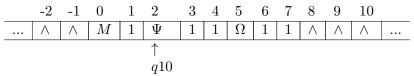
Tape number: 25



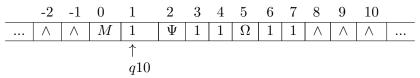
Tape number: 26

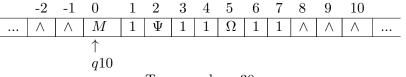


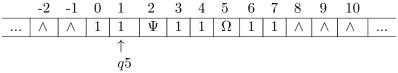
Tape number: 27



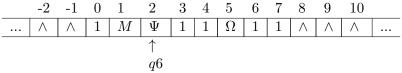
Tape number: 28



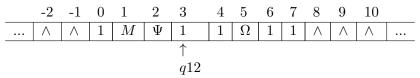




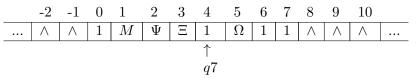
Tape number: 31



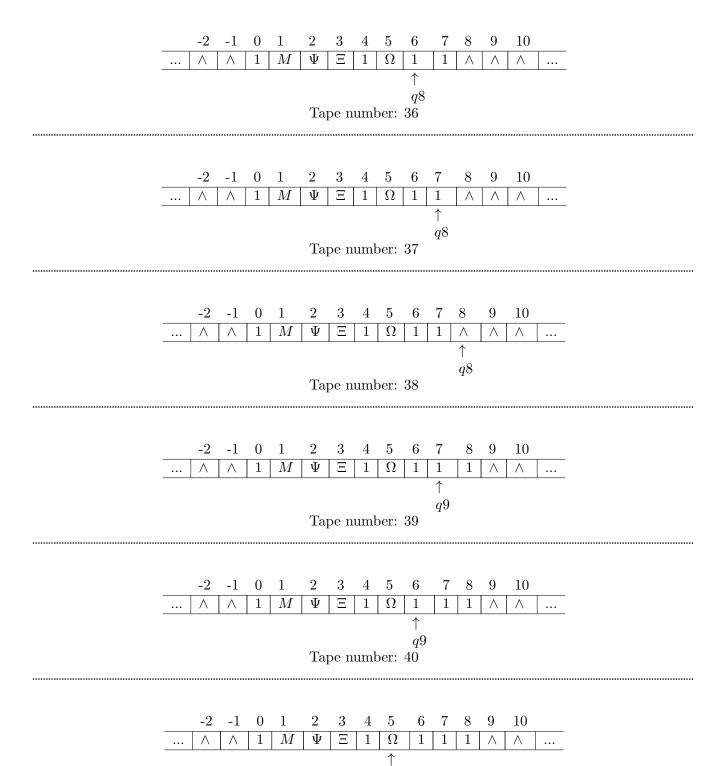
Tape number: 32



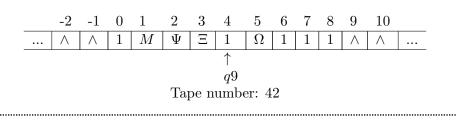
Tape number: 33

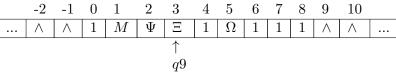


Tape number: 34

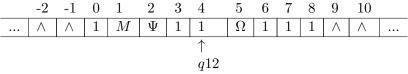


q9

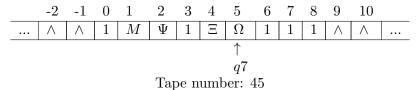


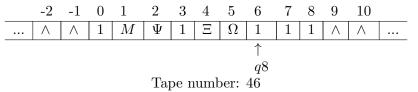


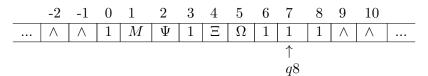
Tape number: 43



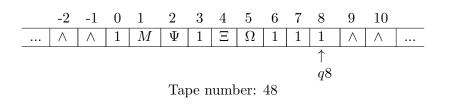
Tape number: 44



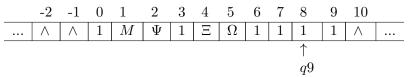




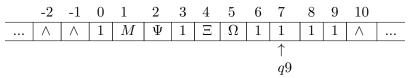
Tape number: 47



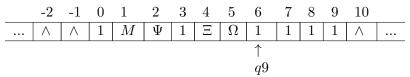
Tape number: 49



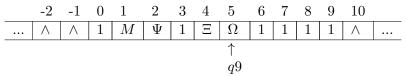
Tape number: 50

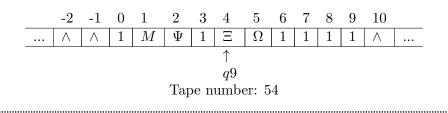


Tape number: 51

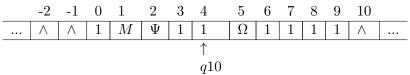


Tape number: 52

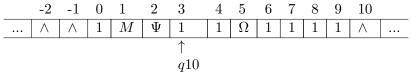




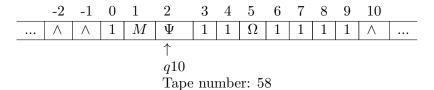
Tape number: 55

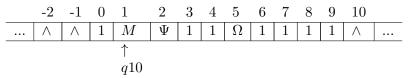


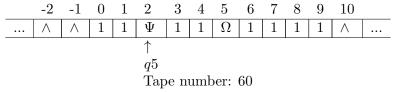
Tape number: 56

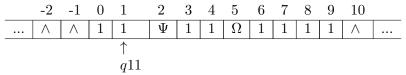


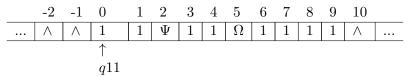
Tape number: 57



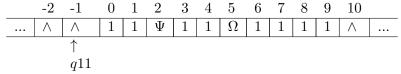




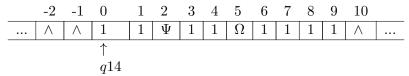




Tape number: 62

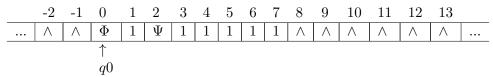


Tape number: 63

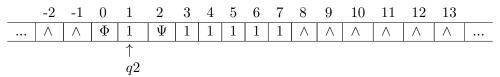


5.3 -1 x 5

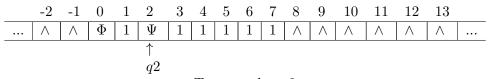
Number of tapes: 111 Tape execution time: 4ms



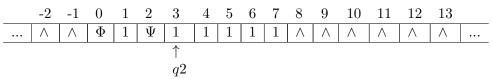
Tape number: 0



Tape number: 1

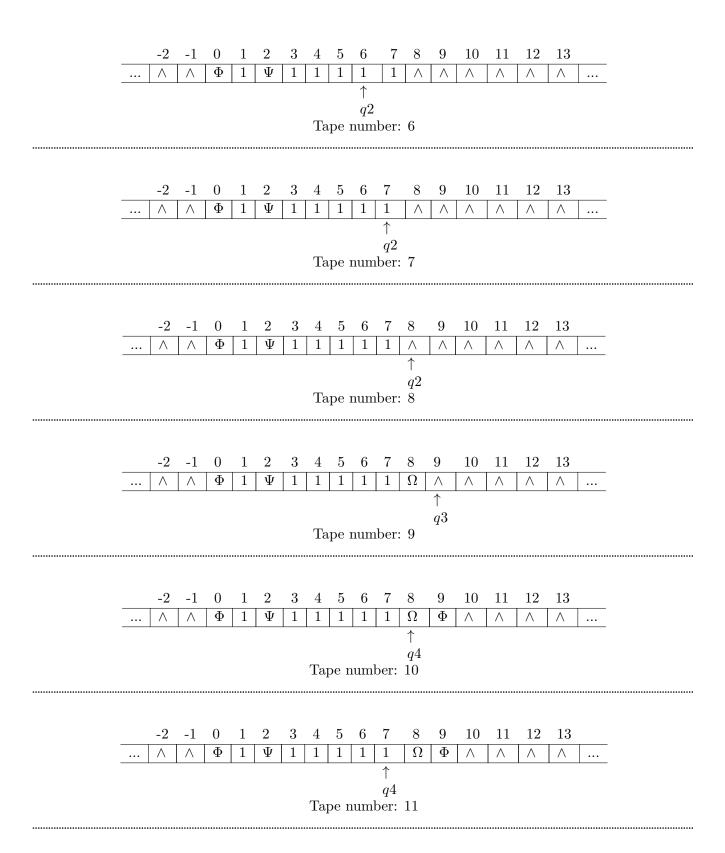


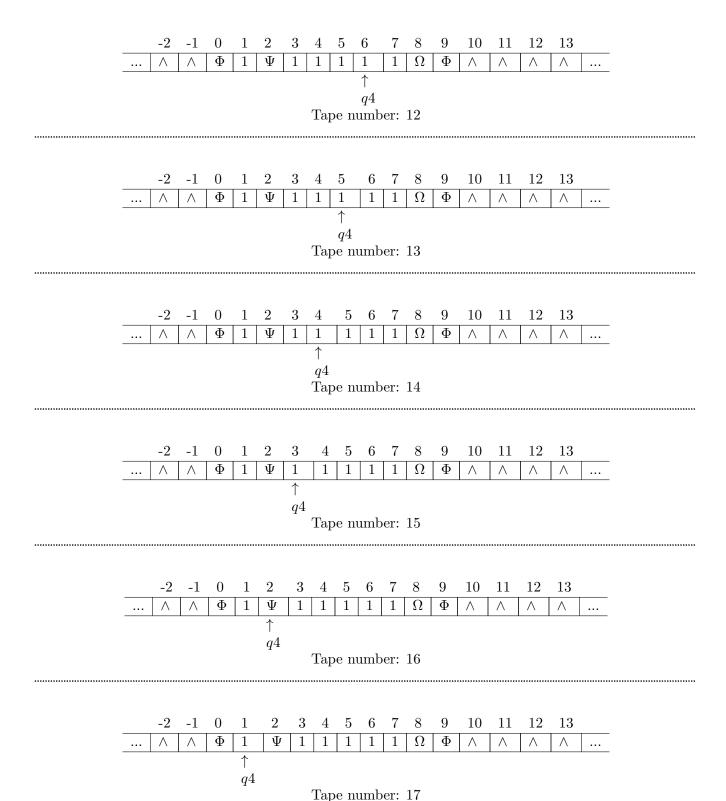
Tape number: 2

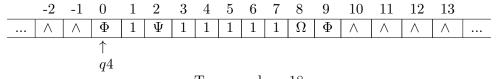


Tape number: 3

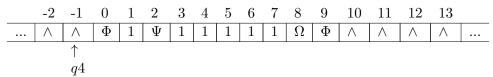
Tape number: 4



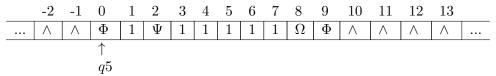




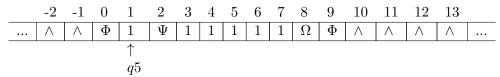
Tape number: 18



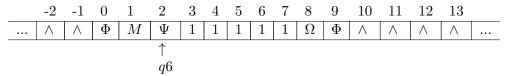
Tape number: 19



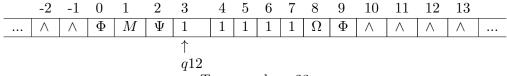
Tape number: 20



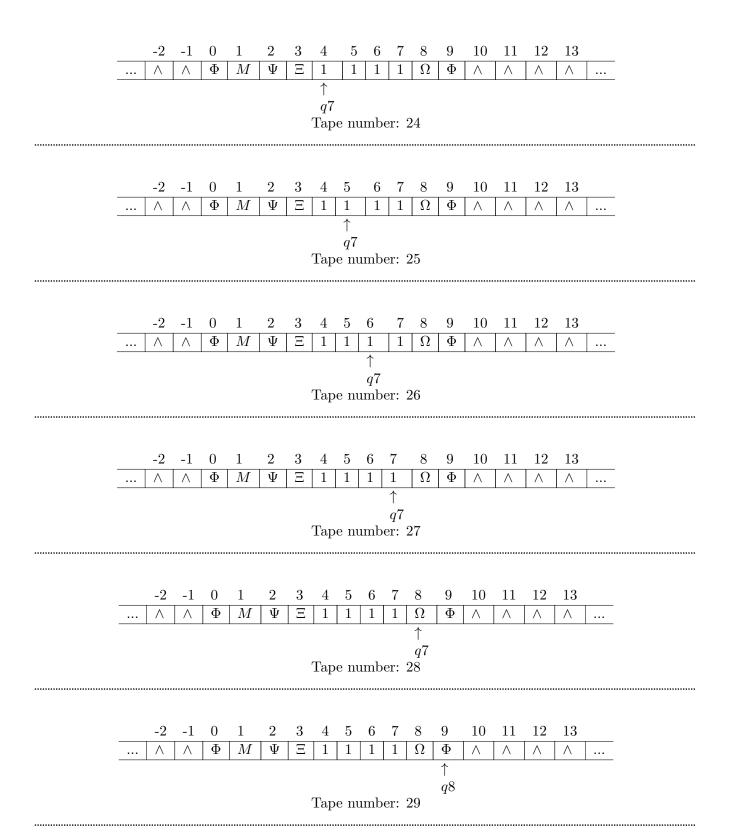
Tape number: 21

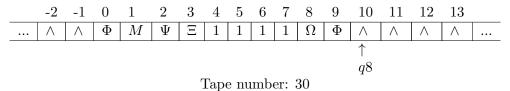


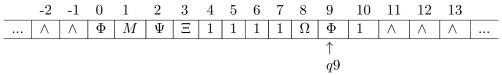
Tape number: 22



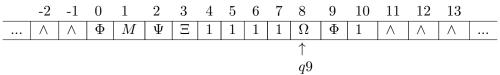
Tape number: 23



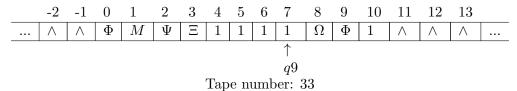




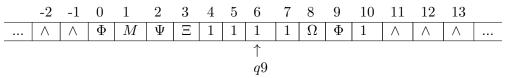
Tape number: 31



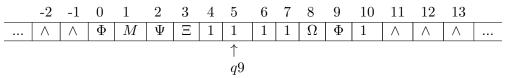
Tape number: 32

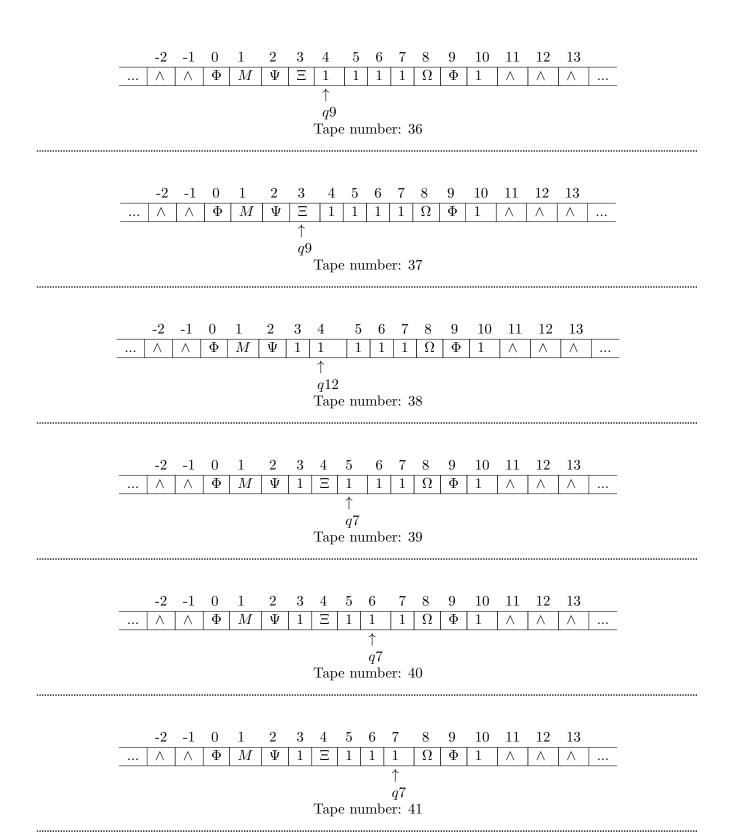


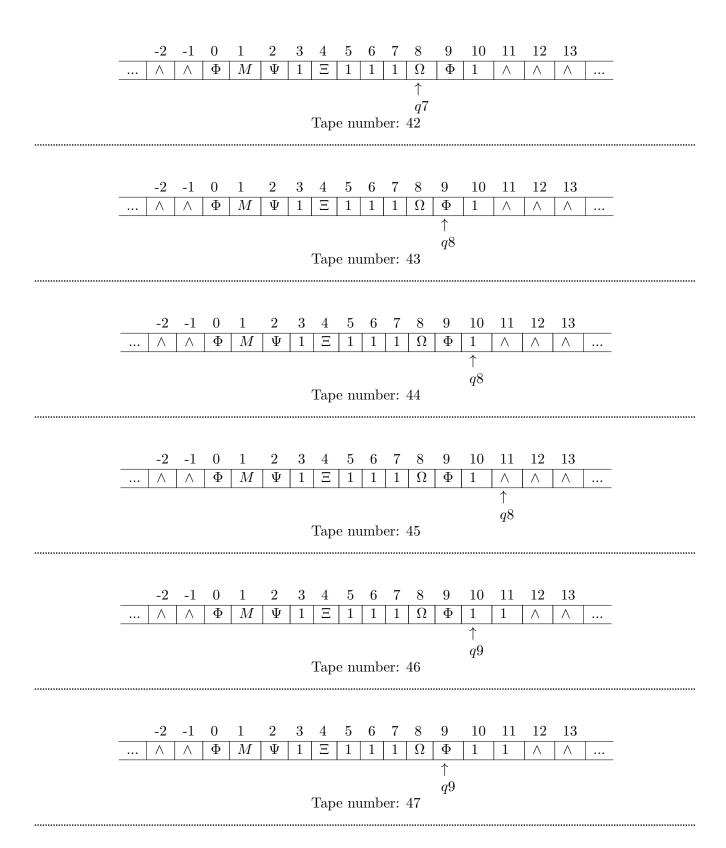
Tape number. 99

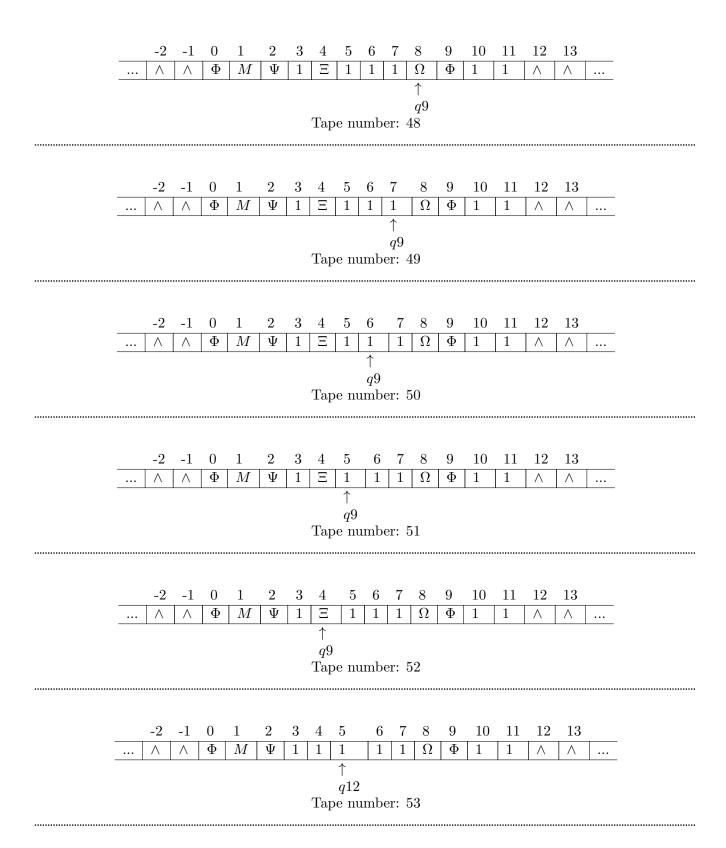


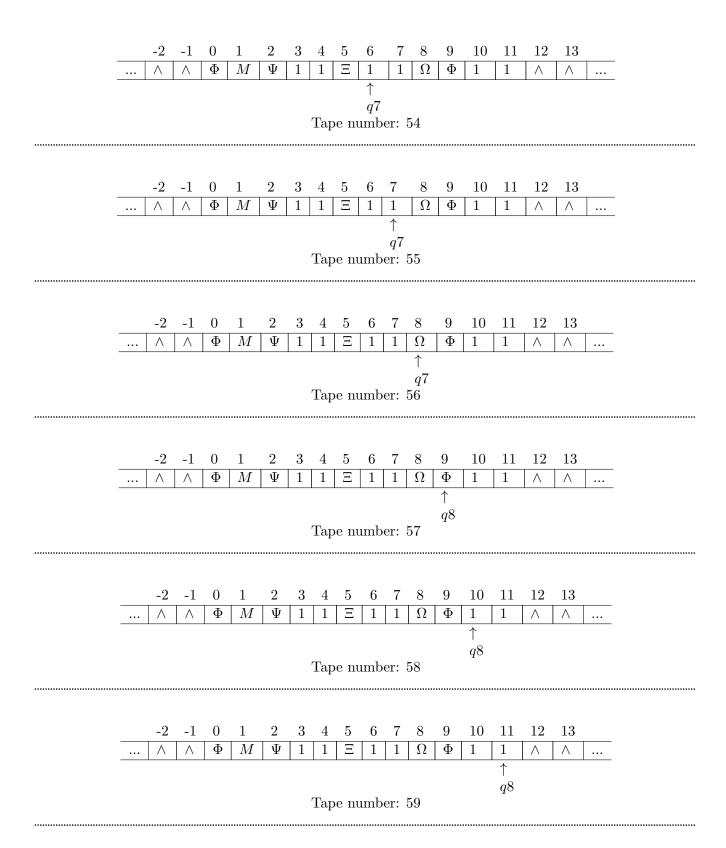
Tape number: 34

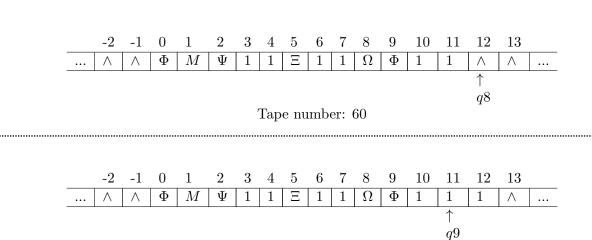


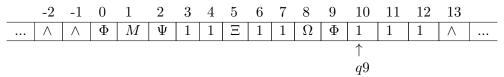




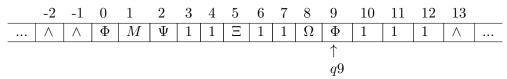




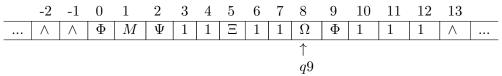




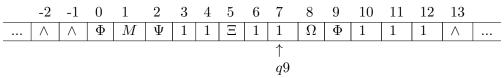
Tape number: 62

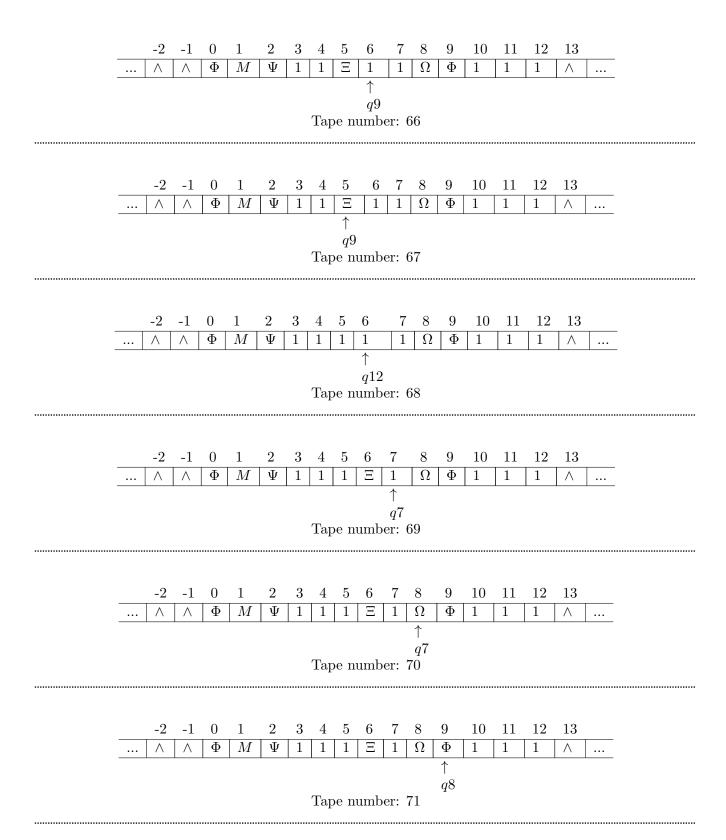


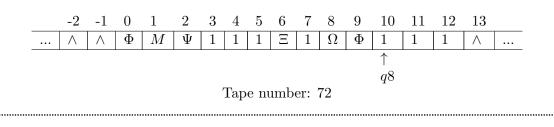
Tape number: 63

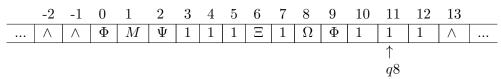


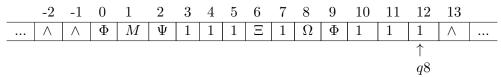
Tape number: 64



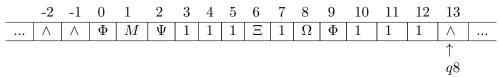




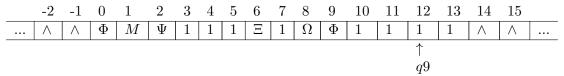




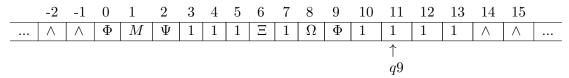
Tape number: 74

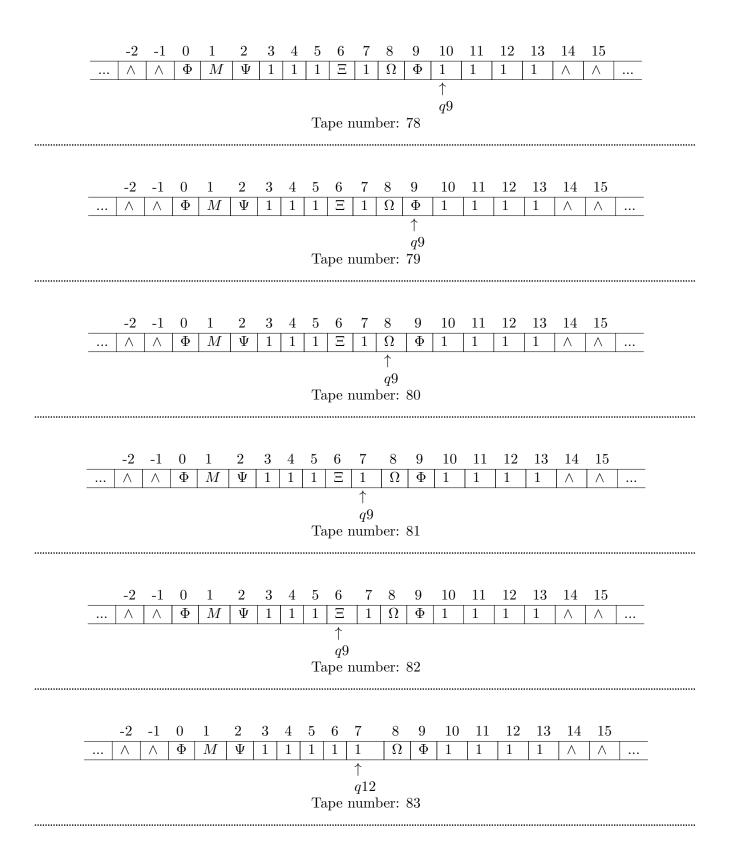


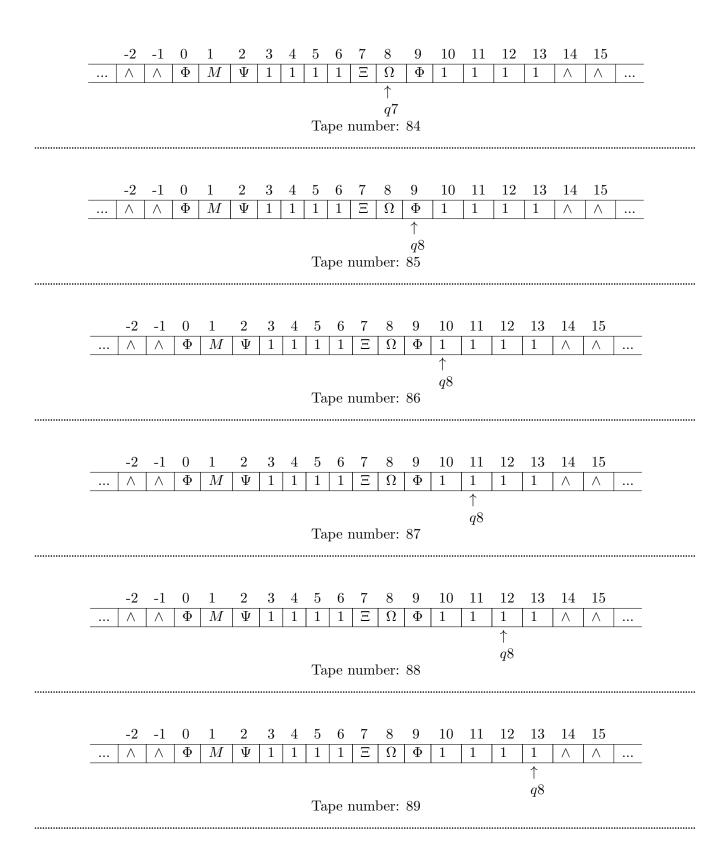
Tape number: 75

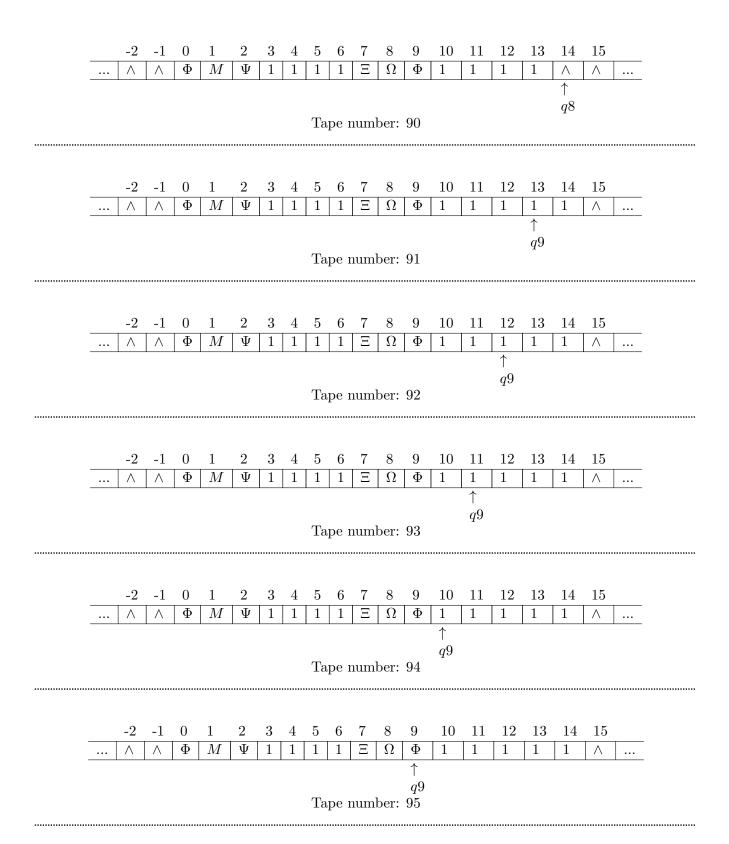


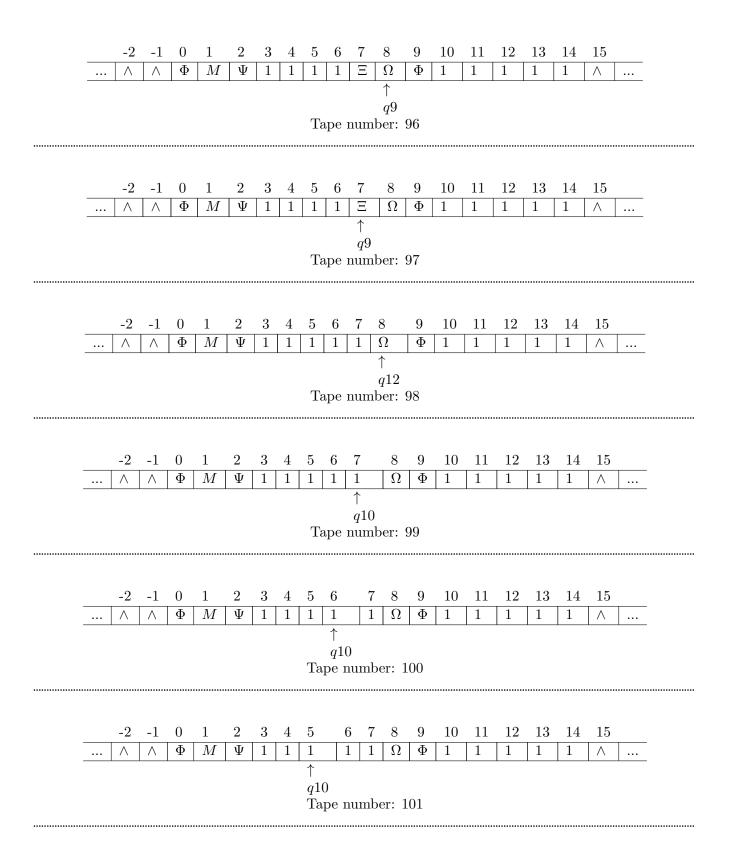
Tape number: 76

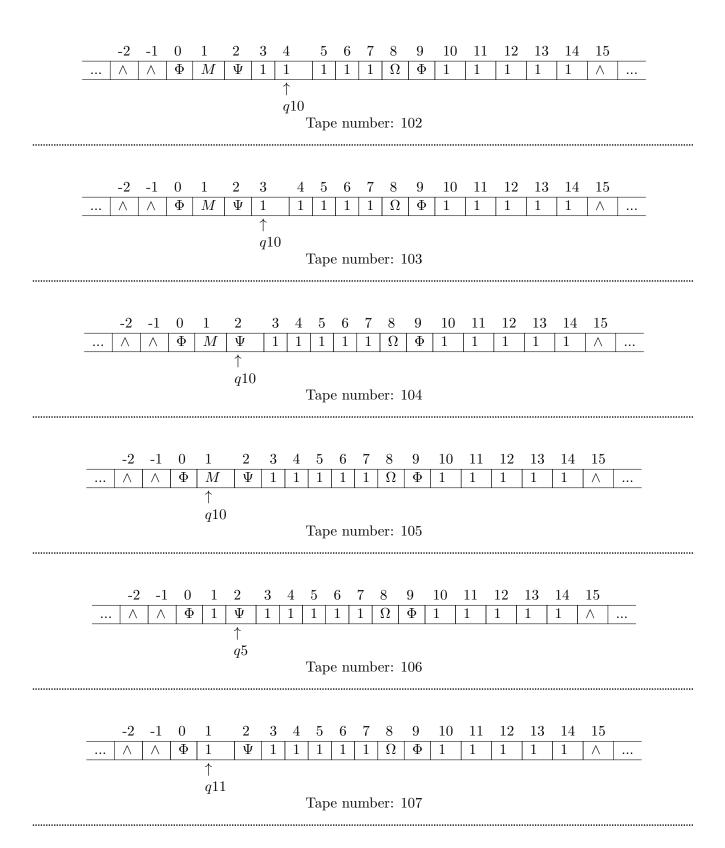


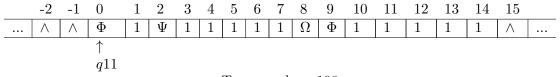




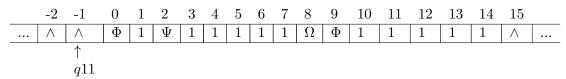




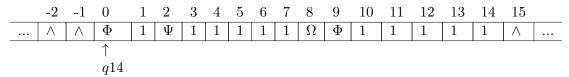




Tape number: 108



Tape number: 109



Tape number: 110

6 Efficiency of program

comment on the number of computation sequences produced and the time taked to complete each test

7 Power of 3 machine

 asd