THEORY OF COMPUTATION
CS302

PROJECT REPORT

BINARY DIVISION TURING MACHINE

GROUP MEMBERS-

Kumari Renuka(U101115FCS111)
Rishabh K Kandoi(U101115FCS283)
Shailesh Mohta(U101115FCS305)
Tanmay Patil(U101115FCS164)

BINARY DIVISION TUNING MACHINE

PROJECT DESCRIPTION

The designed Turing Machine divides a binary number by a divisor and provides the remainder and quotient as an output. Hence in order to design the Turing Machine the following logic mentioned below was applied by keeping the following constraints into mind.

Input constraint:

Give two binary numbers divided by a space, and both the numbers should be of same number of digits.

Basic concept:

Binary division is nothing but successive subtraction of divisor from dividend till the subtraction yields a negative number.

Calculating binary division consists of following phases:

1. Copying the divisor.
2. Calculating 2's compliment for the divisor.
3. Using binary adder to add the dividend with the 2's compliment of divisor and thus update the output by overriding the dividend.
4. Use binary comparator to compare if the new dividend is still greater than the divisor or not. If greater, again go to step 3, else halt the Turing Machine. Halting gives the updated dividend as remainder which is the final output.
5. Also insert a binary counter each time the loop runs, as it would provide us with the quotient.

OBJECTVE:

To design a Turing Machine in order to calculate remainder and quotient when binary numbers are divided.

COMPLEXITY:

The complexity of the designed Turing Machine is-

a) Number of states in total are 40.

b) The Turing machine is designed with single tape, if it would have been designed with the multiple tape, the number of states might have reduced with less number of steps to calculated the desired output.

PRODUCTION SET

TURING MACHINE CODE :

The following code is run on morphett.info in order to check the correctness. And it works perfectly. The code gives the correct output as remainder and quotient whenever two binary numbers are divided.

Halting condition is-

- When binary comparator finds that dividend is less than the divisor.

Code Snippet-

```
0 * * r 0

0 _ _ r q1

q1 1 u r q2

q1 0 v r q6

q1 # # l q8
```

q2 * * r q2

q2 _ # r q3

q2 # # r q3

q3 * * r q3

q3 _ 1 l q4

q4 * * l q4

q4 # # l q5

q5 * * l q5

q5 u u r q1

q5 v v r q1

q6 * * r q6

q6 _ # r q7

q6 # # r q7

q7 * * r q7

q7 _ 0 l q4

q8 v 0 l q8

q8 u 1 l q9

q9 u 0 l q9

q9 v 1 l q9

q9 _ _ l q10

q10 * * l q10

q10 _ _ r q11

q11 * * r q11

q11 _ _ r q12

q12 * * r q12

q12 # # l q13

q13 0 u l q14

q13 _ _ r q19

q13 1 v l q15

q13 v v l q13

q13 u u l q13

q14 * * l q14

q14 _ _ l q36

q36 * * l q36

q36 _ _ r q11

q36 1 y r q11

q36 0 x r q11

q19 u 0 r q19

q19 v 1 r q19

q19 # # l q37

q20 x 0 l q20

q20 y 1 l q20

q20 * * l q20

q20 _ _ l q39

q39 1 0 l q39

q39 _ 1 r q40

q39 0 1 r q40

q40 * * r q40

q40 _ _ r q41

q41 _ _ r q41

q41 * _ r q22

q15 * * l q15

q15 _ _ l q16

q16 * * l q16

q16 1 0 l q16

q16 _ 1 * q17

q16 0 1 * q17

q17 * * r q17

q17 _ _ l q18

q17 x x l q18

q17 y y l q18

q18 1 y r q11

q18 0 x r q11

q37 * * l q37

q37 _ _ l q20

q22 _ _ r q35

q22 1 y r q23

q22 0 x r q27

q23 * * r q23

q23 # # r q24

q24 * * r q24

q24 1 y l q25

q24 _ _ l q29

q24 0 0 l q29

q25 * * l q25

q25 _ _ l q26

q26 * * l q26

q26 x x r q22

q26 y y r q22

q27 * * r q27

q27 # # r q28

q28 * * r q28

q28 0 x l q25

q28 _ _ l q29

q28 1 1 r q32

q29 x 0 l q29

q29 y 1 l q29

q29 # # l q30

q30 * * l q30

q30 _ _ l q31

q31 x 0 l q31

q31 y 1 l q31

q31 _ _ r q11

q31 0 0 l q31

q31 1 1 l q31

q32 * * r q32

q32 _ _ l q33

q33 * _ l q33

q33 _ _ l q34

q34 x 0 l q34

q34 y 1 l q34

q34 _ _ r halt

q34 * * l q34

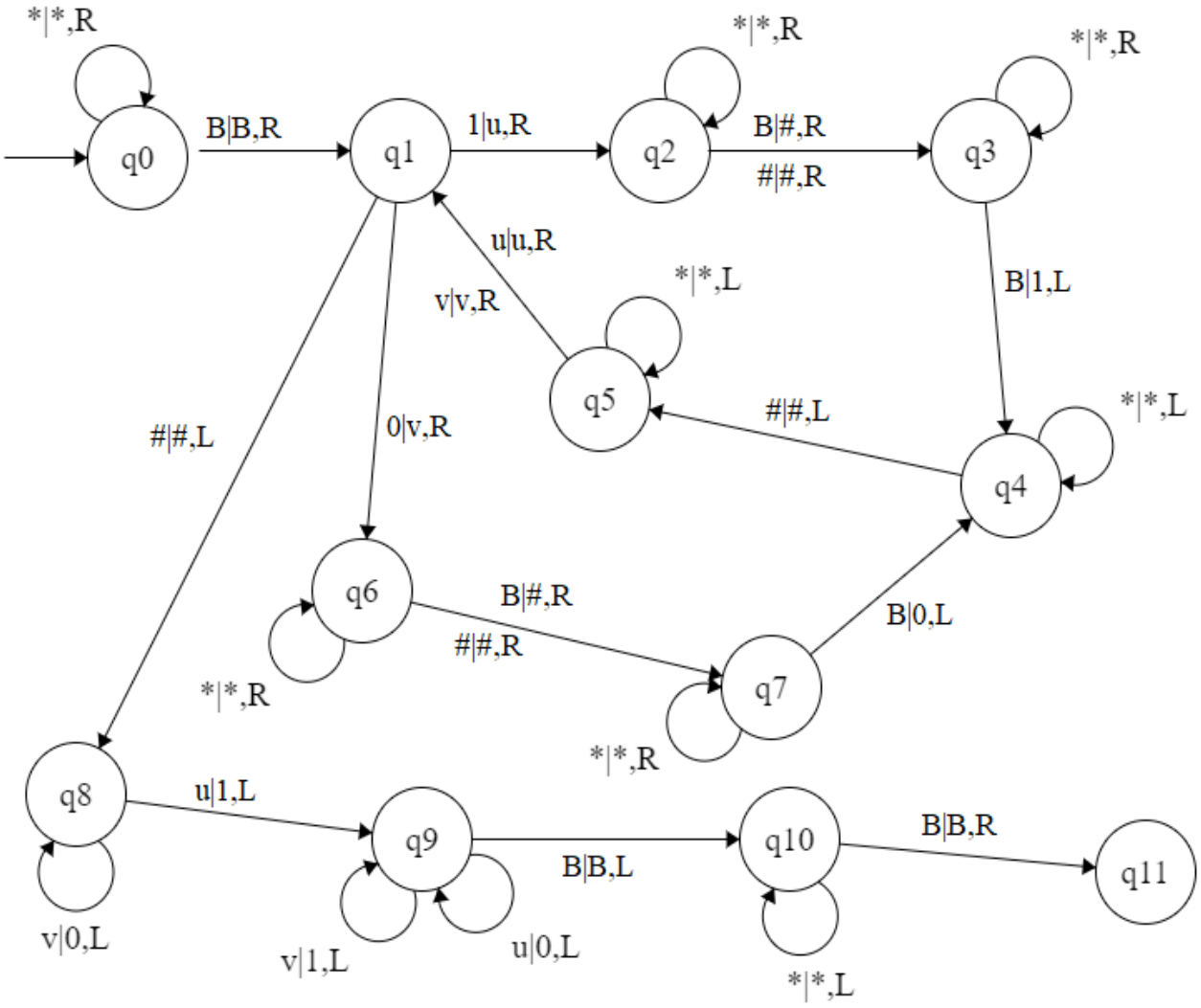
q35 * * r q35

q35 _ _ l q29

TRANSITION DIAGRAM

Step 1:

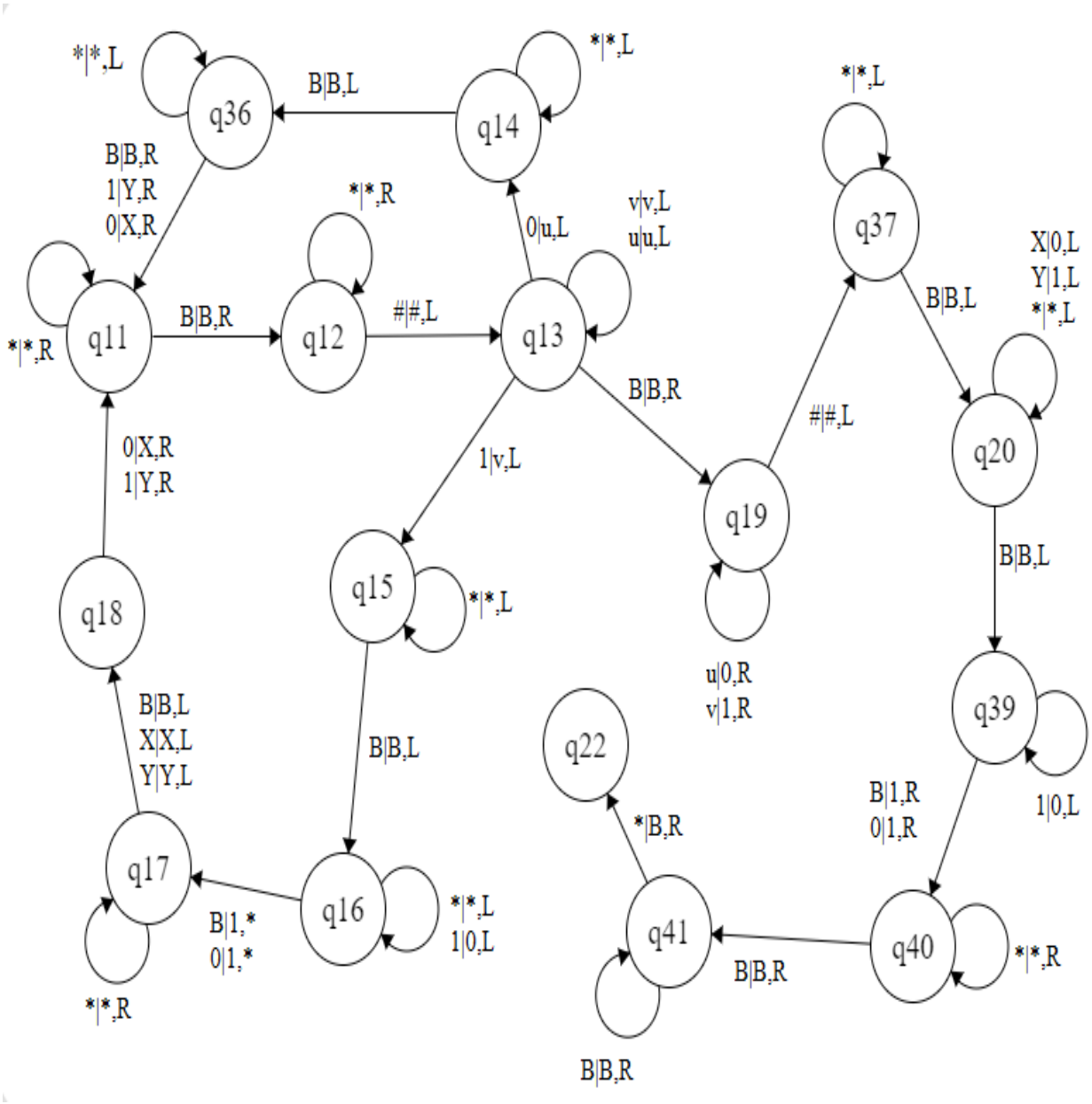
Copy the divisor, along with conversion of divisor to it’s 2’s compliment



“q11 continued in next diagram”

Step 2:

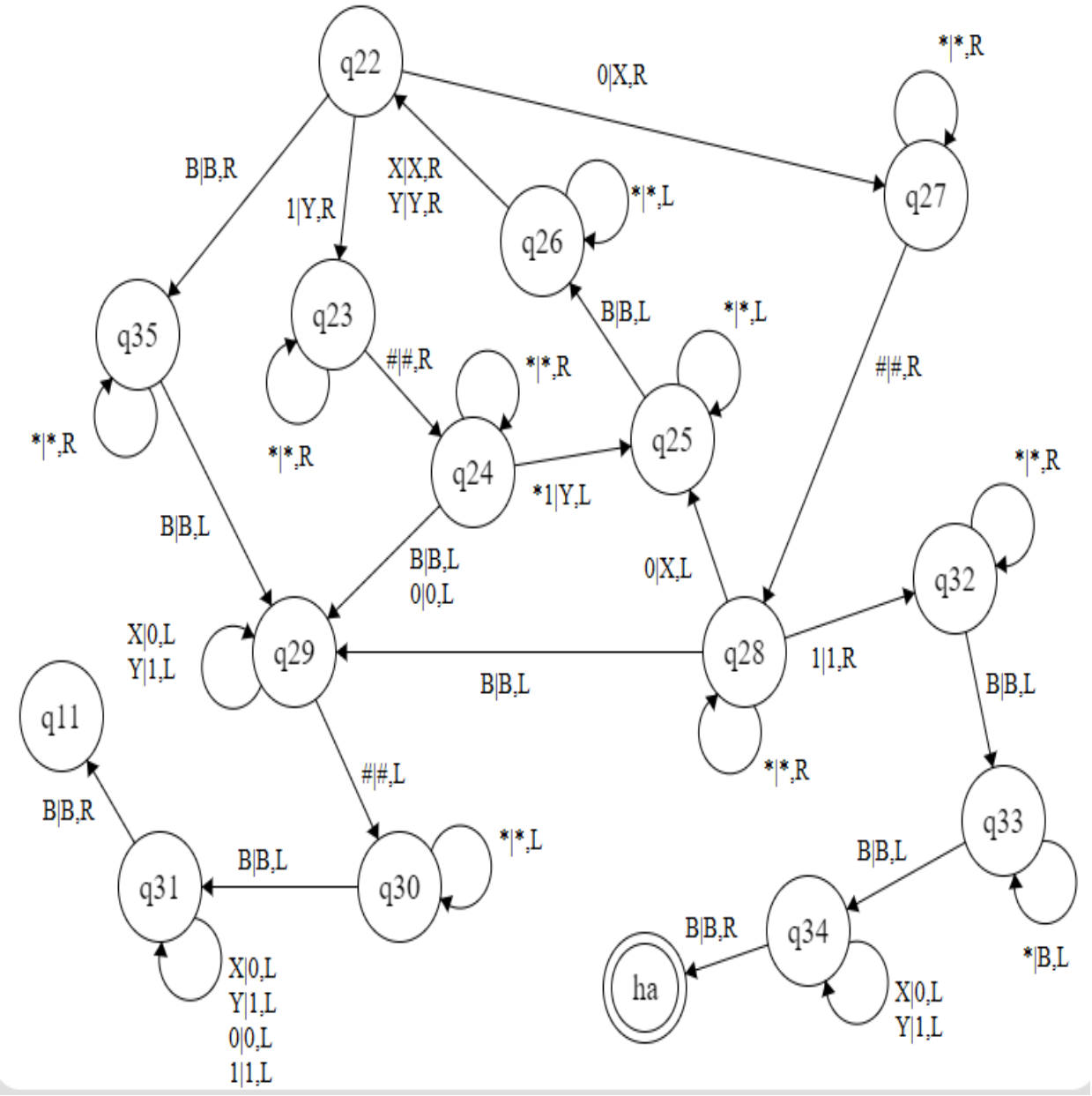
Add the dividend to the 2’s compliment of divisor, and after that, update the binary counter to maintain value of quotient alongside.



“q22 continued in next diagram”

Step 3:

Compare the updated dividend with the copied divisor as an exit condition from the loop.



“Either halts with acceptance or goes to loop to q11 again”

TRANSITION TABLE :-

	0	1	B	u	v	x	y	#
0	0,r,0	1,r,q1	B,r,q1	u,r,0	v,r,0	x,r,0	y,r,0	#,r,0
q1	v,r,q6	u,r,q2	-	-	-	-	-	#,l,q8
q2	0,r,q2	1,r,q2	#,r,q3	u,r,q2	v,r,q2	x,r,q2	y,r,q2	#,r,q3
q3	0,r,q3	1,r,q3	1,l,q4	u,r,q3	v,r,q3	x,r,q3	y,r,q3	#,r,q3
q4	0,l,q4	1,l,q4	B,l,q4	u,l,q4	v,l,q4	x,l,q4	y,l,q4	#,l,q5
q5	0,l,q5	1,l,q5	B,l,q5	u,r,q1	v,r,q1	x,l,q5	y,l,q5	#,l,q5
q6	0,r,q6	1,r,q6	#,r,q7	u,r,q6	v,r,q6	x,r,q6	y,r,q6	#,r,q7
q7	0,r,q7	1,r,q7	0,l,q4	u,r,q7	v,r,q7	x,r,q7	y,r,q7	#,r,q7
q8	-	-	-	1,l,q9	0,l,q8	-	-	-
q9	-	-	B,l,q10	0,l,q9	1,l,q9	-	-	-
q10	0,l,q10	1,l,q10	B,r,q11	u,l,q10	v,l,q10	x,l,q10	y,l,q10	#,l,q10
q11	0,r,q11	1,r,q11	B,r,q12	u,r,q11	v,r,q11	x,r,q11	y,r,q11	#,r,q11
q12	0,r,q12	1,r,q12	B,r,q12	u,r,q12	v,r,q12	x,r,q12	y,r,q12	#,l,q13
q13	0,l,q14	v,l,q15	B,r,q19	u,l,q13	v,l,q13	-	-	-
q14	0,l,q14	1,l,q14	B,l,q36	u,l,q14	v,l,q14	x,l,q14	y,l,q14	#,l,q14
q15	0,l,q15	1,l,q15	B,l,q16	u,l,q15	v,l,q15	x,l,q15	y,l,q15	#,l,q15
q16	1,*,q17	0,l,q16	1,*,q17	u,l,q16	v,l,q16	x,l,q16	y,l,q16	#,l,q16
q17	0,r,q17	1,r,q17	B,l,q18	u,r,q17	v,r,q17	x,l,q18	y,l,q18	#,l,q17
q18	x,r,q11	y,r,q11	-	-	-	-	-	-
q19	-	-	-	0,r,q19	1,r,q19	-	-	#,l,q37
q20	0,l,q20	1,l,q20	B,l,q39	u,l,q20	v,l,q20	0,l,q20	1,l,q20	#,l,q20
q22	x,r,q27	y,r,q23	B,r,q35	-	-	-	-	-
q23	0,r,q23	1,r,q23	B,r,q23	u,r,q23	v,r,q23	x,r,q23	y,r,q23	#,r,q24
q24	0,l,q29	y,l,q25	B,l,q29	u,r,q24	v,r,q24	x,r,q24	y,r,q24	#,r,q24
q25	0,l,q25	1,l,q25	B,l,q26	u,l,q25	v,l,q25	x,l,q25	y,l,q25	#,l,q25

	0	1	B	u	v	x	y	#
q26	0,l,q26	1,l,q26	B,l,q26	u,l,q26	v,l,q26	x,r,q22	y,r,q22	#,l,q26
q27	0,r,q27	1,r,q27	B,r,q27	u,r,q27	v,r,q27	x,r,q27	y,r,q27	#,r,q28
q28	x,l,q25	1,r,q32	B,l,q29	u,r,q28	v,r,q28	x,r,q28	y,r,q28	#,r,q28
q29	-	-	-	-	-	0,l,q29	1,l,q29	#,l,q30
q30	0,l,q30	1,l,q30	B,l,q31	u,l,q30	v,l,q30	x,l,q30	y,l,q30	#,l,q30
q31	0,l,q31	1,l,q31	B,r,q11	-	-	0,l,q31	1,l,q31	-
q32	0,r,q32	1,r,q32	B,l,q33	u,r,q32	v,r,q32	x,r,q32	y,r,q32	#,r,q32
q33	B,l,q33	B,l,q33	B,l,q34	B,l,q33	B,l,q33	B,l,q33	B,l,q33	B,l,q33
q34	-	-	B,r,ha	-	-	0,l,q34	1,l,q34	-
q35	0,r,q35	1,r,q35	B,l,q29	u,r,q35	v,r,q35	x,r,q35	y,r,q35	#,r,q35
q36	x,r,q11	y,r,q11	B,r,q11	u,l,q36	v,l,q36	x,l,q36	y,l,q36	#,l,q36
q37	0,l,q37	1,l,q37	B,l,q20	u,l,q37	v,l,q37	x,l,q37	y,l,q37	#,l,q37
q39	1,r,q40	0,l,q39	1,r,q40	-	-	-	-	-
q40	0,r,q40	1,r,q40	B,r,q41	u,r,q40	v,r,q40	x,r,q40	y,r,q40	#,r,q40
q41	B,r,q22	B,r,q22	B,r,q41	B,r,q22	B,r,q22	B,r,q22	B,r,q22	B,r,q22

EXAMPLE:

The following is an example in which 5 is divided by 2 in order to get the remainder and divisor.

101 010

Step 1 : 101 010

Step 2 : 101 010

Step 3 : 101 010

Step 4 : 101_010

Step 5 : 101 010

Step 6 : 101 v10

Step 7 : 101 v10

Step 8 : 101 v10_

Step 9 : 101 v10#_

Step 10 : 101 v10#0

Step 11 : 101 v10#0

Step 12 : 101 v10#0

Step 13 : 101 v10#0

Step 14 : 101 v10#0

Step 15 : 101 vu0#0

Step 16 : 101 vu0#0

Step 17 : 101 vu0#0

Step 18 : 101 vu0#0_

Step 19 : 101 vu0#01

Step 20 : 101 vu0#01

Step 21 : 101 vu0#01

Step 22 : 101 vu0#01

Step 23 : 101 vu0#01

Step 24 : 101 vuv#01

Step 25 : 101 vuv#01

Step 26 : 101 vuv#01

Step 27 : 101 vuv#01_

Step 28 : 101 vuv#010

Step 29 : 101 vuv#010

Step 30 : 101 vuv#010

Step 31 : 101 vuv#010

Step 32 : 101 vu0#010

Step 33 : 101 v10#010

Step 34 : 101_110#010

Step 35 : 101 110#010

Step 36 : 101 110#010

Step 37 : 101 110#010

Step 38 : _101 110#010

Step 39 : 101 110#010

Step 40 : 101 110#010

Step 41 : 101 110#010

Step 42 : 101_110#010

Step 43 : 101 110#010

Step 44 : 101 110#010

Step 45 : 101 110#010

Step 46 : 101 110#010

Step 47 : 101 110#010

Step 48 : 101 11u#010

Step 49 : 101 11u#010

Step 50 : 101_11u#010

Step 51 : 101 11u#010

Step 52 : 10y_11u#010

Step 53 : 10y 11u#010

Step 54 : 10y 11u#010

Step 55 : 10y 11u#010

Step 56 : 10y 11u#010

Step 57 : 10y 11u#010

Step 58 : 10y 11u#010

Step 59 : 10y 1vu#010

Step 60 : 10y_1vu#010

Step 61 : 10y 1vu#010

Step 62 : 10y 1vu#010

Step 63 : 11y 1vu#010

Step 64 : 11y 1vu#010

Step 65 : 1yy 1vu#010

Step 66 : 1yy_1vu#010

Step 67 : 1yy 1vu#010

Step 68 : 1yy 1vu#010

Step 69 : 1yy 1vu#010

Step 70 : 1yy 1vu#010

Step 71 : 1yy 1vu#010

Step 72 : 1yy 1vu#010

Step 73 : 1yy 1vu#010

Step 74 : 1yy 1vu#010

Step 75 : 1yy 1vu#010

Step 76 : 1yy 1vu#010

Step 77 : 1yy 1vu#010

Step 78 : 1yy_vvu#010

Step 79 : 1yy_vvu#010

Step 80 : 1yy vvu#010

Step 81 : 1yy vvu#010

Step 82 : 1yy vvu#010

Step 83 : _0yy vvu#010

Step 84 : 10yy vvu#010

Step 85 : 10yy vvu#010

Step 86 : 10yy vvu#010

Step 87 : 1xyy vvu#010

Step 89 : 1xyy vvu#010

Step 90 : 1xyy_vvu#010

Step 91 : 1xyy vvu#010

Step 92 : 1xyy vvu#010

Step 93 : 1xyy vvu#010

Step 94 : 1xyy vvu#010

Step 95 : 1xyy vvu#010

Step 96 : 1xyy vvu#010

Step 97 : 1xyy vvu#010

Step 98 : 1xyy 1vu#010

Step 99 : 1xyy 11u#010

Step 100 : 1xyy 110#010

Step 101 : 1xyy 110#010

Step 102 : 1xyy 110#010

Step 103 : 1xyy 110#010

Step 104 : 1xyy 110#010

Step 105 : 1xyy 110#010

Step 106 : 1xy1 110#010

Step 107 : 1x11 110#010

Step 108 : 1011 110#010

Step 109 : 1011 110#010

Step 110 : 1011 110#010

Step 111 : 1 1011 110#010

Step 112 : 1 1011 110#010

Step 113 : 1 011 110#010

Step 114 : 1 x11 110#010

Step 115 : 1 x11 110#010

Step 116 : 1 x11 110#010

Step 117 : 1 x11 110#010

Step 118 : 1 x11 110#010

Step 120 : 1 x11 110#010

Step 121 : 1 x11 110#010

Step 122 : 1 x11 110#010

Step 123 : 1 x11 110#x10

Step 124 : 1 x11 110#x10

Step 125 : 1 x11 110#x10

Step 126 : 1 x11 110#x10

Step 127 : 1 x11_110#x10

Step 128 : 1 x11 110#x10

Step 129 : 1 x11 110#x10

Step 130 : 1 xy1 110#x10

Step 131 : 1 xy1_110#x10

Step 132 : 1 xy1 110#x10

Step 132 : 1 xy1 110#x10

Step 133 : 1 xy1 110#x10

Step 134 : 1 xy1 110#x10

Step 135 : 1 xy1 110#x10

Step 136 : 1 xy1 110#x10

Step 137 : 1 xy1 110#xy0

Step 138 : 1 xy1 110#xy0

Step 139 : 1 xy1 110#xy0

Step 140 : 1 xy1 110#xy0

Step 141 : 1 xy1 110#xy0

Step 142 : 1 xy1_110#xy0

Step 143 : 1 xy1 110#xy0

Step 144 : 1 xy1 110#xy0

Step 145 : 1 xy1 110#xy0

Step 146 : 1 xyy_110#xy0

Step 147 : 1 xyy 110#xy0

Step 148 : 1 xyy 110#xy0

Step 149 : 1 xyy 11<u>0</u>#xy0

Step 150 : 1 xyy 110<u>#</u>xy0

Step 151 : 1 xyy 110#<u>x</u>10

Step 152 : 1 xyy 110#x<u>1</u>0

Step 153 : 1 xyy 110#<u>x</u>10

Step 154 : 1 xyy 110<u>#</u>010

Step 155 : 1 xyy 11<u>0</u>#010

Step 156 : 1 xyy 1<u>1</u>0#010

Step 157 : 1 xyy <u>1</u>10#010

Step 158 : 1 xyy_110#010

Step 159 : 1 xy<u>1</u> 110#010

Step 161 : 1 x<u>y</u>1 110#010

Step 162 : 1 <u>x</u>11 110#010

Step 163 : 1_011 110#010

Step 164 : 1 <u>0</u>11 110#010

Step 165 : 1 0<u>1</u>1 110#010

Step 166 : 1 01<u>1</u> 110#010

Step 167 : 1 011_110#010

Step 168 : 1 011 <u>1</u>10#010

Step 169 : 1 011 1<u>1</u>0#010

Step 170 : 1 011 11<u>0</u>#010

Step 171 : 1 011 110<u>#</u>010

Step 171 : 1 011 11<u>0</u>#010

Step 172 : 1 011 1<u>1</u>u#010

Step 173 : 1 011 <u>1</u>1u#010

Step 174 : 1 011_11u#010

Step 175 : 1 01<u>1</u> 11u#010

Step 176 : 1 01y_11u#010

Step 177 : 1 01y 11u#010

Step 179 : 1 01y 11u#010

Step 180 : 1 01y 11u#010

Step 181 : 1 01y 11u#010

Step 182 : 1 01y 11u#010

Step 183 : 1 01y 11u#010

Step 184 : 1 01y 1vu#010

Step 185 : 1 01y 1vu#010

Step 186 : 1 01y 1vu#010

Step 187 : 1 01y 1vu#010

Step 188 : 1 00y 1vu#010

Step 189 : 1 10y 1vu#010

Step 190 : 1 10y 1vu#010

Step 191 : 1 10y 1vu#010

Step 192 : 1 1xy 1vu#010

Step 193 : 1 1xy 1vu#010

Step 194 : 1 1xy 1vu#010

Step 195 : 1 1xy 1vu#010

Step 196 : 1 1xy 1vu#010

Step 197 : 1 1xy 1vu#010

Step 198 : 1 1xy 1vu#010

Step 199 : 1 1xy 1vu#010

Step 200 : 1 1xy 1vu#010

Step 201 : 1 1xy vvu#010

Step 202 : 1 1xy vvu#010

Step 203 : 1 1xy vvu#010

Step 204 : 1 1xy vvu#010

Step 205 : 1 0xy vvu#010

Step 206 : 1 1<u>0</u>xy vvu#010

Step 207 : 1 10<u>x</u>y vvu#010

Step 208 : 1 1<u>0</u>xy vvu#010

Step 209 : 1 1x<u>x</u>y vvu#010

Step 210 : 1 1xx<u>y</u> vvu#010

Step 211 : 1 1xxy_vvu#010

Step 212 : 1 1xxy <u>v</u>vu#010

Step 213 : 1 1xxy v<u>v</u>u#010

Step 214 : 1 1xxy vv<u>u</u>#010

Step 215 : 1 1xxy vvu<u>#</u>010

Step 216 : 1 1xxy vv<u>u</u>#010

Step 217 : 1 1xxy v<u>v</u>u#010

Step 218 : 1 1xxy <u>v</u>vu#010

Step 219 : 1 1xxy_vvu#010

Step 220 : 1 1xxy <u>v</u>vu#010

Step 221 : 1 1xxy 1<u>v</u>u#010

Step 222 : 1 1xxy 11<u>u</u>#010

Step 223 : 1 1xxy 110<u>#</u>010

Step 224 : 1 1xxy 11<u>0</u>#010

Step 225 : 1 1xxy 1<u>1</u>0#010

Step 226 : 1 1xxy <u>1</u>10#010

Step 227 : 1 1xxy_110#010

Step 228 : 1 1xx<u>y</u> 110#010

Step 229 : 1 1x<u>x</u>1 110#010

Step 230 : 1 1<u>x</u>01 110#010

Step 231 : 1 <u>1</u>001 110#010

Step 232 : 1_1001 110#010

Step 233 : <u>1</u> 1001 110#010

Step 234 : _0 1001 110#010

Step 235 : 1_0 1001 110#010

Step 236 : 10_1001 110#010

Step 237 : 10 _001 110#010

Step 238 : 10 x_01 110#010

Step 239 : 10 x0_1 110#010

Step 240 : 10 x01_110#010

Step 241 : 10 x01 _110#010

Step 242 : 10 x01 1_10#010

Step 243 : 10 x01 11_0#010

Step 244 : 10 x01 110_#010

Step 245 : 10 x01 110#_010

Step 246 : 10 x01 110#_x10

Step 247 : 10 x01 11_0#x10

Step 247 : 10 x01 1_10#x10

Step 248 : 10 x01 _110#x10

Step 249 : 10 x01_110#x10

Step 250 : 10 x0_1 110#x10

Step 251 : 10 x_01 110#x10

Step 252 : 10 _x01 110#x10

Step 253 : 10 x_01 110#x10

Step 254 : 10 xx_1 110#x10

Step 255 : 10 xx1_110#x10

Step 256 : 10 xx1 _110#x10

Step 257 : 10 xx1 1_10#x10

Step 258 : 10 xx1 11_0#x10

Step 259 : 10 xx1 110_#x10

Step 260 : 10 xx1 110#_x10

Step 261 : 10 xx1 110#x10

Step 262 : 10 xx1 110#x10

Step 263 : 10 xx1 110#x10

Step 264 : 10 xx1 110#x10

Step 265 : 10 xx1 110#x1

Step 266 : 10 xx1 110#x

Step 267 : 10 xx1 110#

Step 268 : 10 xx1 110

Step 269 : 10 xx1 11

Step 270 : 10 xx1 1

Step 271 : 10 xx1

Step 246 : 10 xx1

Step 246 : 10 xx1

Step 246 : 10 x01

Step 246 : 10 001

Step 246 : 10 001 –halted with acceptance

Since the 5 was divided by 2 hence the remainder obtained is 1 with quotient as 2.

STUDENT TAKEAWAY

By this project we learnt how to design a Turing Machine, how to think logically and find solution to a given problem statement. Since it was a team work we learnt how to work in a team.