VISTULA UNIVERSITY

ALGORITHMS AND COMPLEXITY

TURING MACHINE

Student : Nihat Allahverdiyev 46219

Teacher: Zaitsev Dmitry

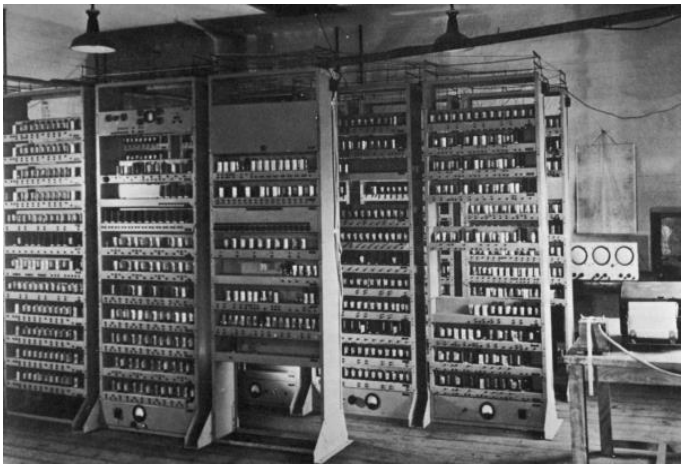
Erase odd letters in a alphabet {a,b,c}.

1.ALGORITHM

In mathematics and computer science, an **algorithm** is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing and automated reasoning tasks.

An algorithm is an effective method that can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function. Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input.

The concept of *algorithm* has existed for centuries; however, a partial formalization of what would become the modern *algorithm* began with attempts to solve the Entscheidungsproblem (the "decision problem") posed by David Hilbert in 1928. Subsequent formalizations were framed as attempts to define "effective calculability" or "effective method" ; those formalizations included the Gödel–Herbrand–Kleene recursive functions of 1930, 1934 and 1935, Alonzo Church's lambda calculus of 1936, Emil Post's "Formulation 1" of 1936, and Alan Turing's Turing machines of 1936–7 and 1939. Giving a formal definition of algorithms, corresponding to the intuitive notion, remains a challenging problem.



1.Convert first letter to # go to right

2.Convert present letter to numbers as “a” to “1”, “b” to “2” and “c” to “3” go to right

3.Shift letter to # go to left.

4.Change present number to # go to right

5.Convert present # to the previous number go to left until empty, if there is another number go to step 3

6. After finding empty go to right find first letter and do step 2 if there is no letter go to step 7

7. Go to right till finding empty

8. After finding empty go to left convert all numbers to letter as “1” to “a”, “2” to “b” and “3” to “c”

9.When find “#” sign go to left and delete all, and go to right till first letter

2.EXAMPLES AND TESTS

a) abca -> ba

b) abcba -> bba

c) abbaca -> baa

abca -> #bca -> #2ca -> #2#a -> ###a -> ##2a -> ##21 -> ##2a -> ##ba-> #ba-> ba.

3.PROGRAM

0 a # r q1

0 b # r q1

0 c # r q1

0 _ _ l q2

q1 a 1 r q2

q1 b 2 r q2

q1 c 3 r q2

q1 # # l q1

q1 _ _ r q7

q1 1 1 r q1

q1 2 2 r q1

q1 3 3 r q1

q2 a # l q3

q2 b # l q3

q2 c # l q3

q2 _ _ l q10

q3 1 # r q4

q4 # 1 l q8

q3 2 # r q5

q5 # 2 l q8

q3 3 # r q6

q6 # 3 l q8

q3 # # r q3

q8 # # l q8

q8 1 1 * q3

q8 2 2 * q3

q8 3 3 * q3

q8 _ _ r q9

q9 # # r q9

q9 1 1 r q1

q9 2 2 r q1

q9 3 3 r q1

q10 1 a l q10

q10 2 b l q10

q10 3 c l q10

q10 # _ l q10

q10 _ _ r q10

q10 a a * q11-halt

q10 b b * q11-halt

q10 c c * q11-halt

q7 1 1 * q10

q7 2 2 * q10

q7 3 3 * q10

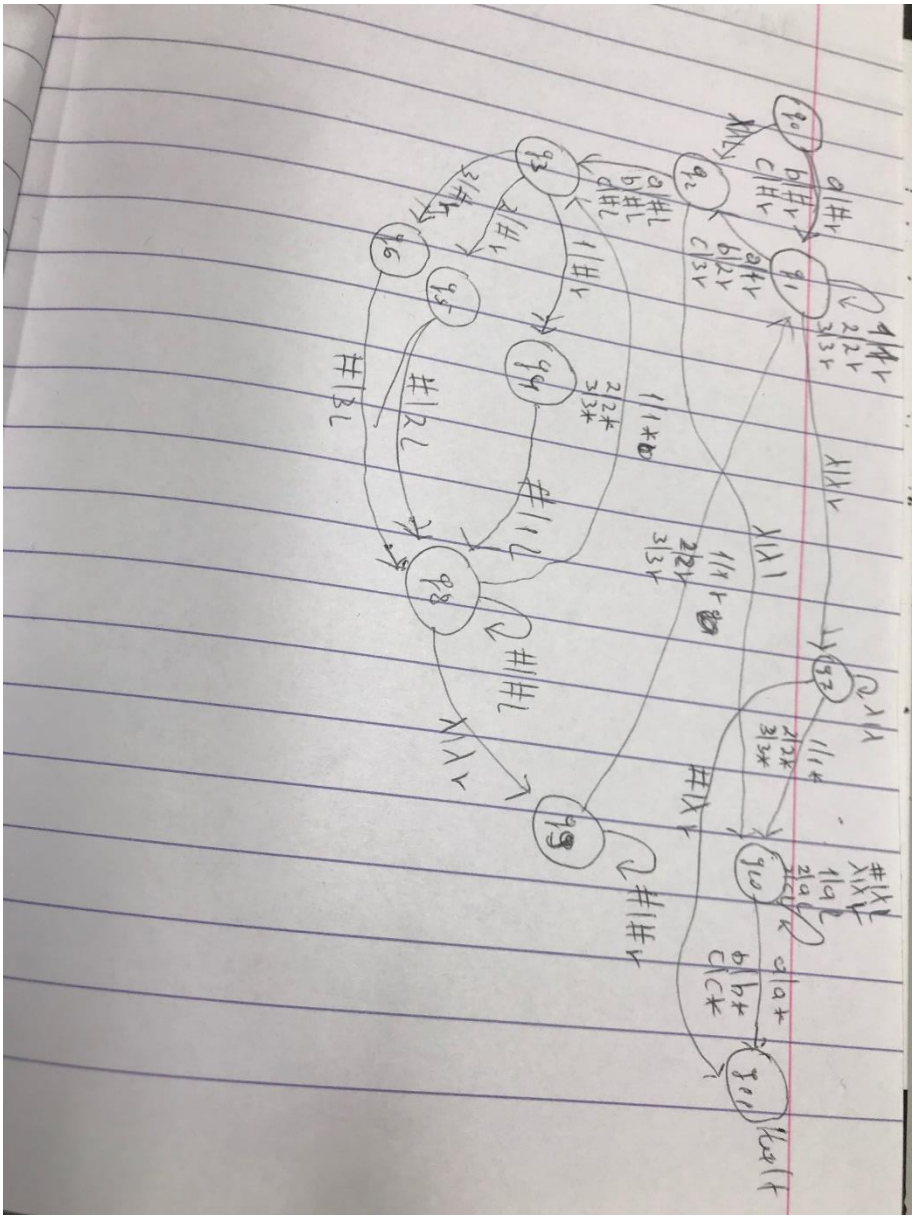
q7 _ _ l q7

q7 1 1 * q10

q7 2 2 * q10

q7 3 3 * q10

DIAGRAM.



CONCLUSION.

Whatever we obtain from user we delete odd letters from there and keep even letters.