Assignment Day- 8:

**Question 1**

A Barua number is a number which consists of only zeroes and ones and has only one 1. Barua number will start with 1. Given numbers, find out the multiplication of the numbers. Note: The input may contain one decimal number and all other Barua numbers. (Assume that each number is very large and total number of values give is also very large)

**Answer 1**

Let us consider that input is stored in array and n is total no. of elements in the array.

int i= 0, count= 0, multi=1;

while(a[i]!= ‘\0’) {

while(a[i]%10 == 0) {

count++;

a[i]= a[i]/10;

}

i++;

}

for(i= 0; i<n; i++)

multi= multi\*a[i];

print(multi);

for(i= 1; i<= count; i++)

printf("0");

**Question 2**

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

**Answer 2**

#include <bits/stdc++.h>

using namespace std;

struct MyStack

{

stack<int> s;

int minEle;

void getMin()

{

if (s.empty())

cout << "Stack is empty\n";

else

cout <<"Minimum Element in the stack is: "

<< minEle << "\n";

}

void peek()

{

if (s.empty())

{

cout << "Stack is empty ";

return;

}

int t = s.top();

cout << "Top Most Element is: ";

(t < minEle)? cout << minEle: cout << t;

}

void pop()

{

if (s.empty())

{

cout << "Stack is empty\n";

return;

}

cout << "Top Most Element Removed: ";

int t = s.top();

s.pop();

if (t < minEle)

{

cout << minEle << "\n";

minEle = 2\*minEle - t;

}

else

cout << t << "\n";

}

void push(int x)

{

if (s.empty())

{

minEle = x;

s.push(x);

cout << "Number Inserted: " << x << "\n";

return;

}

if (x < minEle)

{

s.push(2\*x - minEle);

minEle = x;

}

else

s.push(x);

cout << "Number Inserted: " << x << "\n";

}

};

int main()

{

MyStack s;

s.push(3);

s.push(5);

s.getMin();

s.push(2);

s.push(1);

s.getMin();

s.pop();

s.getMin();

s.pop();

s.peek();

return 0;

}