Data Structures and Algorithms

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)

Input 1: 100 10 12 1000

Output 1: 12000000

Input 2: 100 121 1000000000000000

Output 2: 12100000000000000000

Input 3: 10 100 1000

Output 3: 1000000

#include <iostream>

using namespace std;

int main()

{

unsigned long long int number, answer = 1;

while (cin >> number)

answer \*= number;

cout << answer;

return 0;

}

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

#include <iostream>

#include <stack>

using namespace std;

stack<int> s;

int minEle;

void min()

{

if (s.empty())

return;

else

cout << "Min element: " << minEle << endl;

}

void peek()

{

if (s.empty())

return;

int t = s.top();

(t < minEle) ? cout << "Top: " << minEle << endl : cout << "Top: " << t << endl;

}

void pop()

{

if (s.empty())

return;

int t = s.top();

s.pop();

if (t < minEle)

minEle = 2 \* minEle - t;

}

void push(int x)

{

if (s.empty())

{

minEle = x;

s.push(x);

return;

}

if (x < minEle)

{

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

minEle = x;

}

else

s.push(x);

}

int main()

{

int c;

do

{

cout << "0. Exit\n";

cout << "1. Push\n";

cout << "2. Pop\n";

cout << "3. Peek\n";

cout << "4. Get min\n";

cin >> c;

switch (c)

{

case 1:

int x;

cout << "Element: ";

cin >> x;

push(x);

break;

case 2:

pop();

break;

case 3:

peek();

break;

case 4:

min();

break;

}

} while (c);

return 0;

}