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.

Ans –

#include<stdio.h>

void main()

{

long int a,b,c,g,e,z=0,i,x;

printf("\nEnter numbers Containing one and zero - ");

scanf("%d %d %d",&a,&b,&c);

printf("\nEnter Decimal number - ");

scanf("%d",&g);

e=a\*b\*c;

printf("\n");

while(e>0)

{

x=e%10;

if(x==0)

{

z++;

}

e=e/10;

}

printf("%d",g);

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

printf("0");

}

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

Ans –

#include<iostream>

using namespace std;

struct Block {

int value, localMin;

};

class Stack {

private:

struct Block\* S;

int size, top;

public:

Stack(int);

void push(int);

void pop();

void min();

};

Stack::Stack(int n)

{

size = n;

S = new Block[n];

top = -1;

}

void Stack::push(int n)

{

if (top == size - 1) {

cout << "Stack is full" << endl;

}

else {

top++;

if (top == 0) {

S[top].value = n;

S[top].localMin = n;

}

else {

if (S[top - 1].localMin < n) {

S[top].value = n;

S[top].localMin = S[top - 1].localMin;

}

else {

S[top].value = n;

S[top].localMin = n;

}

}

cout << n << " inserted in stack" << endl;

}

}

void Stack::pop()

{

if (top == -1) {

cout << "Stack is empty" << endl;

}

else {

top--;

cout << "Element popped" << endl;

}

}

void Stack::min()

{

if (top == -1) {

cout << "Stack is empty" << endl;

}

else {

cout << "Minimum value in the stack: "

<< S[top].localMin << endl;

}

}

int main()

{

Stack S1(5);

S1.push(2);

S1.min();

S1.push(6);

S1.min();

S1.pop();

S1.min();

return 0;

}