Assignment Day 8 | 2nd December 2020

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)

Input 1: 100 10 12 1000

Output 1: 12000000

Input 2: 100 121 10000000000000000

Output 2: 121000000000000000000

Input 3: 10 100 1000

Output 3: 1000000

c-code:

```
#include<iostream>
using namespace std;
int main()
{
	int n, count=0; cin>>n;
	long long int a[n];
	for(int i=0; i<n; i++)
		cin>>a[i];
	for(int i=0; i<n; i++)
	{
		while(a[i]%10==0)
		{
			count++;
			a[i]/=10;
```

```
            }
        }
        long long int prod=1;
        for(int i=0; i<n; i++)
            prod*=a[i];
        cout<<prod;
        for(int i=1; i<=count; i++)
            cout<<0;
}
```

Question 2

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

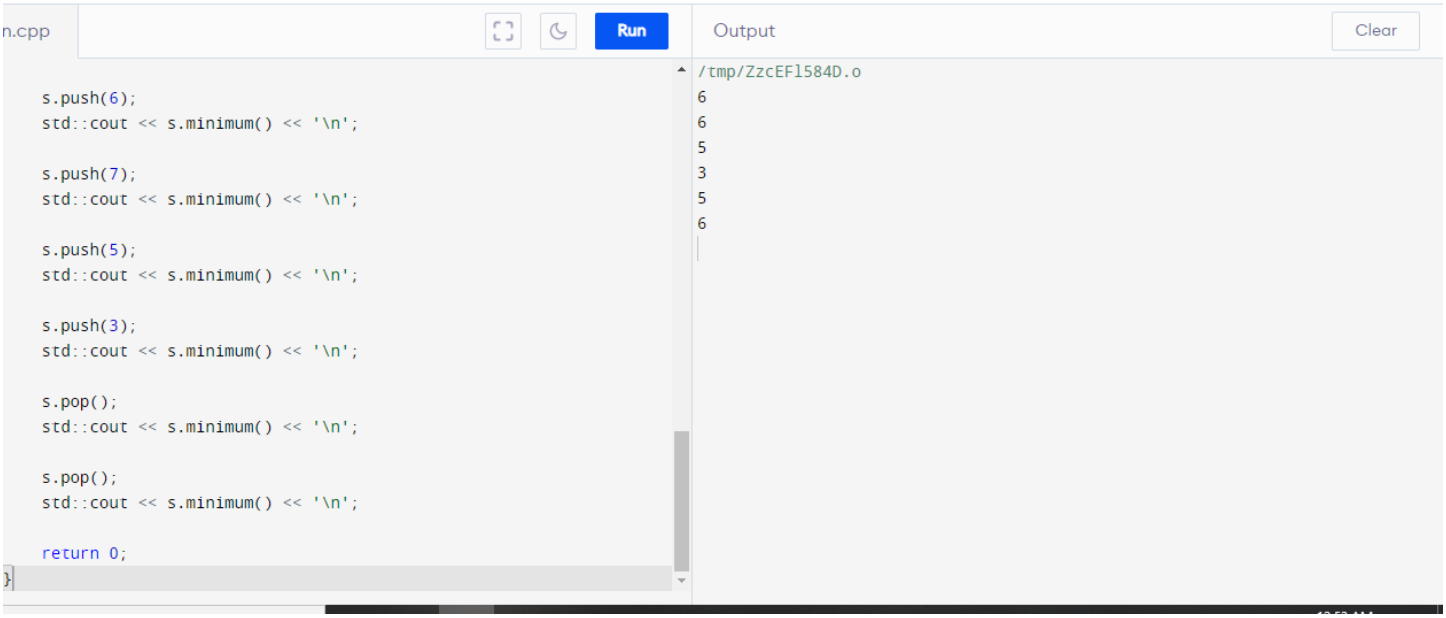
C-code:

```
#include <iostream>
#include <stack>

class Stack
{
   // main stack to store elements
   std::stack<int> s;

   // variable to store minimum element
   int min;

public:

   // Inserts a given element on top of the stack
   void push(int x)
   {
      if (s.empty()) {
         s.push(x);
         min = x;
      }
      else if (x > min) {
         s.push(x);
      }
```

```
        else {
            s.push(2 * x - min);
            min = x;
        }
    }

    // Removes top element from the stack and returns it
    void pop()
    {
        if (s.empty()) {
            std::cout << "Stack underflow!!" << '\n';
        }

        int top = s.top();
        if (top < min)
            min = 2 * min - top;
        s.pop();
    }

    // Returns the minimum element from the stack in constant time
    int minimum()
    {
        return min;
    }
};

int main()
{
    Stack s;

    s.push(6);
    std::cout << s.minimum() << '\n';

    s.push(7);
    std::cout << s.minimum() << '\n';

    s.push(5);
    std::cout << s.minimum() << '\n';

    s.push(3);
    std::cout << s.minimum() << '\n';

    s.pop();
    std::cout << s.minimum() << '\n';

    s.pop();
    std::cout << s.minimum() << '\n';

    return 0;
```

}

Output:


n.cpp
Run
Output
Clear
s.push(6);
std::cout << s.minimum() << '\n';
s.push(7);
std::cout << s.minimum() << '\n';
s.push(5);
std::cout << s.minimum() << '\n';
s.push(3);
std::cout << s.minimum() << '\n';
s.pop();
std::cout << s.minimum() << '\n';
s.pop();
std::cout << s.minimum() << '\n';
return 0;
}
/tmp/ZzcEFl584D.o
6
6
5
3
5
6