Assignment Day-6:

Question 1 Write a function to find the maximum element in the stack.

#include <bits/stdc++.h>

using namespace std;

struct MyStack {

stack<int> s;

int maxEle;

void getMax()

{

if (s.empty())

cout << "Stack is empty\n";

else

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

<< maxEle << "\n";

}

void peek()

{

if (s.empty()) {

cout << "Stack is empty ";

return;

}

int t = s.top(); // Top element.

cout << "Top Most Element is: ";

(t > maxEle) ? cout << maxEle : 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 > maxEle) {

cout << maxEle << "\n";

maxEle = 2 \* maxEle - t;

}

else

cout << t << "\n";

}

void push(int x)

{

if (s.empty()) {

maxEle = x;

s.push(x);

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

return;

}

if (x > maxEle) {

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

maxEle = x;

}

else

s.push(x);

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

}

};

int main()

{

MyStack s;

s.push(3);

s.push(5);

s.getMax();

s.push(7);

s.push(19);

s.getMax();

s.pop();

s.getMax();

s.pop();

s.peek();

return 0;

}

Question 2 Write a function to find the minimum element in the stack.

#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(); // Top element.

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;

}