Assignment No 8 Data Structures and Algorithms

**Question 2:**

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

Answer:

#include<stdio.h>

#include<conio.h>

int main\_stack[100] , supporting\_stack[100];

int push(int element , int \*top , int \*stack)

{

\*top = \*top + 1;

stack[\*top] = element;

}

int pop(int \*stack , int \*top)

{

int element;

if(\*top > -1)

{

element = stack[\*top];

\*top = \*top - 1;

return element;

}

else {

printf("\n== STACK EMPTY == \n");

return -99999; // means nothing is popped

}

}

int main()

{

int choice , element , top\_main=-1 , top\_support=-1 ,i ,supp\_stack\_pop\_element, popped\_element;

printf("Enter the operation : \n 1. Push \n 2.Pop \n 3.check minimum \n 4.STOP \n");

scanf("%d",&choice);

while(choice != 5){

if (choice == 1){

printf("\nEnter the number to be pushed");

scanf("%d",&element);

push(element , &top\_main , main\_stack);

if (top\_support >= 0 && element < supporting\_stack[top\_support])

{

push(element , &top\_support , supporting\_stack);

}

else if (top\_support == -1)

{

push(element , &top\_support , supporting\_stack);

}

}

else if (choice == 2)

{

popped\_element = pop(main\_stack , &top\_main);

if (popped\_element != -99999)

printf("\n Popped element = %d",popped\_element);

if (popped\_element != -99999){

if (popped\_element == supporting\_stack[top\_support]){

supp\_stack\_pop\_element = pop(supporting\_stack , &top\_support);

}

}

}

else if( choice == 3)

{

if (top\_support > -1)

printf("\nMinimum element in the stack = %d \n\n" , supporting\_stack[top\_support] );

else

printf("\n ==== Stack Empty ======");

}

else if (choice == 4)

{

if (top\_main > -1)

{

printf("\n === Main Stack === \n ");

for (i=top\_main;i>=0;i--)

{

printf("\n%d",main\_stack[i]);

}

}

else

printf(" \n ==== STACK EMPTY === \n ");

printf("\n top\_support = %d", top\_support);

for (i=top\_support;i>=0;i--)

{

printf("\n%d",supporting\_stack[i]);

}

}

printf("\n\nEnter the operation : \n 1. Push \n 2.Pop \n 3.check minimum \n 4.See Full Stack \n 5.STOP \n");

scanf("%d",&choice);

}

return 0;

}