Camlin Page
Date / /

Q Implement push, pop and find the minimum element in a stack in O(1). Time Complexity?

```
#include<stdio.h>
#include<conio.h>

int stack[100], support 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 -9999; // means nothing is
```

popped.
}
}
int main ()
{
int choice, Element, top_main = -1;
int top_dupport = -1, i, dupp_pop_element;
int pop_element;

printf ("Enter the operation \n");
Printf ("\n 1. push \n 3. pop \n 4. check main \n \n");

Print f ("4. Stop \n");
scan f ("%d", &choice);
while (choice != 5)
{
if (choice == 1)
{
printf ("\n Enter num");
scanf ("%d", & Element);

push (Element, & top_main, main_stack);

```
if (top_support >= 0 && Element < support
                                stack[top_support])
{
push(Element, &top_support, support_stack);
}
else if (top_support == -1)
{
push(element, &top_support, support_stack);
}
else if (choice == 2)
{
pop_element = pop(stack, &top_main);
if (pop_element != -99999)
{
if (pop_element == support_stack[top_support])
{
supp_pop_element = pop(support_stack, &top_support);
}
}
else if (choice == 3)
{
if (top_support > -1)
```

```
printf("\n Min Element = %d", support_
                        stack[top_support]);
Else
Printf("\n STACK EMPTY");
}
else if (choice == 4)
{
if (top_main > -1)
{
Printf("\n MAIN STOCK \n");
for(i, top_main; i >= 0; i--)
{
Printf("\n%d", stack[i]);
}
}
Else
Printf("\n STACK EMPTY");
Printf("\n top_support = %d", top_support);
for(i, top_support; i >= 0; i--)
```

```
{
printf("\n%d", duppost_stack[i]);
}
}

printf("\n Enter the operation :\n 1. push\n");
printf("\n 2. pop\n 3. check minimum\n 4. see full
                                    stack\n");
printf("\n 5. Stop\n");
scanf("%d", &choice);
}
return 0;
}
```