

LAB PROGRAM - 1

23-09-2020

1) program to simulate working of stack using an array

- a) pop
- b) Display
- c) push

The program should print appropriate message for stack overflow and underflow:

```
#include <stdio.h>
int stack[100], choice, n, top, x, i;
void push(void);
void pop(void);
void display(void);
int main()
{
    clrscr();
    top = -1;
    printf("\nEnter the size of STACK [MAX=100]: ");
    scanf("%d", &n);
    printf("\n\t STACK OPERATIONS USING ARRAY\n");
    printf("\n\t --- --\n");
    printf("\n\t 1. PUSH\n\t 2. POP\n\t 3. DISPLAY\n\t 4. EXIT\n");
    do
    {
        printf("\nEnter the choice: ");
        scanf("%d", &choice);
        switch (choice)
```

}

case 1:

{ push();

break;

}

case 2:

{

pop();

break;

}

case 3:

{

display();

break;

}

case 4:

{

printf ("int EXIT POINT");

break;

}

default;

}

default:

{

printf ("n\t Please Enter a valid choice (1/2/3/4)");

}

}

}

while (choice != 4);

return 0;

}

void push()

```
{  
    if (top >= n-1)  
    {  
        printf ("m1t STACK is over flow");  
    }  
    else  
    {  
        printf ("Enter a value to be pushed : ");  
        scanf ("%d", &x);  
        top++;  
        stack[top] = x;  
    }  
}
```

void pop()

```
{  
    if (top <= -1)  
    {  
        printf ("m1t Stack is under flow");  
    }  
    else  
    {  
        printf ("m1t The popped elements is : %d", stack[top]);  
        top--;  
    }  
}
```

void display()

```
{  
    if (top >= 0)
```

```

}
printf ("In The elements in STACK\n");
for (i=top; i>=0; i--)
    printf ("%d", Stack[i]);
printf ("In Press Next choice");

}
else
{
    printf ("In The STACK is empty");
}
}

```

2) STACK [using parameters and pointers]

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

```
#include <process.h>
```

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

```
#define STACK_SIZE 5
```

```
int top = -1;
```

```
void push (int item, int S[], int *top)
```

```
{
```

```
if (*top == STACK_SIZE - 1)
```

```
{
```

```
printf ("Stack overflow\n");
```

```
return;
```

```
}
```

```
*top = *top + 1;
```

```
S[*top] = item;
```

```
}
```

```
int pop (int S[], int *top)
```

```
{
```

```
int item_deleted;
```

```
if (*top == -1)
```

```
{
```

```
printf ("Stack underflow, cannot delete\n");
```

```
return 0;
```

```
}
```

```
item_deleted = S[*top];
```

```
*top = *top - 1;
```

```
return item_deleted;
```

```
}
```

```
void display (int top, int S[])
```

```
{
```

```
int i;
```

```
if (top == -1)
```

```
{
```

```
printf ("Stack is empty\n");
```

```
return;
```

```
}
```

```
printf ("Contents of the stack\n");
```

```
for (i = 0; i <= top; i++)
```

```
{
```

```
printf ("%d\n", s[i]);
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
int item, s[10];
```

```
int item_deleted;
```

```
int choice;
```

```
clear();
```

```
for (;)
```

```
{
```

```
printf ("1: push 2: pop 3: display 4: exit\n");
```

```
printf ("Enter the choice\n");
```

```
scanf ("%d", &choice);
```

```
switch (choice)
```

```
{
```

```
case 1: printf ("enter the item to be inserted\n");
```

```
scanf ("%d", &item);
```

```
push (item, s, &top);
```

```
break;
```

```
case 2: item-deleted = pop(s, &top);  
        if (item-deleted != 0)  
            printf("Item deleted is %d\n", item-deleted);  
        break;
```

```
case 3: display(top, s);
```

```
        break;
```

```
        default : exit(0);
```

```
    }
```

```
}
```

```
getch();
```

```
}
```