

Write a program to simulate the working of stack using an array with the following

a). Push b). Pop c). Display

The program should print appropriate messages for stack overflow, stack underflow.

classmate

Date _____

Page _____

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

```
#include <stdlib.h>
```

```
#define MAX 10
```

```
int top = -1, stack[MAX];
```

```
void push();
```

```
void pop();
```

```
void display();
```

```
int main()
```

```
{
```

```
    int ch;
```

```
    while(1)
```

```
{
```

```
    printf("\n** STACK MENU **");
```

```
    printf("\n\n1. PUSH \n2. POP \n3. DISPLAY \n4. EXIT");
```

```
    printf("\n\n ENTER YOUR CHOICE (1-4): ");
```

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

```
    switch(ch)
```

```
{
```

```
    case 1: push();
```

```
        break;
```

```
    case 2: pop();
```

```
        break;
```

```
    case 3: display();
```

```
        break;
```

```
    case 4: exit(0);
```

```
    default: printf("\n Wrong choice");
```

```
}
```

```
} // loop continues for next input till user enters exit(0)
```

```
between 0 & stack[10] - 1 will be
```

```
}
```

```
void push()
```

```
{ if (top == MAX - 1) { printf("Stack Overflow"); return; }
```

```
else
```

```
top++;
```

```

int val;
if (top == MAX-1)
{ printf ("\n STACK IS FULL");
}
else
{ printf ("\n ENTER ELEMENTS TO PUSH:");
scanf ("%d", &val);
top = top + 1;
stack [top] = val;
}
void pop ()
{
if (top == -1)
{ printf (" \n STACK IS EMPTY");
}
else
{ printf ("\n DELETED ELEMENT IS %d", stack [top]);
top = top - 1;
}
}
void display ()
{
int i;
if (top == -1)
{ printf ("\n STACK IS EMPTY");
}
else
{ printf ("\n STACK IS ...");
for (i = top; i >= 0; i--)
printf ("\n %d", stack [i]);
}
}

```

OUTPUT

** STACK MENU **

1. PUSH
2. POP
3. DISPLAY
4. EXIT

ENTER YOUR CHOICE (1-4) : 2

STACK IS EMPTY

** STACK MENU **

1. PUSH
2. POP
3. DISPLAY
4. EXIT

ENTER YOUR CHOICE (1-4) : 1

ENTER ELEMENTS TO PUSH : 5

** STACK MENU **

1. PUSH
2. POP
3. DISPLAY
4. EXIT

ENTER YOUR CHOICE (1-4) : 1

ENTER ELEMENTS TO PUSH : 65

** STACK MENU **

1. PUSH
2. POP
3. DISPLAY
4. EXIT

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX 10
4 int top=-1,stack[MAX];
5 void push();
6 void pop();
7 void display();
8
9 int main()
10 {
11     int ch;
12     while(1)
13     {
14         printf("\n**STACK MENU**");
15         printf("\n\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT");
16         printf("\n\nENTER YOUR CHOICE(1-4):");
17         scanf("%d",&ch);
18
19         switch(ch)
20         {
21             case 1:push();
22                     break;
23
24             case 2: pop();
25                     break;
26             case 3: display();
27                     break;
28             case 4: exit(0);
29             default: printf("Wrong choice");
```



```
}

}

return 0;
}

void push()
{
    int val;
    if(top==MAX-1)
    { printf("\nSTACK IS FULL");
    }
    else
    { printf("\n ENTER ELEMENTS TO PUSH:");
        scanf("%d",&val);
        top=top+1;
        stack[top]=val;
    }
}

void pop()
{
    if(top==-1)
    { printf("\nSTACK IS EMPTY");
    }
    else
    { printf("\nDELETED ELEMENT IS %d",stack[top]);
        top=top-1;
    }
}

void display()
{
    int i;
```

```
1.c
```

```
void display()
{
    int i;
    if(top == -1)
    { printf("\n STACK IS EMPTY");
    }
    else
    { printf("\n STACK IS... ");
        for(i=top; i>=0; --i)
            printf("%d\n", stack[i]);
    }
}
```

```
• clang-7 -pthread -lm -o main main.c
• ./main
```

STACK MENU

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

ENTER YOUR CHOICE(1-4) :2

STACK IS EMPTY

STACK MENU

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

ENTER YOUR CHOICE(1-4) :1

ENTER ELEMENTS TO PUSH:3

STACK MENU

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT



ENTER YOUR CHOICE(1-4) :3

STACK IS...3