

- 1) Write a program to simulate the working of stack using an array with the following:
 - a) Push
 - b) Pop
 - c) DisplayThe program should print appropriate messages for stack overflow, stack underflow.

```
#include<stdio.h>
int size= 5;
int stack[5];
int top = -1;
void push(int item)
{
    if(top == size-1)
    {
        printf("The stack is overflowed!!!");
        return;
    }
    else{
        top +=1;
        stack[top] = item;
    }
}
void pop()
{
    if(top == -1)
    {
        printf("The stack is underflowed!!!");
        return ;
    }
    else{
        int ele = stack[top];
        top -= 1;
        printf("The element which was popped is %d",ele);
    }
}
void display()
{
    if(top == -1)
    {
        printf("Nothing to display, Stack is empty!");
    }
    else{
        printf("The stack from top to bottom is :");
        for(int i = top;i>=0;i--)
        {
            printf("%d, ",stack[i]);
        }
    }
}
int getPeek()
```

```

{
    if(top == -1)
    {
        printf("The stack is empty!!");
        return ;
    }
    else{
        printf("The peek element is %d ",stack[top]);
    }
}
int main()
{
    while(1)
    {
        printf("\n\nChoose any of the following operations : ");
        printf("\n1.PUSH ");
        printf("\n2.POP");
        printf("\n3.DISPLAY ");
        printf("\n4.PEEK ");
        printf("\n5.EXIT \n");

        int choice ;
        scanf("%d",&choice);
        if(choice == 1)
        {
            int item;
            printf("Enter the element you want to push : \n");
            scanf("%d",&item);
            push(item);
            printf("%d is added to the stack.",item);
        }
        else if(choice == 2)
        {
            pop();
        }
        else if(choice == 3)
        {

            display();
        }
        else if(choice == 5)
        {
            printf("Exiting the program !!");
            break;
        }
        else if(choice == 4)
        {
    }
}

```

```

        getPeek());
    }
else{
    printf("Enter a valid choice");
}

}

return 0;
}

```

OUTPUT :

```

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
1
Enter the element you want to push :
10
10 is added to the stack.

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
1
Enter the element you want to push :
20
20 is added to the stack.

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
1
Enter the element you want to push :
30
30 is added to the stack.

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
3
The stack from top to bottom is :30, 20, 10,

```

```

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
4
The peek element is 30

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
2
The element which was popped is 30

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT
2
The element which was popped is 20

Choose any of the following operations :
1.PUSH
2.POP
3.DISPLAY
4.PEEK
5.EXIT

```