Question

Design and implement a stack (Array implementation/ Linked list implementation) and demonstrate its working with necessary inputs. Display the appropriate messages in case of exceptions

Aim: To Implement Stacks using Arrays

Algorithm for Push:

1. Check whether the stack is full by checking the value of top.
2. If it's full, throw an exception saying that the stack is full.
3. If it isn't full, then increment the value of top and assign the value to stack[top].

Algorithm for Pop:

1. Check whether the stack is empty by checking the value of top.
2. If it's empty,throw an exception saying that the stack is empty.
3. If it isn't empty, then assign the value of stack[top] to some variable and decrement top.

Algorithm for Display:

1. Check whether the stack is empty by checking the value of top.
2. If it's empty, then display a message that the stack is empty.
3. If not, then point at top and print the value
4. Now decrement the pointer and repeat step 3 till the pointer points at the 0th address

Program:

```
#include<stdio.h>
#include<stdlib.h>
int stack[100],choice,n,top,x,i;
top=-1;
void push();
void pop();
void display();
int main()
{
    printf("\n Enter the size of stack[MAX=100]:");
    scanf("%d",&n);
    printf("\n\t--------------------------------");
    printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
    while(1)
     {
        printf("\n Enter the Choice:"); //Performs functions as per user
input
```

```
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display();
                break;
            }
            case 4:
            {
                exit(1);
                break;
            }
            default:
            {
                printf ("\n invalid");
            }

        }
    }
    return 0;
}
void push()  //Push function
{
    if(top>=n-1)
    {
        printf("\n stack overflow");

    }
    else
```

```
    {
        printf(" enter a value to be pushed:");
        scanf("%d",&x);
        top++;
        stack[top]=x;
    }
}
void pop()  //Pop function
{
    if(top<=-1)
    {
        printf("\n\t stack underflow");
    }
    else
    {
        printf("\n\t the popped element is %d",stack[top]);
        top--;
    }
}
void display() //Displays all the elements in the stack
{
    if(top>=0)
    {
        printf("\n The elements in stack \n");
        for(i=top; i>=0; i--)
            printf("\n%d",stack[i]);
    }
    else
    {
        printf("\n The stack is empty");
    }

}
```

Output

```
Enter the size of stack[MAX=100]:40

       --------------------------------
        1.PUSH
        2.POP
        3.DISPLAY
        4.EXIT
Enter the Choice:2

        stack underflow
Enter the Choice:1
enter a value to be pushed:2

Enter the Choice:1
enter a value to be pushed:4

Enter the Choice:3

The elements in stack

4
2
Enter the Choice:2

        the popped element is 4
Enter the Choice:3

The elements in stack

2
Enter the Choice:
The elements in stack

2
Enter the Choice:
The elements in stack

2
Enter the Choice:
```