Program 4 - Stack

Q. Implement an algorith (menu - driven) for a stack, with seperate functions for push, pop and peek.

Algorithm

- Algorithm for push (int n, int arr[], int cu

1. START
2. Take input n which is the total size of the array.
3. Take input array[] ~~which is the~~
4. Take ~~cur~~ input curr // top of the stack.
5. if curr is less than n, then:
   - 5.1 int var ← take the value to push
   - 5.2 array[curr++] ← val
6. else print "stack Overflow".

7. return cur

8. STOP

- Algorithm for pop

1. START
2. input the size as n // no need.
3. input the array
4. input the current top of stack as cur.
5. if current top is greater than 0 then decrease the current top by 1.
6. else print "stack underflow".
7. return current top

- Algorithm for peek

1. START
2. input the array.
3. input the current top of stack.

4. initial a variable ← i = curr-1

5. while i is greater than or equal to 0, do:
   5.1 Print array @ element.
   5.2 decrement i

6. STOP

- Algorithm for main().

1. START
2. Initialise variables with type int ← ~~total_size~~ n, array [50], curr = 0
3. Input the size of stack and store it in ~~total_size~~ n. You can also initial the array to be of the same size.
4. Display the following:
   0 to exit
   1 to push
   2 to pop
   3 to peek

5. Initialise a variable as 'ans' with int data type and ~~take~~ set the value ~~from user~~ as 1.

6. while ~~Ex~~ ans is not 0. // ans != 0

6.1 Take the value from user and store it to ans.

6.2. Use switch (ans) to call the functions.

6.2.1 Case 0: ~~break;~~ print. exiting and break.

6.2.2. Case 1: Call function push and assign the value returned to current top // curr and break.

6.2.3. Case 2: Call pop and store the value return to curr. and break.

6.2.4. Call display function

6.2.5. Default case :-

Try 0, 1, 2, 3

7. STOP.

```
#include <stdio.h>

//inserting into a stack
int push(int n, int array[],int curr){
    int i,val;
    if(curr<n){
        printf("value to insert: ");
        scanf("%d",&val);
        array[curr]=val;
        curr++;
    }
    else
        printf("---Stack Overflow---\n");
    return curr;
}

//deleting an element from stack
int pop(int n,int array[],int curr){
    if(curr>0){
        int num;
        curr--;
        //num=array[curr];
        printf("Top is poped");
    }
    else
        printf("---Stack Underflow---\n");
    return curr;
}

//displaying the stack elements
void display(int n,int array[],int curr){
    int i;
        printf("Current Stack--> ");
        for(i=curr-1;i>0;i--){
                printf("%d , ",array[i]);
        }printf("%d \n",array[i]);
}

int main(){
    int ans=1;
    printf("Enter the total size of the Stack : ");
    int n;
    scanf("%d",&n);
    int array[100];
    int curr=0;
    printf("\n**press** \n0 --> exit\n1 --> push \n2 --> pop , \n3 --> peek
\n***************\n");
    while(ans!=0){
        printf("\nMenu ###: ");
        scanf("%d",&ans);
```

```
        switch(ans){
            case 0:break;
            case 1:curr=push(n,array,curr);
                break;
            case 2:curr=pop(n,array,curr);
                break;
            case 3:display(n,array,curr);
                break;
            default:printf("$#*&  Wrong Input  &*#$");
        }
  }
  printf("\n****************Exiting the stack***************");
}
```

```
Enter the total size of the Stack : 3

**press**
0 --> exit
1 --> push
2 --> pop ,
3 --> peek
**********

Menu ###: 2
---Stack Underflow---

Menu ###: 1
value to insert: 5

Menu ###: 1
value to insert: 6

Menu ###: 3
Current Stack--> 6 , 5

Menu ###: 1
value to insert: 9

Menu ###: 1
---Stack Overflow---

Menu ###: 2
Top is poped
Menu ###: 3
Current Stack--> 6 , 5

Menu ###: 7
$#*&  Wrong Input  &*#$
Menu ###: 3
Current Stack--> 6 , 5

Menu ###: 2
Top is poped
Menu ###: 2
Top is poped
Menu ###: 2
---Stack Underflow---

Menu ###: 0

**********Exiting the stack**********
```