Program 3 -

Stack Implementation

a) Push
b) Pop
c) Display

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

structure stack {
        int arr;
        int top;
        int capacity;
        int i;
};

void initialize (structure stack stack *stack
                int capacity ) {
    stack -> arr = (int) malloc (capacity * sizeof
    stack -> top = -1;
    stack -> capacity = capacity;
```

```
int isEmpty (structure stack * stack) {
    return (stack -> top == -1);
}


{
  int isfull (structure stack * stack) {
    return (stack -> top == stack -> capicity -1);
void push (structure stack * stack, int value) {
   if (isfull (stack)) {
   printf ("stack overflow. Cannot push, value);
   }
  else {
     stack = arr[++ stack => top ] = value;
     print (" %d pushed in stack. ", value;
   }
}


int pop (structure stack * stack) {
      if (isEmpty (stack)) {
t));  printf ("stack underflow ");
      return poppedValue;
   }
}
```

```
void display (structure stack * stack ){
  if (is Empty (stack)) {
     printf ("stack is empty ");
    }
    else {
      printf ("Elements of the stack :)
      for (i =0 ; i<= stack ->arr[i]);
     }
      printf ("\n");
    }
}

int main () {
      printf ("Rohit. Gandhi ");
      printf ("My USN is 2023BMS02599");

int capacity ;
    printf (("Enter no of elements to push
              in stack ");
    Scanf ("%d" ,&numelements);

for (int i =0 ; i< numelements ; ++i){
     int element ;
    printf("Enter element %d :, i+1);
    scanf ("%d", &element);

      display (&mystack);
  }

  pop(&mystack);

  display (&mystack);

  free (mystack.arr);

  return 0;
}
```

Output:-

```
Enter the capacity of the stack : 5

Enter the no of elements to push : 4

Enter element 1 : 10
10 pushed to the stack

Enter element 2 : 20
```

```
20 pushed in stack

Enter element 3: 30
30 pushed in stack

Enter element 4: 40
40 pushed in the stack

Enter element 5: 20
stack overflow cant push 20.
Enter the pop element: 10
10 popped from the stack.

Elements of the stack:
 20, 30, 40
```

21/11/2023.