MULTIPOP

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

struct stack
{
   int top;
   int size;
   int *arr;
};

int isFull(struct stack *sp)
{
   if (sp->top == sp->size - 1)
   {
      return 1;
   }
   else
   {
      return 0;
   }
}

int isEmpty(struct stack *sp)
{
   if (sp->top == -1)
   {
      return 1;
   }
   else
   {
```

```
        return 0;
    }
}

void Push(struct stack *sp, int data)
{
    if (isFull(sp))
    {
        printf("Stack Overflow\n");
    }
    else
    {
        sp->top++;
        sp->arr[sp->top] = data;
    }
}

int Pop(struct stack *sp)
{
    if (isEmpty(sp))
    {
        printf("Stack Underflow\n");
    }
    else
    {
        int data = sp->arr[sp->top];
        sp->top--;
        return data;
    }
}
```

```
void multiPop(struct stack *sp, int k)
{
   while (!isEmpty(sp) && k > 0)
   {
      int number = Pop(sp);
      k = k - 1;
      printf("Element: %d\n", number);
   }
}

int main()
{
   struct stack *sp = (struct stack *)malloc(sizeof(struct stack));
   sp->top = -1;
   int sz;
   printf("Enter the size of the stack: ");
   scanf("%d", &sz);
   sp->size = sz;
   sp->arr = (int *)malloc(sp->size * sizeof(int));
   int num, n;
   printf("Enter how many times do you want to push operation: ");
   scanf("%d", &num);
   printf("Push elements in the stack:-\n");
   for (int i = 0; i < num; i++)
   {
      printf("Enter %d element: ", i + 1);
      scanf("%d", &n);
      Push(sp, n);
   }
   int nmbr;
   printf("Enter how many elements do you want to pop: ");
```

```
    scanf("%d", &nmbr);
    multiPop(sp, nmbr);
    return 0;
}
```

OUTPUT

```
Enter how many times do you want to push operation: 5
Push elements in the stack:-
Enter 1 element: 6
Enter 2 element: 1
Enter 3 element: 9
Enter 4 element: 3
Enter 5 element: 2
Enter how many elements do you want to pop: 3
Element: 2
Element: 3
Element: 9
```