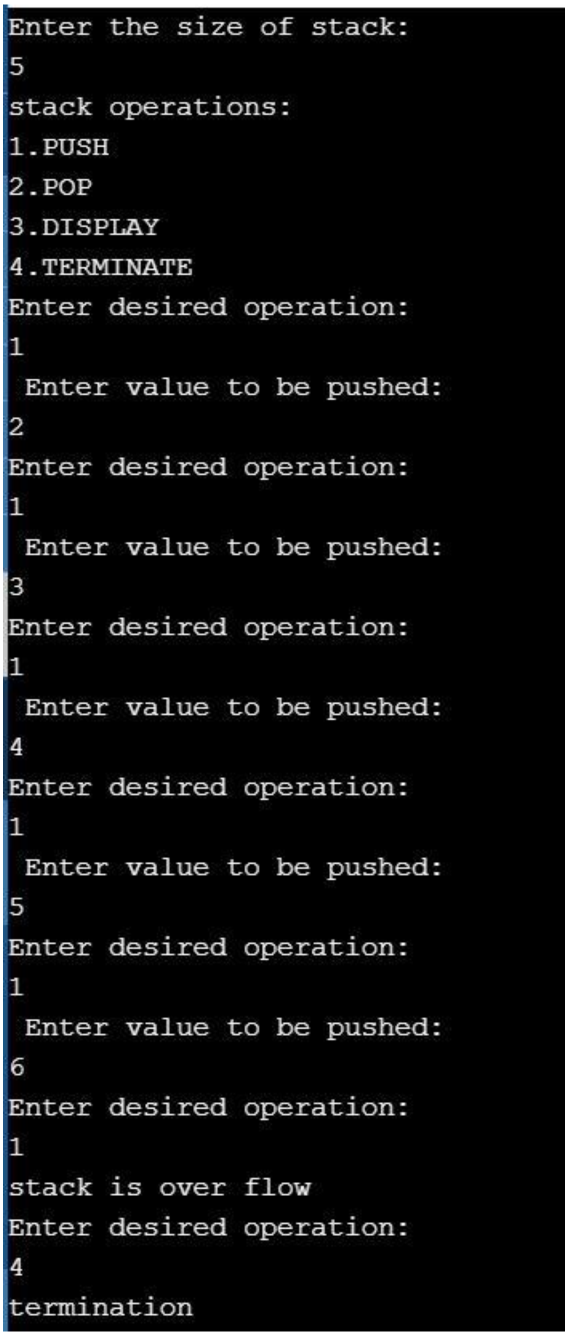
```
#include<stdio.h>
int stack[10],operation,n,top,x,i;
void push(void);
void pop(void);
void display(void);
int main()
{
   top=-1;
   printf("Enter the size of stack: \n");
   scanf("%d",&n);
   printf("stack operations: \n");
   printf("1.PUSH\n");
   printf("2.POP\n");
   printf("3.DISPLAY\n");
   printf("4.TERMINATE\n");
   do
   {
      printf("Enter desired operation:\n");
      scanf("%d",&operation);
      switch(operation)
      {
         case 1:
         {
            push();
            break;
         }
         case 2:
         {
            pop();
            break;
         }
```

```
            case 3:

            {

               display();

               break;

            }

            case 4:

            {

               printf("termination ");

               break;

            }

            }



      }

      while(operation!=4);

      return 0;

   }

   void push()

   {

      if(top>=n-1)

      {

         printf("stack is over flow\n");


      }

      else

      {

         printf(" Enter value to be pushed:\n");

         scanf("%d",&x);

         top++;

         stack[top]=x;

      }
```

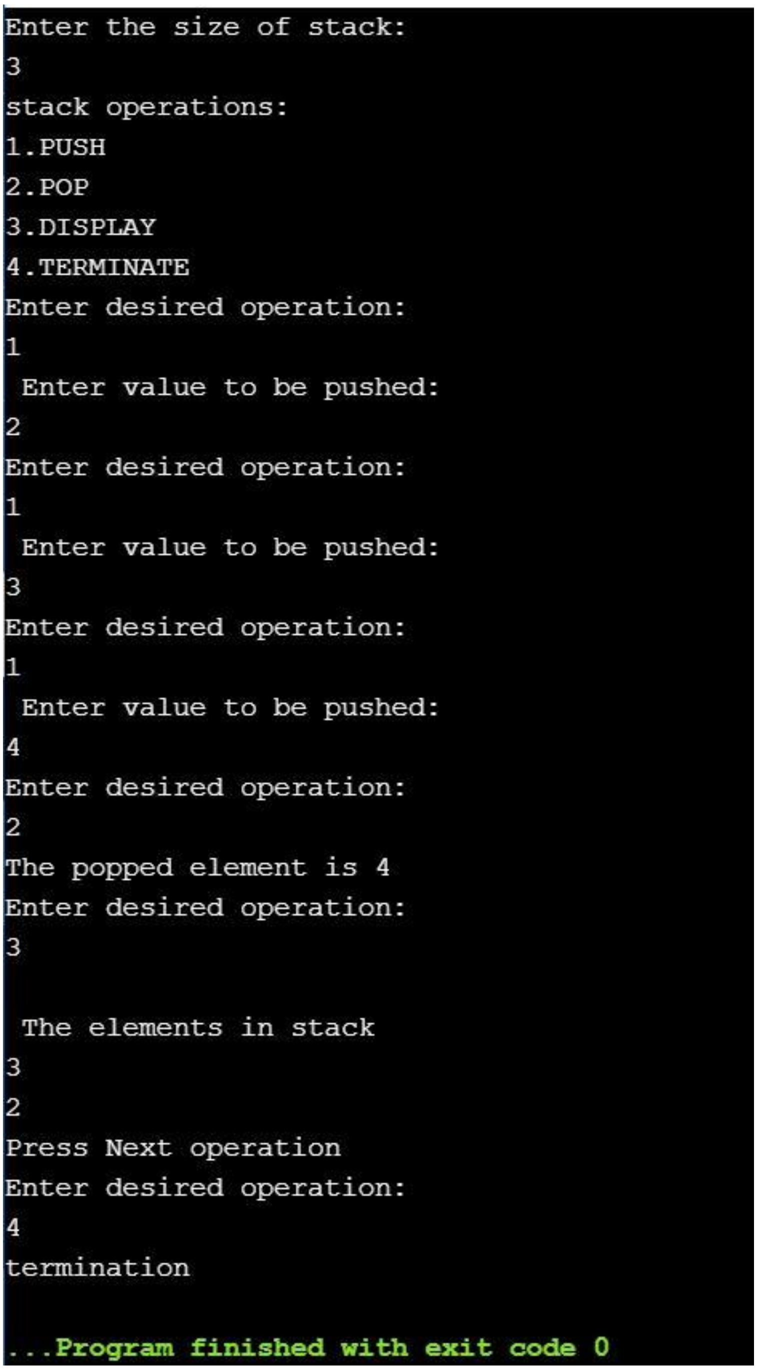
```
}
void pop()
{
   if(top<=-1)
   {
      printf( "Stack is under flow");
   }
   else
   {
      printf("The popped elements is %d",stack[top]);
      top--;
   }
}
void display()
{
   if(top>=0)
   {
      printf("\n The elements in stack \n");
      for(i=top; i>=0; i--)
         printf("%d\n",stack[i]);
      printf("Press Next operation\n");
   }
   else
   {
      printf("The stack is empty\n");
   }

}
```

OVERFLOW OUTPUT



```
Enter the size of stack:
5
stack operations:
1.PUSH
2.POP
3.DISPLAY
4.TERMINATE
Enter desired operation:
1
 Enter value to be pushed:
2
Enter desired operation:
1
 Enter value to be pushed:
3
Enter desired operation:
1
 Enter value to be pushed:
4
Enter desired operation:
1
 Enter value to be pushed:
5
Enter desired operation:
1
 Enter value to be pushed:
6
Enter desired operation:
1
stack is over flow
Enter desired operation:
4
termination
```

OUTPUT



```
Enter the size of stack:
3
stack operations:
1.PUSH
2.POP
3.DISPLAY
4.TERMINATE
Enter desired operation:
1
 Enter value to be pushed:
2
Enter desired operation:
1
 Enter value to be pushed:
3
Enter desired operation:
1
 Enter value to be pushed:
4
Enter desired operation:
2
The popped element is 4
Enter desired operation:
3

 The elements in stack
3
2
Press Next operation
Enter desired operation:
4
termination

...Program finished with exit code 0
```