LAB-03

Date : 01-01-24
Page No :

1. WAP to stimulate working of stack using an array to show i) push ii) pop iii) display

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

#define N 5
int stack [N];
int top=-1 ;

void push (int var);
void pop () ;
void display () ;

void main()
 { int choice , num;
   printf(" Enter the operation:\n 1.push
         2.pop\n  3.display\n");

   scanf ("%d", &choice);

   switch (choice).
    {
   case 1  : printf("Enter the number to be
                   pushed\n");
             scanf ("%d", &num);
              push (num);
             break;

   case 2  : pop ();
             break;
```

Date :
Page No :

```
     case 3 : display ( );
              break;

      default: printf("invalid input");
   }
}

void push (int var)
 {
    if (top = = N-1)
       {
         printf(" stack overflow");
       }
    else
       {   top + + ;
           stack [top] = var ;
       }
 }

void pop ( )
  {
     if  (top = = -1)
         {  printf("stack underflow");
         }
     else
         {
            printf("poped element = %d", stack[top]);
            top - - ;
         }
  }
```

[illegible]

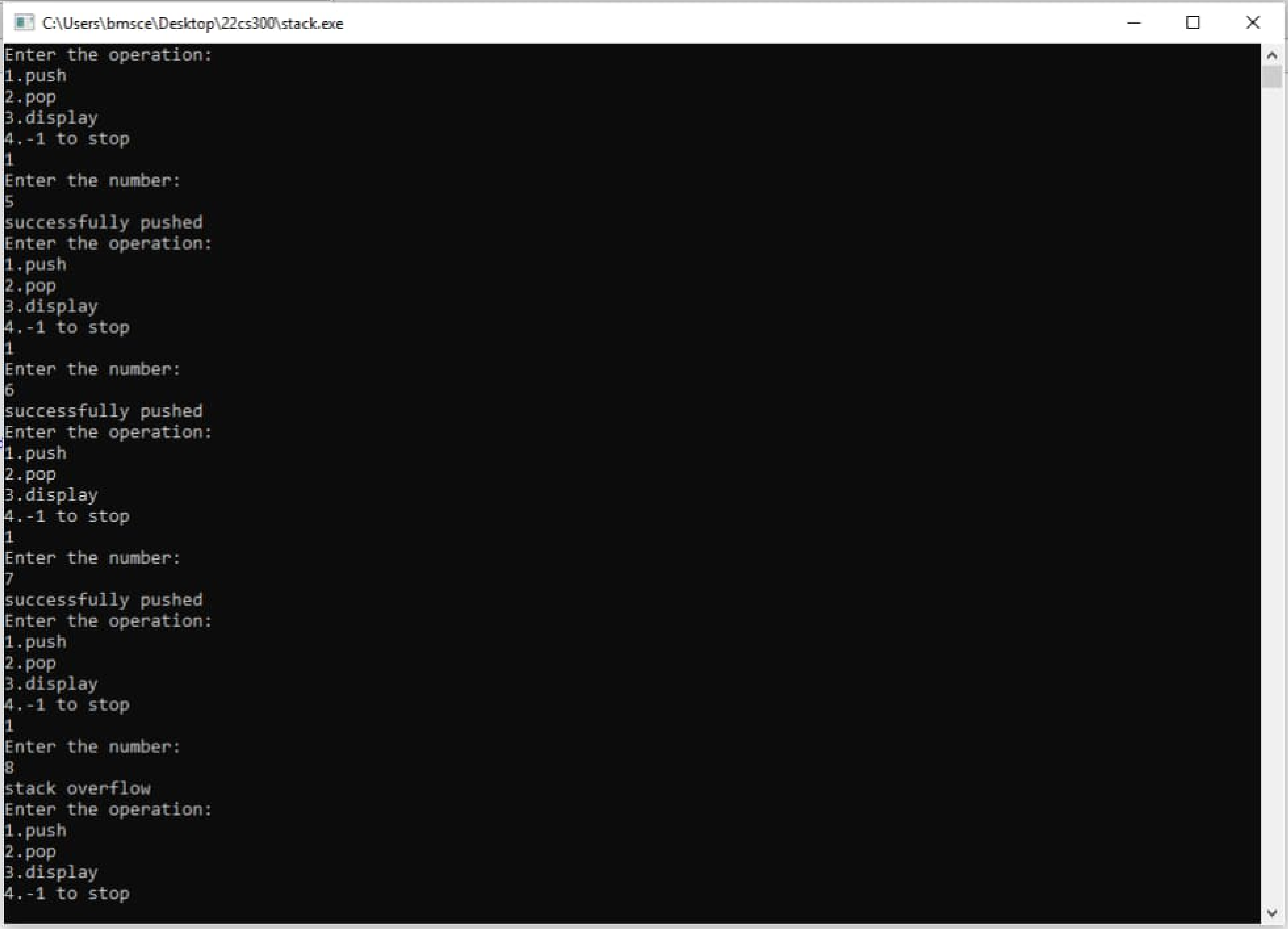
```
void display()
 {
    int i ;
    for ( ~~top=~~ i= top ; i >= 0 ; i--)
      {
        printf ("%d", stack[top]);
      }
 }
```

```
//output
Enter the operation:
1. push
 2. pop
3. display
4. -1 to stop

1
Enter the number
7
successfully pushed

Enter the operation:
1. push
2. pop
3. display
4. -1 to stop

-1
operation completed.
```



C:\Users\bmsce\Desktop\22cs300\stack.exe

```
Enter the operation:
1.push
2.pop
3.display
4.-1 to stop
1
Enter the number:
5
successfully pushed
Enter the operation:
1.push
2.pop
3.display
4.-1 to stop
1
Enter the number:
6
successfully pushed
Enter the operation:
1.push
2.pop
3.display
4.-1 to stop
1
Enter the number:
7
successfully pushed
Enter the operation:
1.push
2.pop
3.display
4.-1 to stop
1
Enter the number:
8
stack overflow
Enter the operation:
1.push
2.pop
3.display
4.-1 to stop
```

Date :
Page No :

2. WAP to convert infix expression to postfix expression using stack.

```
#include <stdio.h>
#include <string.h>
# define N 30

int index =0, ~~temp~~ pos=0, top=-1, length;
char symbol, temp, infix [N], postfix [N],
                                   stack [N];

void intopo ();
void push (char symbol);
char pop();~~(char c)~~
int prec (char c) ;

void main ()
{
     printf ("Enter the infix expression:");
     scanf ("%s", infix) ;

     intopo ();

     printf ("The infix expression is:%s", infix);
     printf ("The postfix expression is:%s", postfix
                                                  );
}
```

```
void push (char symbol)
 {
     top++;
     stack [top] = symbol;


 }

char pop ( )
 {
     return ( stack [top]) ;
      top --;
  }

int prec (char c)
  {
     if (c = = '^')
       {
           return 3;
       }
    else if (c = = '*' || c = = '/')
       {
           return 2;
       }
    else if ( c = = '-' || c = = '+')
         {
             return 1;
         }
     else
         return -1;

 }
```

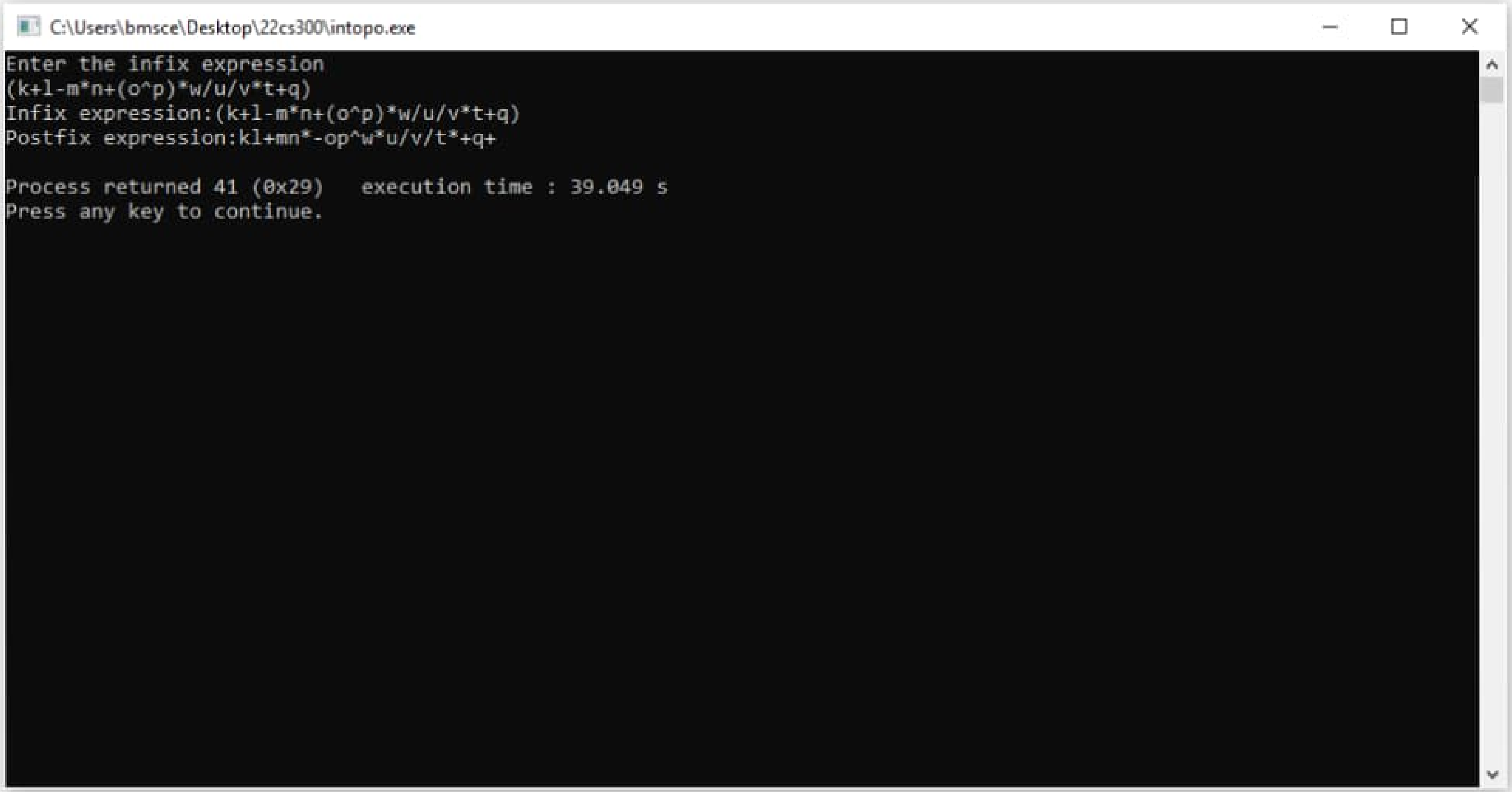
```
void intopo ()
 {
   length = strlen (infix);

   while (index < length)
     {
        symbol = infix [index];

        if (symbol = = '(')
          {
             push (symbol);
          }
        else if (symbol = = ')')
          {
             temp = pop()
             while (temp != '(')
          {  postfix [pos] = temp ;
                pos ++ ;
             temp = pop();
          }


      else
        {
           while (prec (stack[top]) > =
                                prec (symbol)
            {
               temp = pop()
               postfix [pos] = temp ;
                 pos ++ ;
            }
     index ++;
 }
while (top > 0)
 {
    ~~top~~ temp = pop()
    postfix [pos++] = temp ;
  }
}
```

Ented [illegible] 1/11/2024



```
Enter the infix expression
(k+l-m*n+(o^p)*w/u/v*t+q)
Infix expression:(k+l-m*n+(o^p)*w/u/v*t+q)
Postfix expression:kl+mn*-op^w*u/v/t*+q+

Process returned 41 (0x29)   execution time : 39.049 s
Press any key to continue.
```