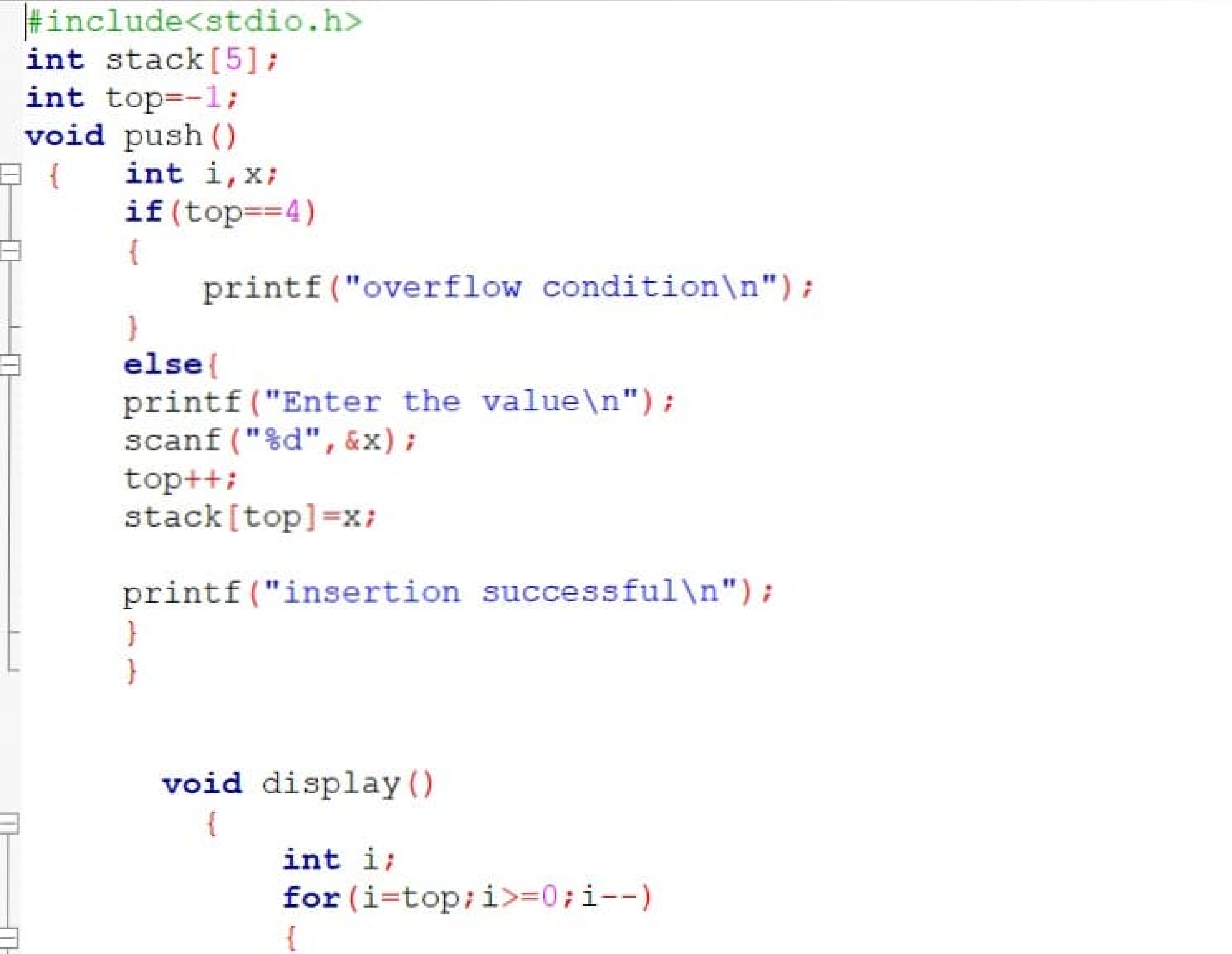
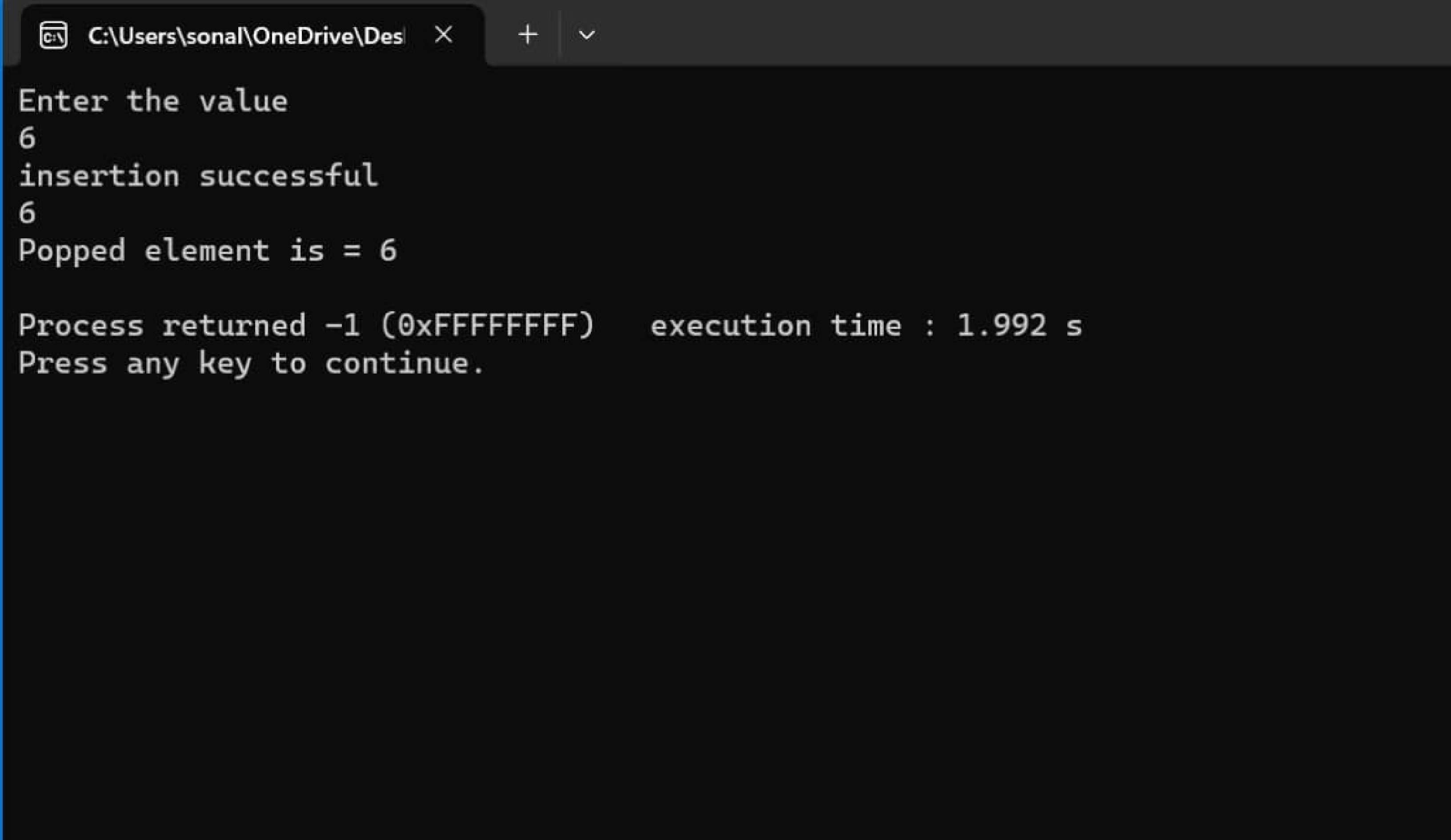


```
#include<stdio.h>
int stack[5];
int top=-1;
void push()
 {    int i,x;
      if(top==4)
      {
          printf("overflow condition\n");
      }
      else{
      printf("Enter the value\n");
      scanf("%d",&x);
      top++;
      stack[top]=x;

      printf("insertion successful\n");
      }
      }


        void display()
          {
              int i;
              for(i=top;i>=0;i--)
              {
```

```
        for(i=top;i>=0;i--)
        {
            printf("%d\n",stack[top]);
        }
    }

void pop()
{
    int i,item;
    if(top==-1)
    {
        printf("underflow condition\n");
    }
    else{

         item=stack[top];
         printf("Popped element is = %d\n",item);
         top--;

       }

    }
```



```
Enter the value
6
insertion successful
6
Popped element is = 6

Process returned -1 (0xFFFFFFFF)   execution time : 1.992 s
Press any key to continue.
```

1/1/24

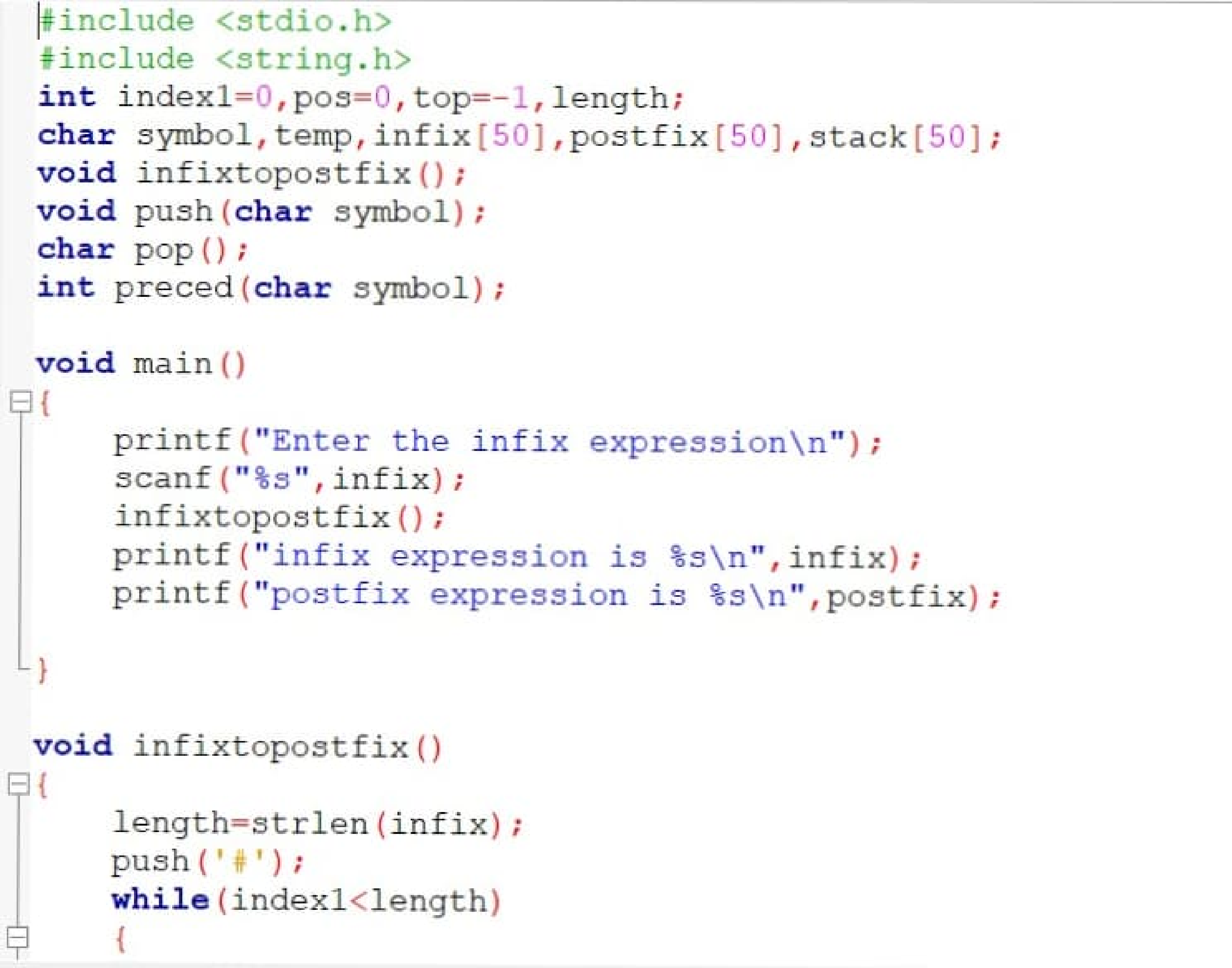
PUSH POP AND DISPLAY OPERATIONS

```
# include <stdio.h>
~~defin~~ int stack [5], top=-1;
void. main()
{
    push();
    ~~pop ();~~
    display ();
}   pop();

~~push()~~
void push()
{
   int a;
   printf ("Enter the value of a\n");
   scanf("%d", &a);
   ~~stack[top] = a;~~ top ++;
   stack [top] = a;
} ~~printf~~ printf ("insertion successful\n");

void pop()
{
  int temp;
  temp = stack [top].
  printf("Deleted element is %d",temp);
  top --;
}

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

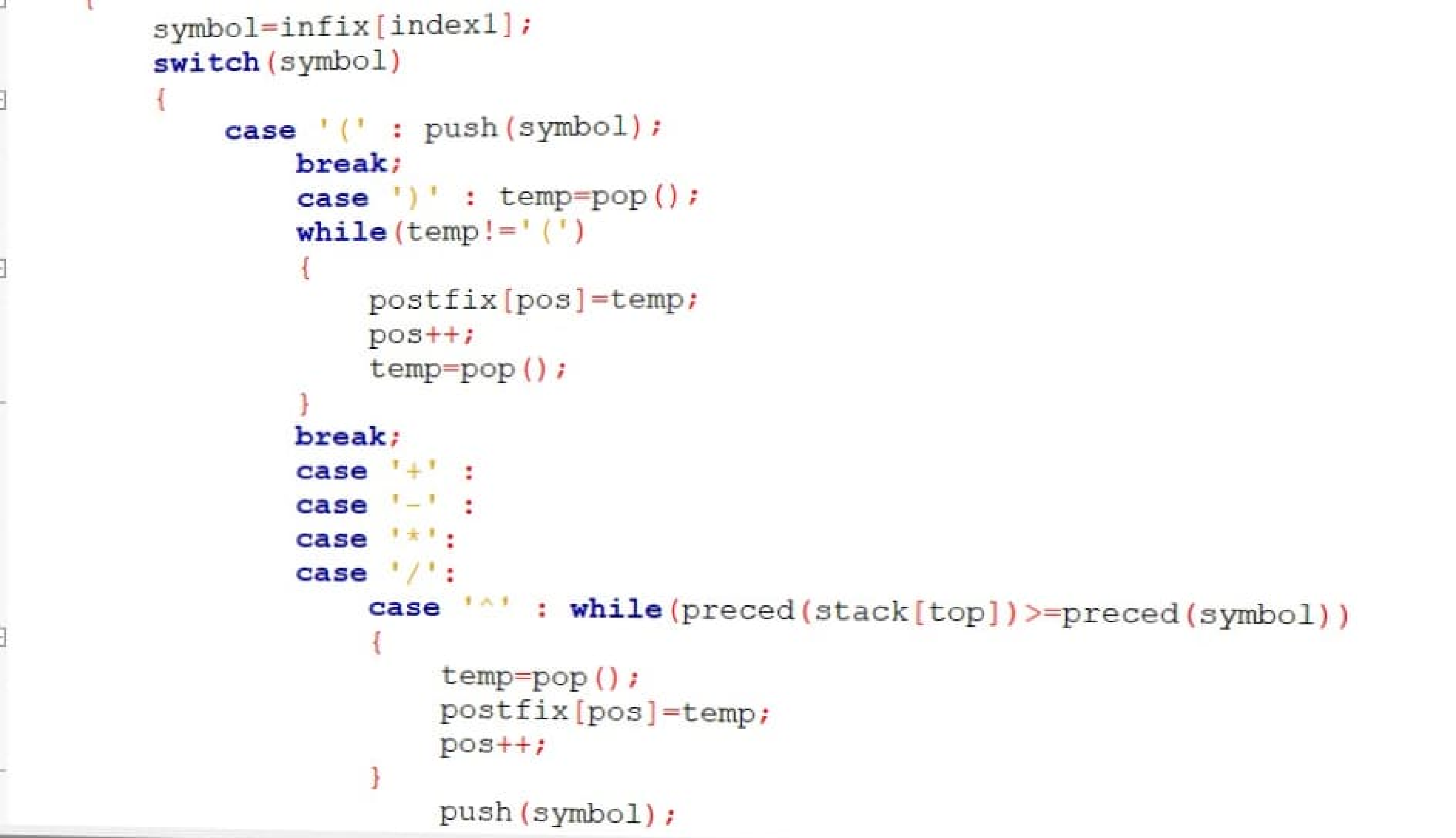
Ent
1/1/2024

Output :

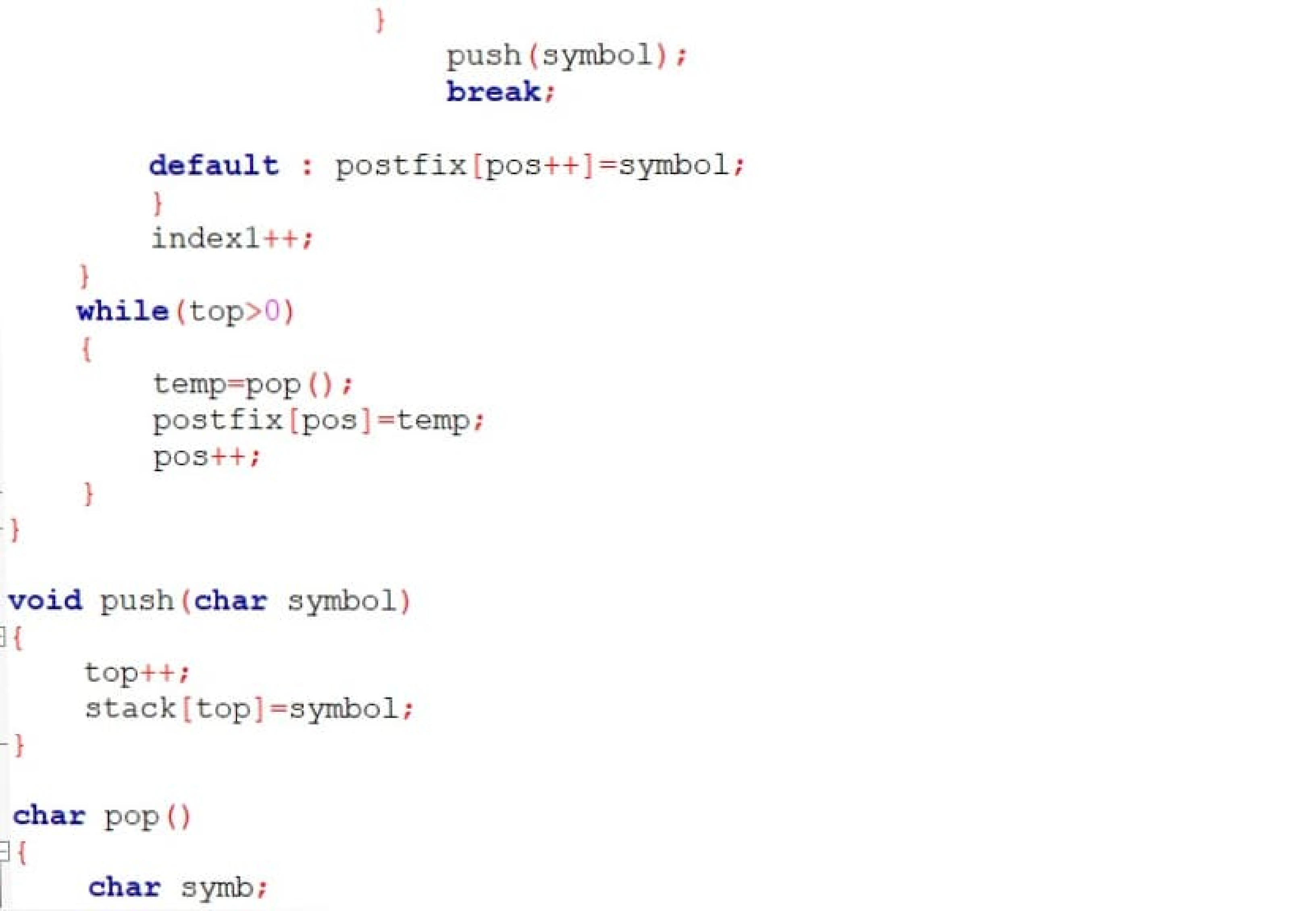
```
Enter the value of a
6
Insertion successful
~~deleted element is~~ 6
6
deleted element is 6
```



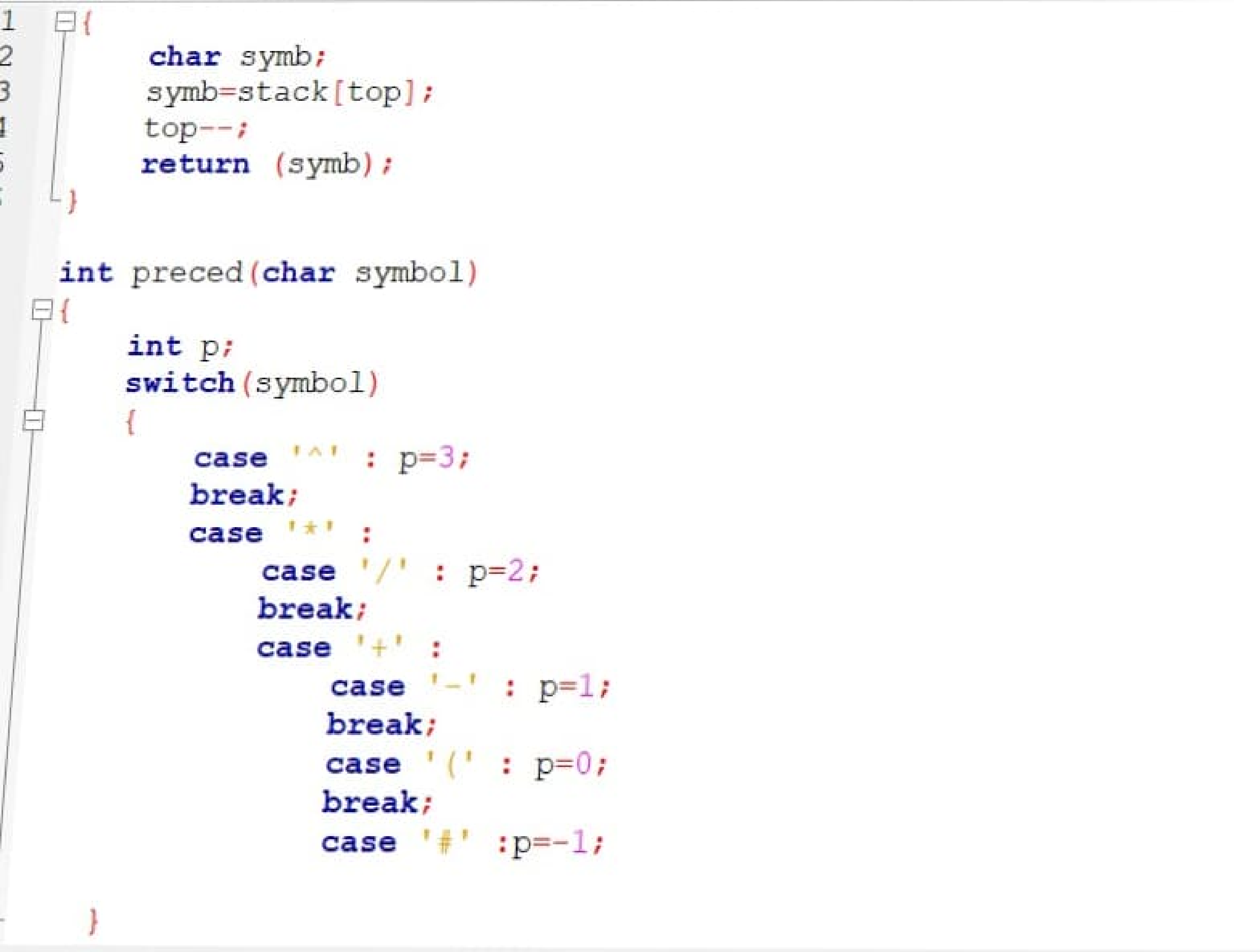
```c
#include <stdio.h>
#include <string.h>
int index1=0,pos=0,top=-1,length;
char symbol,temp,infix[50],postfix[50],stack[50];
void infixtopostfix();
void push(char symbol);
char pop();
int preced(char symbol);

void main()
{
    printf("Enter the infix expression\n");
    scanf("%s",infix);
    infixtopostfix();
    printf("infix expression is %s\n",infix);
    printf("postfix expression is %s\n",postfix);

}

void infixtopostfix()
{
    length=strlen(infix);
    push('#');
    while(index1<length)
    {
```



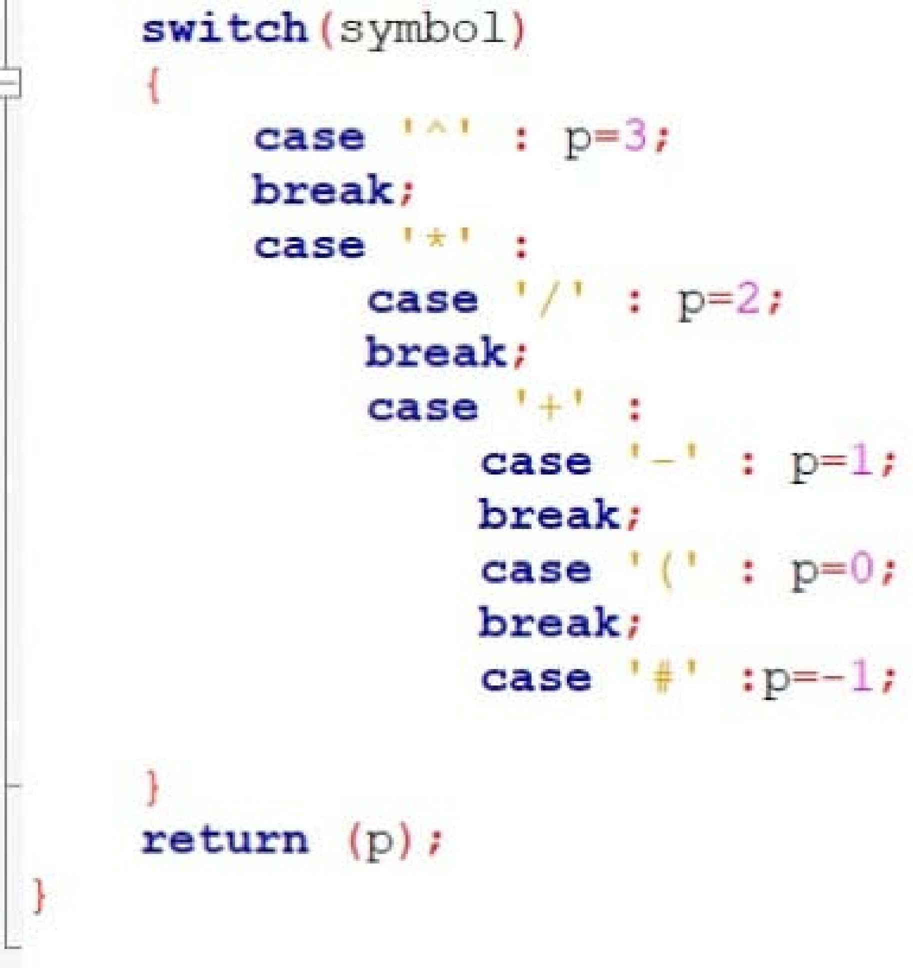
```
{
    symbol=infix[index1];
    switch(symbol)
    {
        case '(' : push(symbol);
            break;
            case ')' : temp=pop();
            while(temp!='(')
            {
                postfix[pos]=temp;
                pos++;
                temp=pop();
            }
            break;
            case '+' :
            case '-' :
            case '*':
            case '/':
                case '^' : while(preced(stack[top])>=preced(symbol))
                {
                    temp=pop();
                    postfix[pos]=temp;
                    pos++;
                }
                    push(symbol);
```



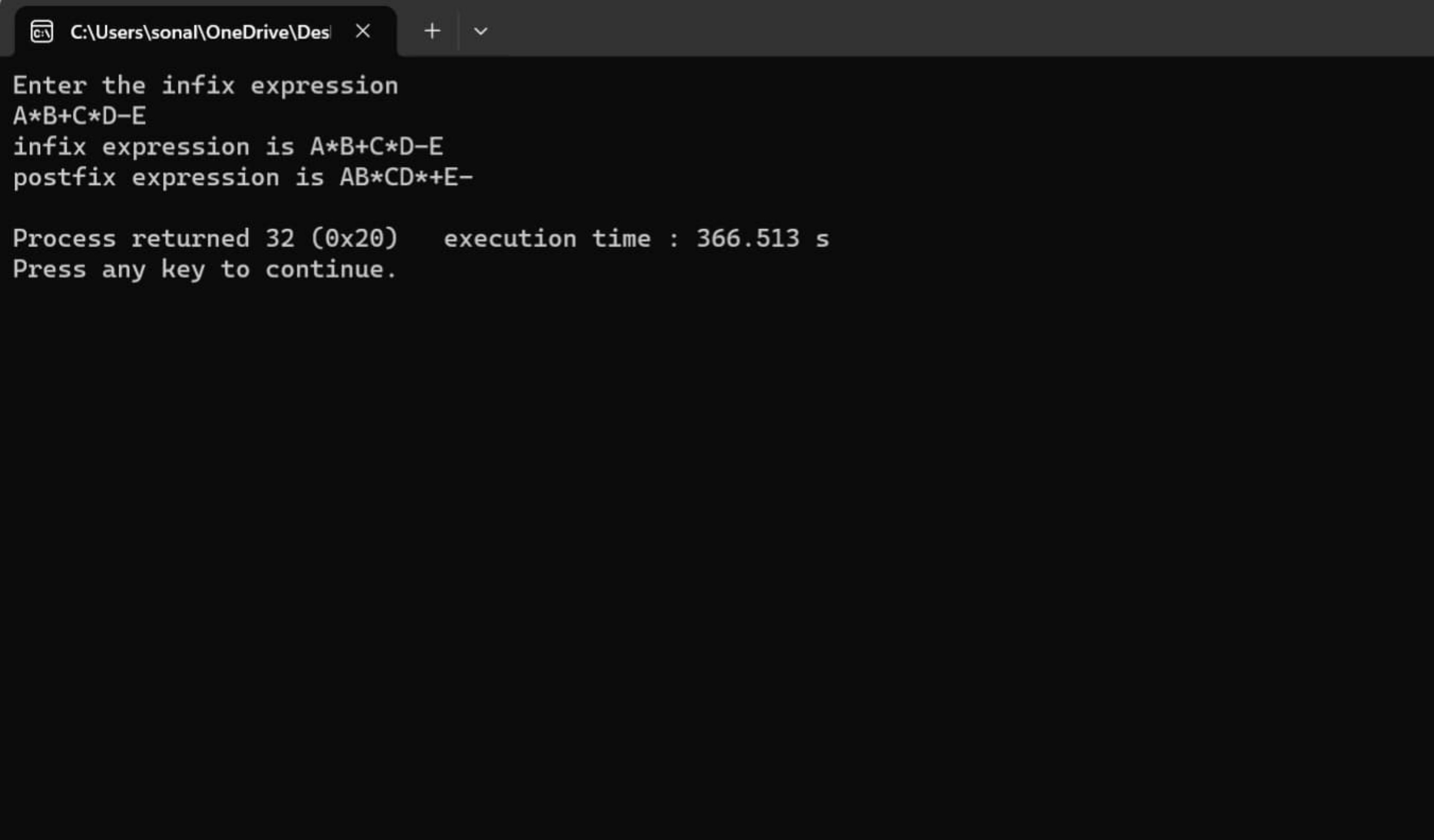
```
                    }
                        push(symbol);
                        break;

        default : postfix[pos++]=symbol;
        }
        index1++;
    }
    while(top>0)
    {
        temp=pop();
        postfix[pos]=temp;
        pos++;
    }
}

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

char pop()
{
    char symb;
```



```
{
    char symb;
    symb=stack[top];
    top--;
    return (symb);
}

int preced(char symbol)
{
    int p;
    switch(symbol)
    {
        case '^' : p=3;
        break;
        case '*' :
            case '/' : p=2;
            break;
            case '+' :
                case '-' : p=1;
                break;
                case '(' : p=0;
                break;
                case '#' :p=-1;


    }
```



```
    switch(symbol)
    {
        case '^' : p=3;
        break;
        case '*' :
            case '/' : p=2;
            break;
            case '+' :
                case '-' : p=1;
                break;
                case '(' : p=0;
                break;
                case '#' :p=-1;


    }
    return (p);
}
```



```
Enter the infix expression
A*B+C*D-E
infix expression is A*B+C*D-E
postfix expression is AB*CD*+E-

Process returned 32 (0x20)   execution time : 366.513 s
Press any key to continue.
```