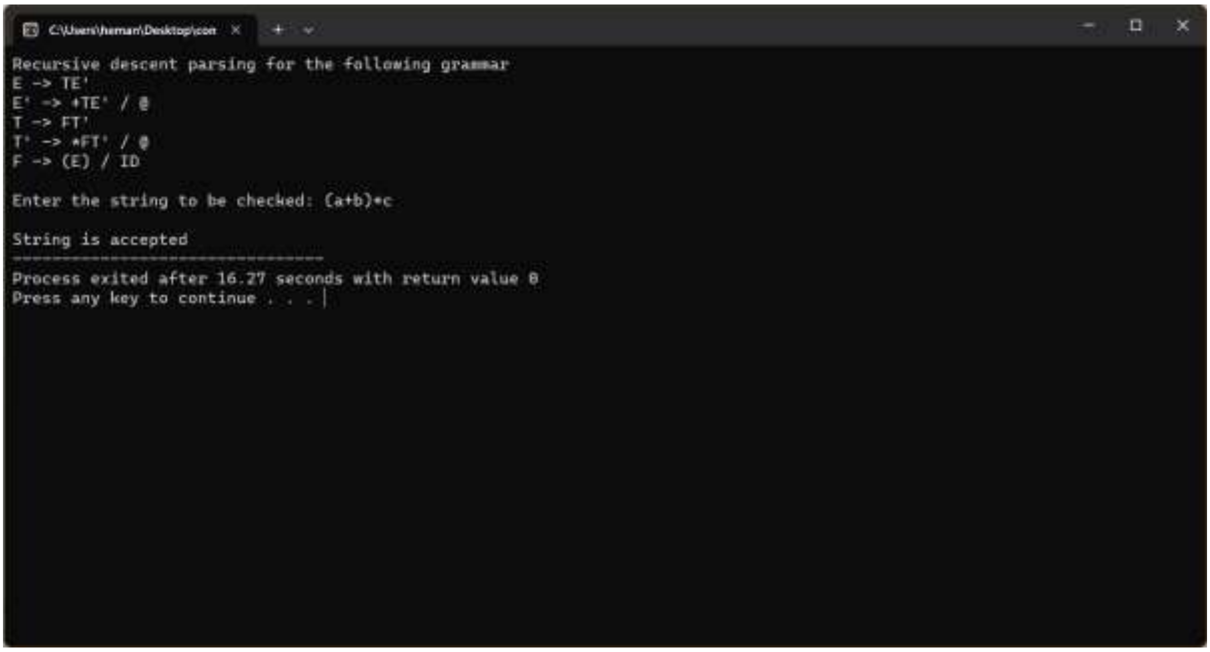
11.Write a C program to construct recursive descent parsing

```
#include<stdio.h>

#include<string.h>

char input[100];

int i;


int E();

int EP();

int T();

int TP();

int F();


int main()

{

   printf("Recursive descent parsing for the following grammar\n");

   printf("E -> TE'\nE' -> +TE' / @\nT -> FT'\nT' -> *FT' / @\nF -> (E) / ID\n");

   printf("\nEnter the string to be checked: ");

   scanf("%s", input);


   if(E())

   {

      if(input[i] == '\0')

         printf("\nString is accepted");

      else

         printf("\nString is not accepted");

   }

   else

      printf("\nString not accepted");
```

```
    return 0;
}

int E()
{
    if(T())
        return EP();
    else
        return 0;
}

int EP()
{
    if(input[i] == '+')
    {
        i++;
        if(T())
            return EP();
        else
            return 0;
    }
    else
        return 1;
}

int T()
{
    if(F())
        return TP();
    else
```

```
        return 0;
}

int TP()
{
    if(input[i] == '*')
    {
        i++;
        if(F())
            return TP();
        else
            return 0;
    }
    else
        return 1;
}

int F()
{
    if(input[i] == '(')
    {
        i++;
        if(E())
        {
            if(input[i] == ')')
            {
                i++;
                return 1;
            }
            else
                return 0;
    }
    else
      return 0;
  }
  else if(input[i] >= 'a' && input[i] <= 'z' || input[i] >= 'A' && input[i] <= 'Z')
  {
    i++;
    return 1;
  }
  else
    return 0;
}
```



```
Recursive descent parsing for the following grammar
E -> TE'
E' -> +TE' / @
T -> FT'
T' -> *FT' / @
F -> (E) / ID

Enter the string to be checked: (a+b)*c

String is accepted
--------------------------------
Process exited after 16.27 seconds with return value 0
Press any key to continue . . . 
```