

# Strings

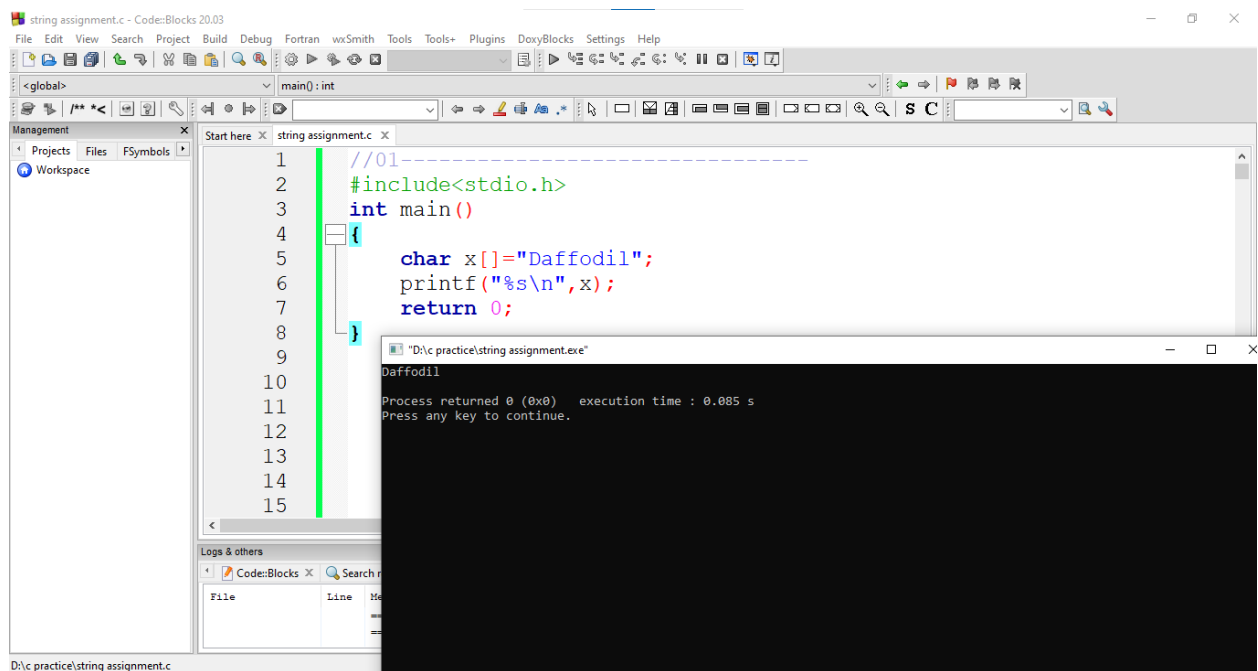
Ashraful Islam Redoy

ID:232-15-178

## 1. Write a C program to print "Daffodil" using the string

```
#include<stdio.h>

int main()
{
    char x[]="Daffodil";
    printf("%s\n",x);
    return 0;
}
```



The screenshot displays the Code::Blocks IDE interface. The main editor window shows a C program named 'string assignment.c' with the following code:

```
1 //01-----
2 #include<stdio.h>
3 int main()
4 {
5     char x[]="Daffodil";
6     printf("%s\n",x);
7     return 0;
8 }
9
10
11
12
13
14
15
```

The left sidebar shows the 'Management' pane with 'Projects', 'Files', and 'FSymbols' tabs. The 'Workspace' tab is active, showing the project structure. The bottom status bar indicates the file path: 'D:\c practice\string assignment.c'.

A terminal window titled '"D:\c practice\string assignment.exe"' is open, showing the output of the program:

```
Daffodil
Process returned 0 (0x0)   execution time : 0.085 s
Press any key to continue.
```

## 2. Write a program in C for taking a string from the user and print the string

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

```
int main()
```

```
{
```

```
    char x[50];
```

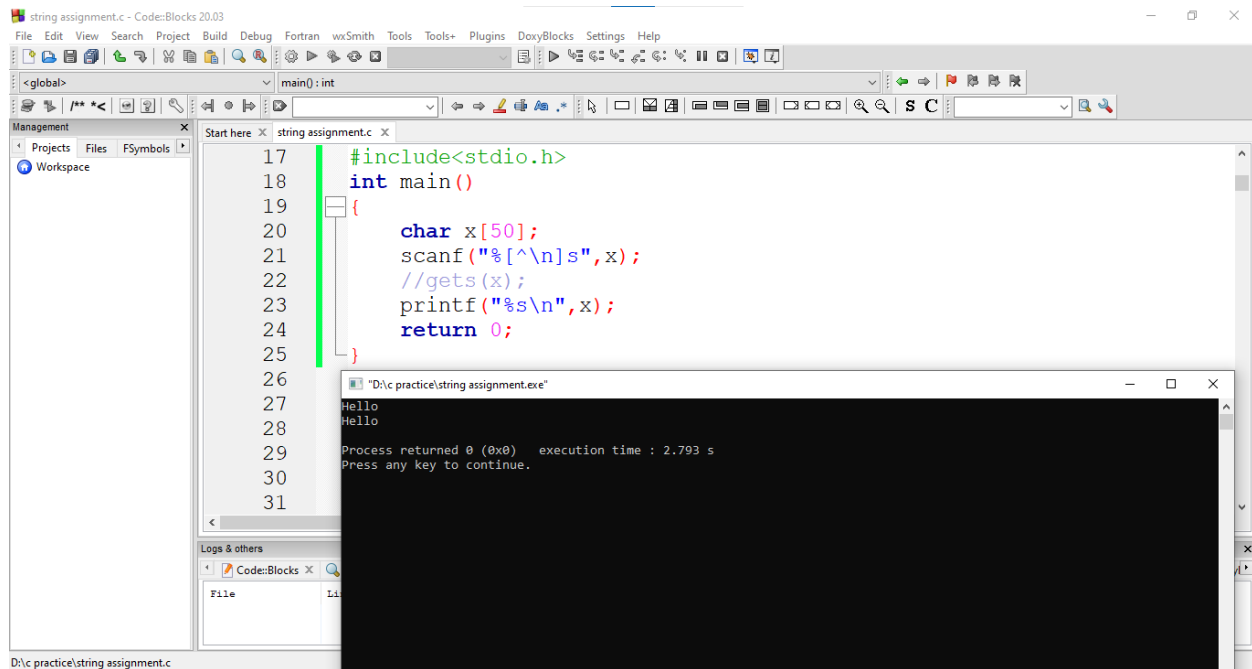
```
    scanf("%[^\\n]s",x);
```

```
    //gets(x);
```

```
    printf("%s\\n",x);
```

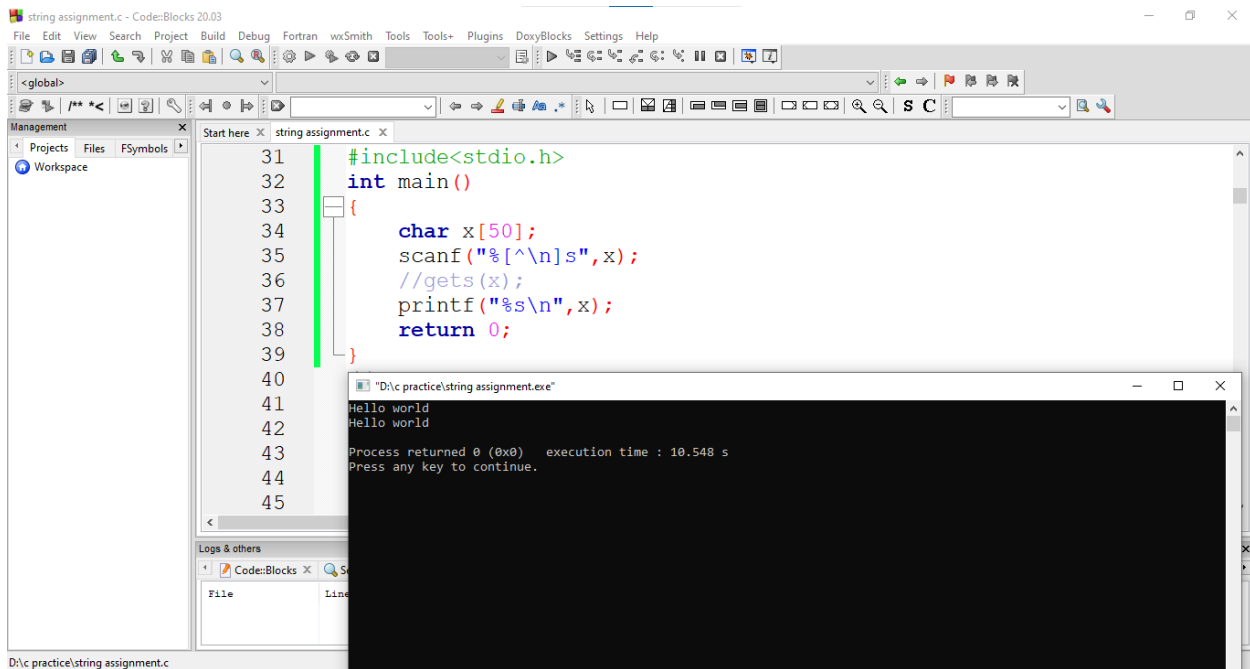
```
    return 0;
```

```
}
```



### 3. Write a program in C for taking a string from the user and print the string.

```
#include<stdio.h>
int main()
{
    char x[50];
    scanf("%[^\\n]s",x);
    //gets(x);
    printf("%s\\n",x);
    return 0;
}
```



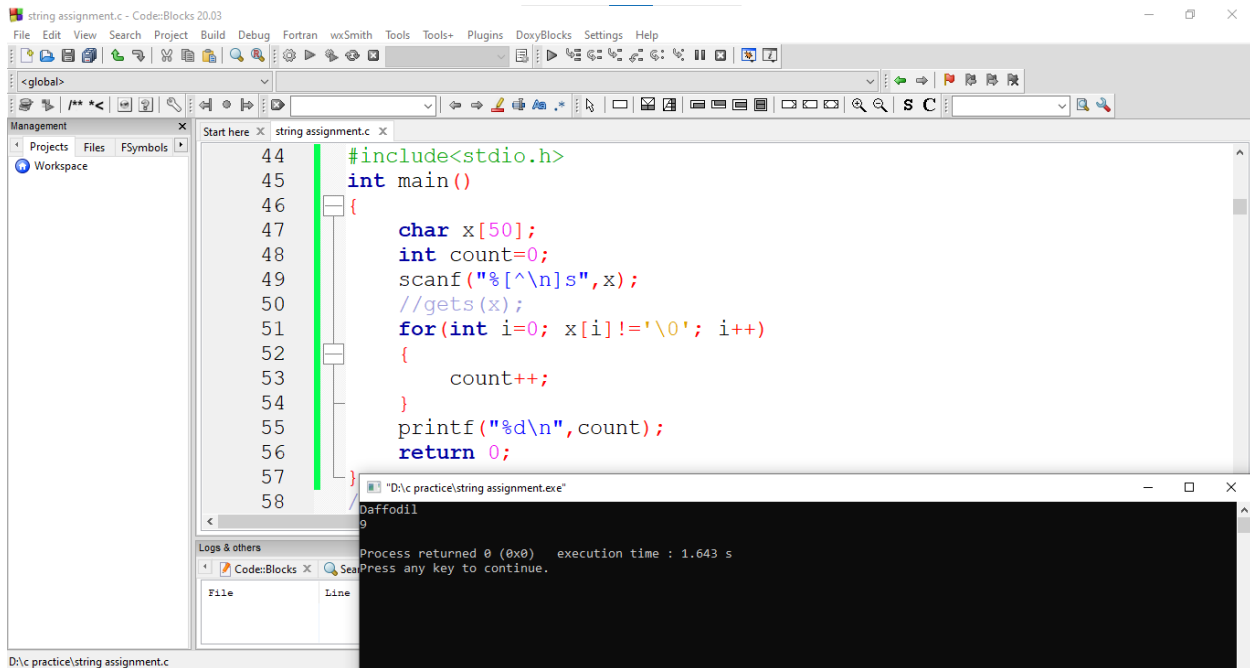
The screenshot displays the Code::Blocks IDE interface. The main editor window shows the C program code, which is identical to the one provided in the previous block. The code is line-numbered from 31 to 45. A green vertical line indicates the current position in the code. Below the editor, the 'Logs & others' panel is visible, showing the execution output of the program. The output consists of two lines of 'Hello world' text, followed by the message 'Process returned 0 (0x0) execution time : 10.548 s' and 'Press any key to continue.'.

```
string assignment.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
  Projects Files FSymbols
  Workspace
Start here string assignment.c
31 #include<stdio.h>
32 int main()
33 {
34     char x[50];
35     scanf("%[^\\n]s",x);
36     //gets(x);
37     printf("%s\\n",x);
38     return 0;
39 }
40
41
42
43
44
45
Logs & others
  Code::Blocks
  File Line
  D:\c practice\string assignment.exe
  Hello world
  Hello world
  Process returned 0 (0x0) execution time : 10.548 s
  Press any key to continue.
```

#### 4. Write a program to find the length of a string.

```
#include<stdio.h>

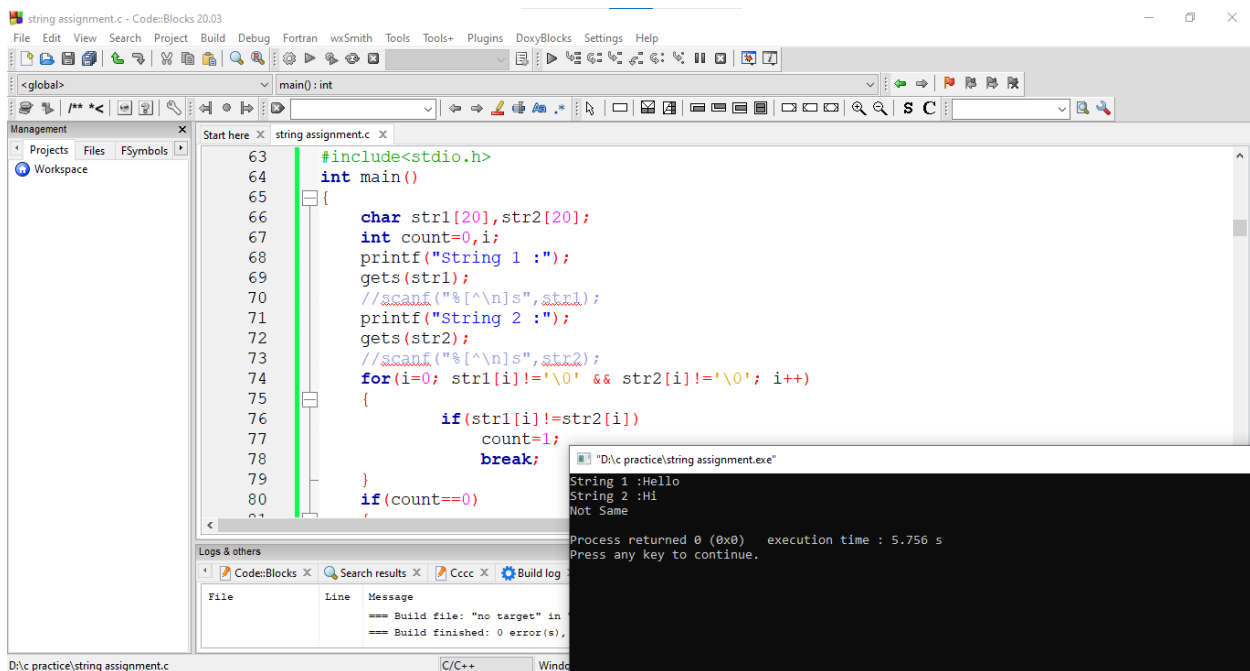
int main()
{
    char x[50];
    int count=0;
    scanf("%[^\n]s",x);
    //gets(x);
    for(int i=0; x[i]!='\0'; i++)
    {
        count++;
    }
    printf("%d\n",count);
    return 0;
}
```



## 5. Write a program to compare two strings.

```
#include<stdio.h>

int main()
{
    char str1[20],str2[20];
    int count=0,i;
    printf("String 1 :");
    gets(str1);
    //scanf("%[^\n]s",str1);
    printf("String 2 :");
    gets(str2);
    //scanf("%[^\n]s",str2);
    for(i=0; str1[i]!='\0' && str2[i]!='\0'; i++)
    {
        if(str1[i]!=str2[i])
            count=1;
            break;
    }
    if(count==0)
        printf("Same\n");
    else
        printf("Not Same\n");
    return 0;
}
```



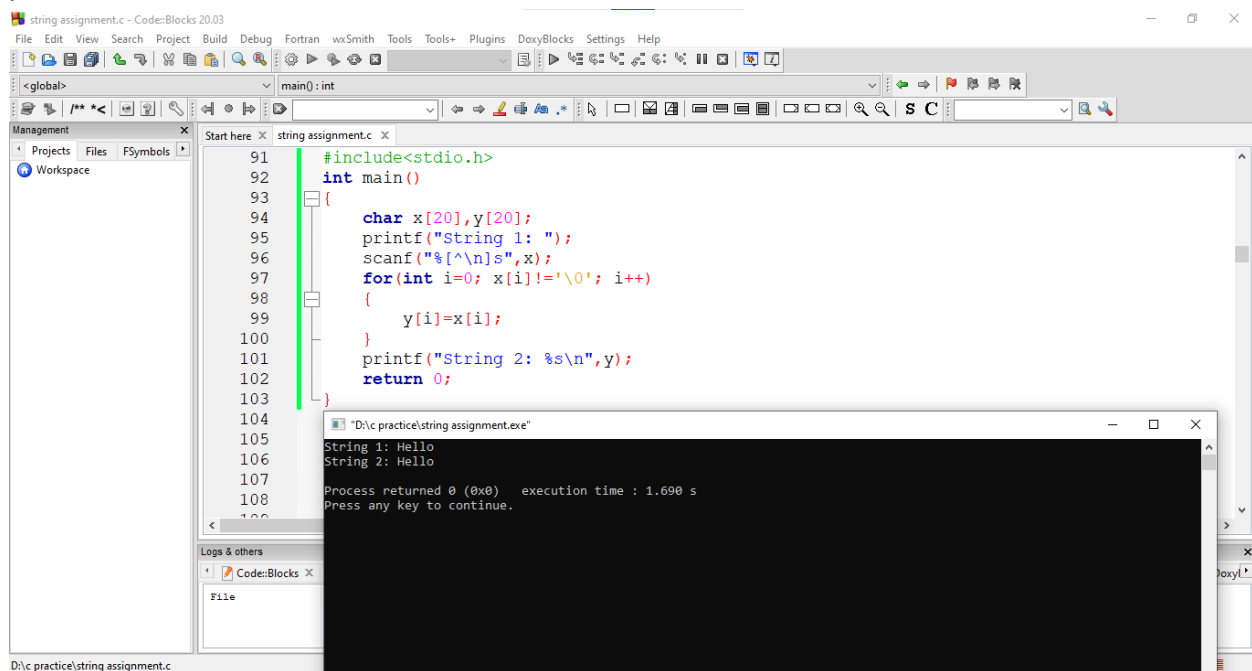
The screenshot shows the Code::Blocks IDE with the following components:

- Editor:** Displays the C program code for string comparison.
- Console:** Shows the program's output: "String 1 :Hello", "String 2 :Hi", and "Not Same". It also displays "Process returned 0 (0x0) execution time : 5.756 s" and "Press any key to continue.".
- Build Log:** Shows messages: "Build file: 'no target' in" and "Build finished: 0 error(s)".
- Project Explorer:** Shows the project structure with "string assignment.c" and "string assignment.exe".

**6. Write a program to copy a string to another string. Suppose you are taking a string from the user and storing that in S1[]. Then, copy the inputted string to the S2[].**

```
#include<stdio.h>

int main()
{
    char x[20],y[20];
    printf("String 1: ");
    scanf("%[^\n]s",x);
    for(int i=0; x[i]!='\0'; i++)
    {
        y[i]=x[i];
    }
    printf("String 2: %s\n",y);
    return 0;
}
```



## 7. Write a program to merge two strings.

```
#include<stdio.h>
int main()
{
    char x[40],y[20];
    int i,j,count=0;
    printf("String 1: ");
    scanf("%[^\n]s",x);
    fflush(stdin);
    printf("String 1: ");
    scanf("%[^\n]s",y);
    for(i=0; x[i]!='\0'; i++)
    {
        count++;
    }
    for(i=0; x[i]!='\0'; i++)
    {
        x[i+count]=y[i];
    }
    x[i+count]='\0';
    printf("String : %s\n",x);
    return 0;
}
```

The screenshot displays the Code::Blocks IDE with a C program for merging two strings. The code is as follows:

```
108 #include<stdio.h>
109 int main()
110 {
111     char x[40],y[20];
112     int i,j,count=0;
113     printf("String 1: ");
114     scanf("%[^\n]s",x);
115     fflush(stdin);
116     printf("String 1: ");
117     scanf("%[^\n]s",y);
118     for(i=0; x[i]!='\0'; i++)
119     {
120         count++;
121     }
122     for(i=0; x[i]!='\0'; i++)
123     {
124         x[i+count]=y[i];
125     }
126     x[i+count]='\0';
127     printf("String : %s\n",x);
128     return 0;
129 }
```

The terminal output shows the program's execution:

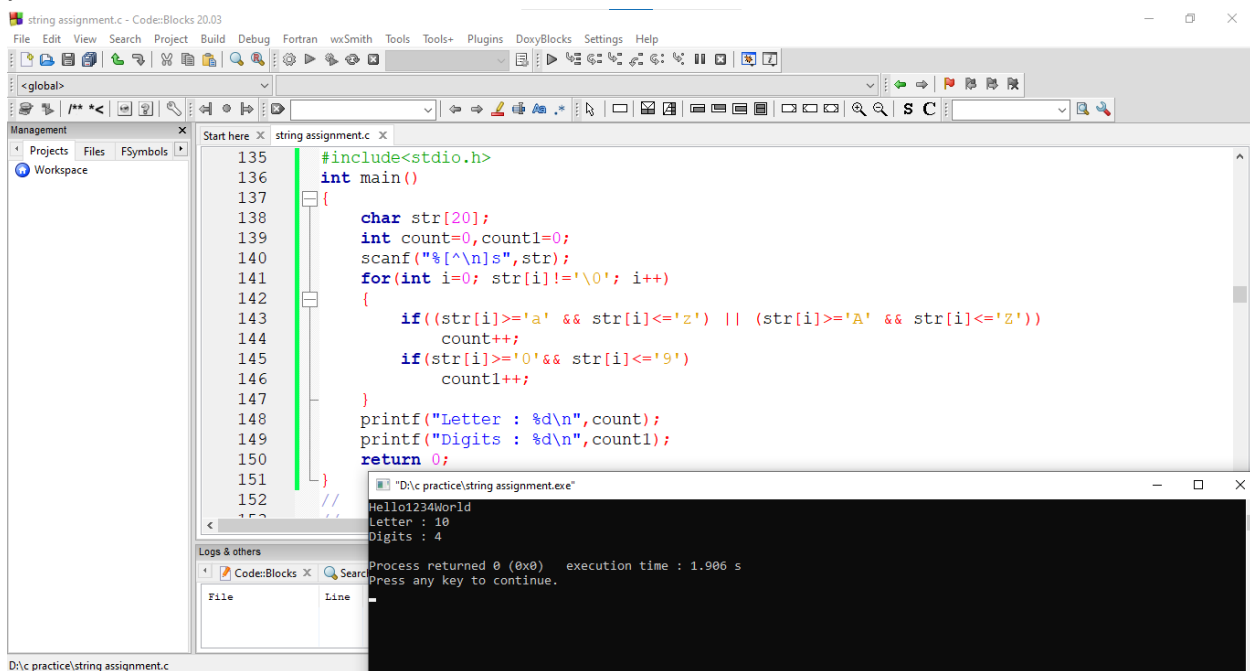
```
String 1: Hello
String 1: Hi
String : HelloHi
Process returned 0 (0x0)   execution time : 11.421 s
Press any key to continue.
```

The build log at the bottom indicates a successful build with no errors or warnings.

## 8. Write a program in C to count the number of letters and digits in a string.

```
#include<stdio.h>

int main()
{
    char str[20];
    int count=0,count1=0;
    scanf("%[^\\n]s",str);
    for(int i=0; str[i]!='\\0'; i++)
    {
        if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))
            count++;
        if(str[i]>='0' && str[i]<='9')
            count1++;
    }
    printf("Letter : %d\\n",count);
    printf("Digits : %d\\n",count1);
    return 0;
}
```



The screenshot shows a code editor window titled "string assignment.c - Code::Blocks 20.03". The code is the same as provided in the previous block. Below the code editor, there is a console window titled "D:\\c practice\\string assignment.exe" showing the output of the program. The output is:

```
Hello1234World
Letter : 10
Digits : 4
```

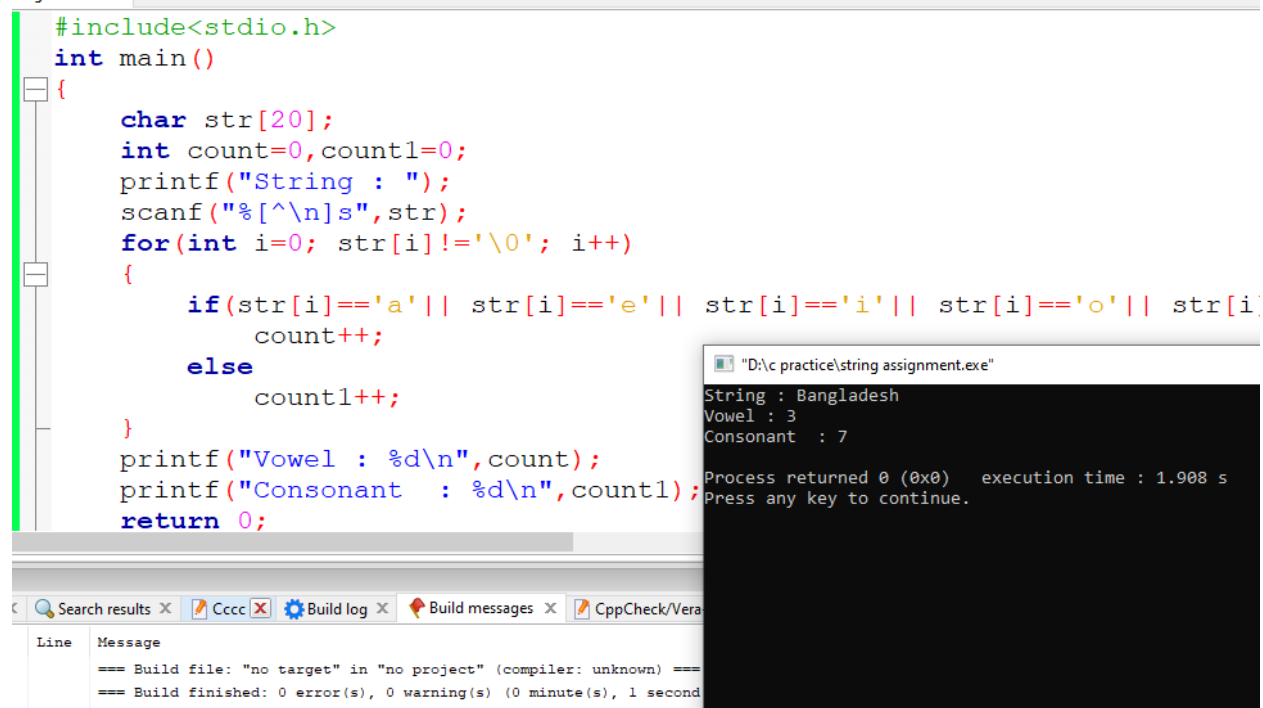
Below the console window, there is a "Logs & others" panel showing the process return code and execution time:

```
Process returned 0 (0x0)   execution time : 1.906 s
press any key to continue.
```



## 9. Write a program in C to count the number of vowels and consonants in a string.

```
#include<stdio.h>
int main()
{
    char str[20];
    int count=0,count1=0;
    printf("String : ");
    scanf("%s",str);
    for(int i=0; str[i]!='\0'; i++)
    {
        if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u' || str[i]=='A' || str[i]=='E' || str[i]=='I' ||
str[i]=='O' || str[i]=='U')
            count++;
        else
            count1++;
    }
    printf("Vowel : %d\n",count);
    printf("Consonant : %d\n",count1);
    return 0;
}
```



The image shows a screenshot of a C program and its execution. The program is a C code editor with a dark theme, showing the same code as above. The output window, titled "D:\c practice\string assignment.exe", displays the results of running the program with the input "Bangladesh". The output shows "String : Bangladesh", "Vowel : 3", and "Consonant : 7". Below this, it states "Process returned 0 (0x0) execution time : 1.908 s" and "Press any key to continue." At the bottom, there is a build log window showing the build process, including the message "Build file: 'no target' in 'no project' (compiler: unknown)" and "Build finished: 0 error(s), 0 warning(s) (0 minute(s), 1 second)".

```
#include<stdio.h>
int main()
{
    char str[20];
    int count=0,count1=0;
    printf("String : ");
    scanf("%s",str);
    for(int i=0; str[i]!='\0'; i++)
    {
        if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u' || str[i]=='A' || str[i]=='E' || str[i]=='I' ||
str[i]=='O' || str[i]=='U')
            count++;
        else
            count1++;
    }
    printf("Vowel : %d\n",count);
    printf("Consonant : %d\n",count1);
    return 0;
}
```

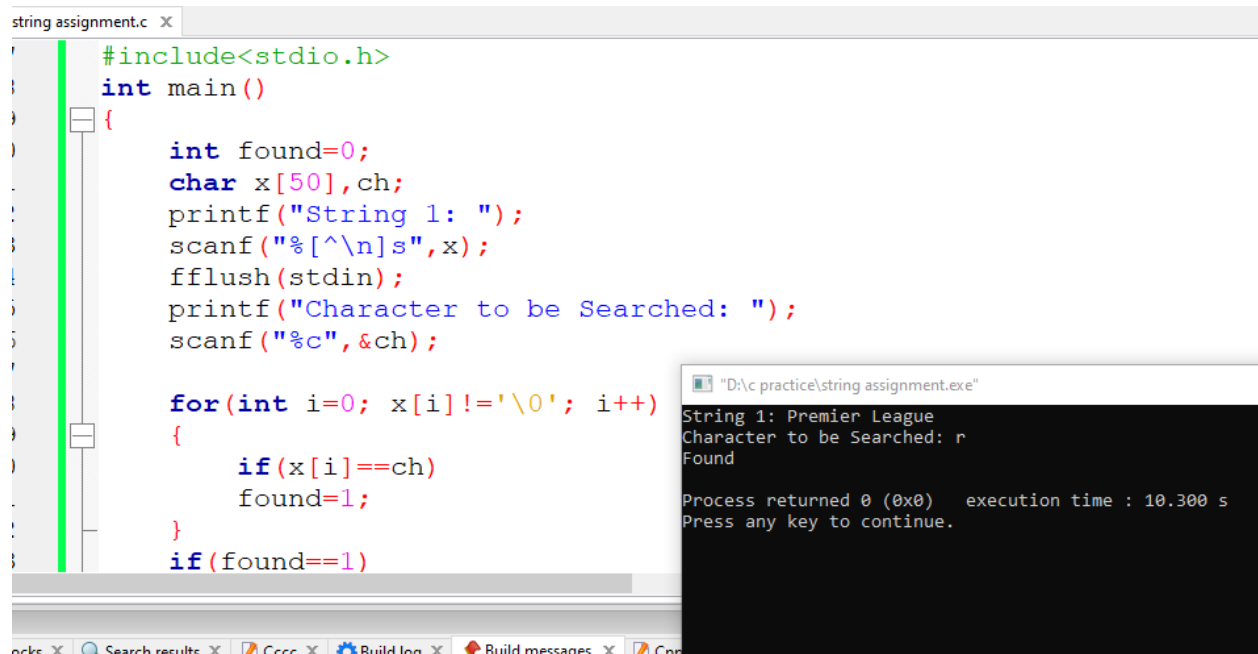
String : Bangladesh  
Vowel : 3  
Consonant : 7  
Process returned 0 (0x0) execution time : 1.908 s  
Press any key to continue.

Line Message  
=== Build file: "no target" in "no project" (compiler: unknown) ===  
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 1 second)

## 10. Write a program to search a given character in string.

```
#include<stdio.h>
int main()
{
    int found=0;
    char x[50],ch;
    printf("String 1: ");
    scanf("%[^\n]s",x);
    fflush(stdin);
    printf("Character to be Searched: ");
    scanf("%c",&ch);

    for(int i=0; x[i]!='\0'; i++)
    {
        if(x[i]==ch)
            found=1;
    }
    if(found==1)
        printf("Found\n");
    else
        printf("Not Found\n");
    return 0;
}
```



The image shows a screenshot of a C program in a text editor and its execution output in a terminal window. The program is named 'string assignment.c' and is located at 'D:\c practice\string assignment.exe'. The code is as follows:

```
#include<stdio.h>
int main()
{
    int found=0;
    char x[50],ch;
    printf("String 1: ");
    scanf("%[^\n]s",x);
    fflush(stdin);
    printf("Character to be Searched: ");
    scanf("%c",&ch);

    for(int i=0; x[i]!='\0'; i++)
    {
        if(x[i]==ch)
            found=1;
    }
    if(found==1)
```

The terminal output shows the program's execution with the following input and output:

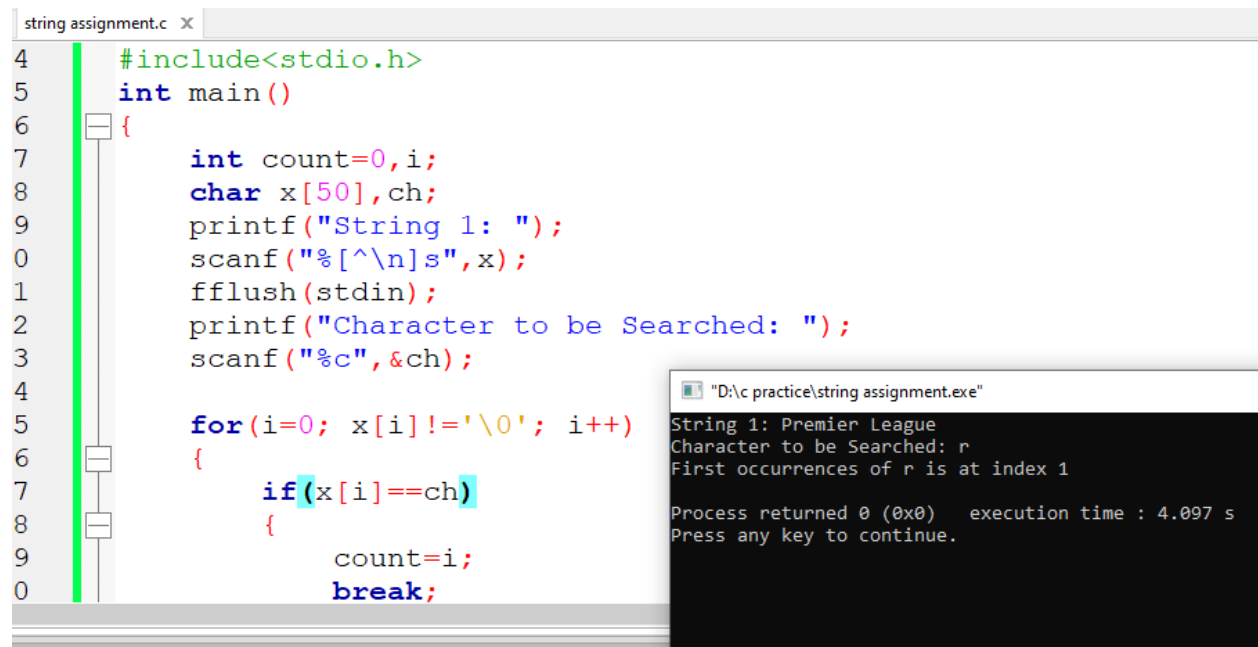
```
String 1: Premier League
Character to be Searched: r
Found

Process returned 0 (0x0)   execution time : 10.300 s
Press any key to continue.
```

## 11. Write a program to search a given character in string

```
#include<stdio.h>
int main()
{
    int count=0,i;
    char x[50],ch;
    printf("String 1: ");
    scanf("%[^\n]s",x);
    fflush(stdin);
    printf("Character to be Searched: ");
    scanf("%c",&ch);

    for(i=0; x[i]!='\0'; i++)
    {
        if(x[i]==ch)
        {
            count=i;
            break;
        }
    }
    printf("First occurrences of %c is at index %d\n",ch,count);
    return 0;
}
```



The screenshot shows a code editor window titled "string assignment.c" with the following C code:

```
4  #include<stdio.h>
5  int main()
6  {
7      int count=0,i;
8      char x[50],ch;
9      printf("String 1: ");
10     scanf("%[^\n]s",x);
11     fflush(stdin);
12     printf("Character to be Searched: ");
13     scanf("%c",&ch);
14
15     for(i=0; x[i]!='\0'; i++)
16     {
17         if(x[i]==ch)
18         {
19             count=i;
20             break;
21         }
22     }
23     printf("First occurrences of %c is at index %d\n",ch,count);
24     return 0;
25 }
```

Below the code editor is a terminal window titled "D:\c practice\string assignment.exe". It displays the program's output:

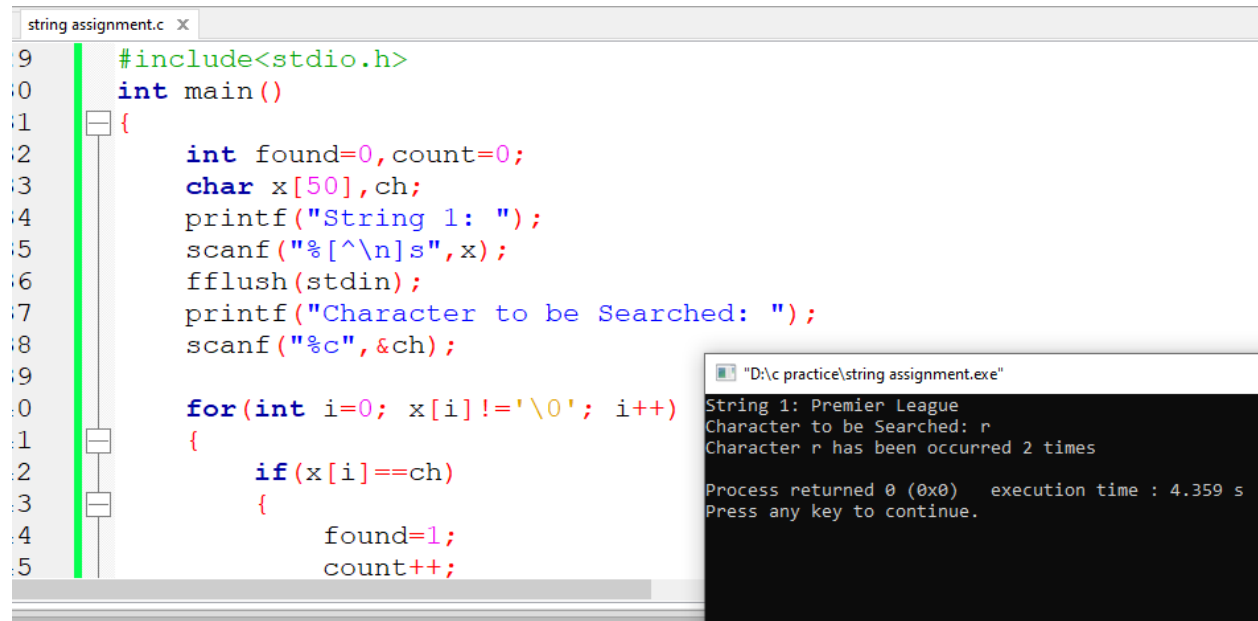
```
String 1: Premier League
Character to be Searched: r
First occurrences of r is at index 1

Process returned 0 (0x0)   execution time : 4.097 s
Press any key to continue.
```

## 12. Write a c program all occurrences of a given character in a string.

```
#include<stdio.h>
int main()
{
    int found=0,count=0;
    char x[50],ch;
    printf("String 1: ");
    scanf("%[^\n]s",x);
    fflush(stdin);
    printf("Character to be Searched: ");
    scanf("%c",&ch);

    for(int i=0; x[i]!='\0'; i++)
    {
        if(x[i]==ch)
        {
            found=1;
            count++;
        }
    }
    if(found==1)
        printf("Character r has been occurred %d times\n",count);
    else
        printf("Not Found\n");
    return 0;
}
```



The screenshot shows a C program in a code editor with line numbers 9 to 15. The code is as follows:

```
9  #include<stdio.h>
10 int main()
11 {
12     int found=0,count=0;
13     char x[50],ch;
14     printf("String 1: ");
15     scanf("%[^\n]s",x);
16     fflush(stdin);
17     printf("Character to be Searched: ");
18     scanf("%c",&ch);
19
20     for(int i=0; x[i]!='\0'; i++)
21     {
22         if(x[i]==ch)
23         {
24             found=1;
25             count++;
26         }
27     }
28     if(found==1)
29         printf("Character r has been occurred %d times\n",count);
30     else
31         printf("Not Found\n");
32     return 0;
33 }
```

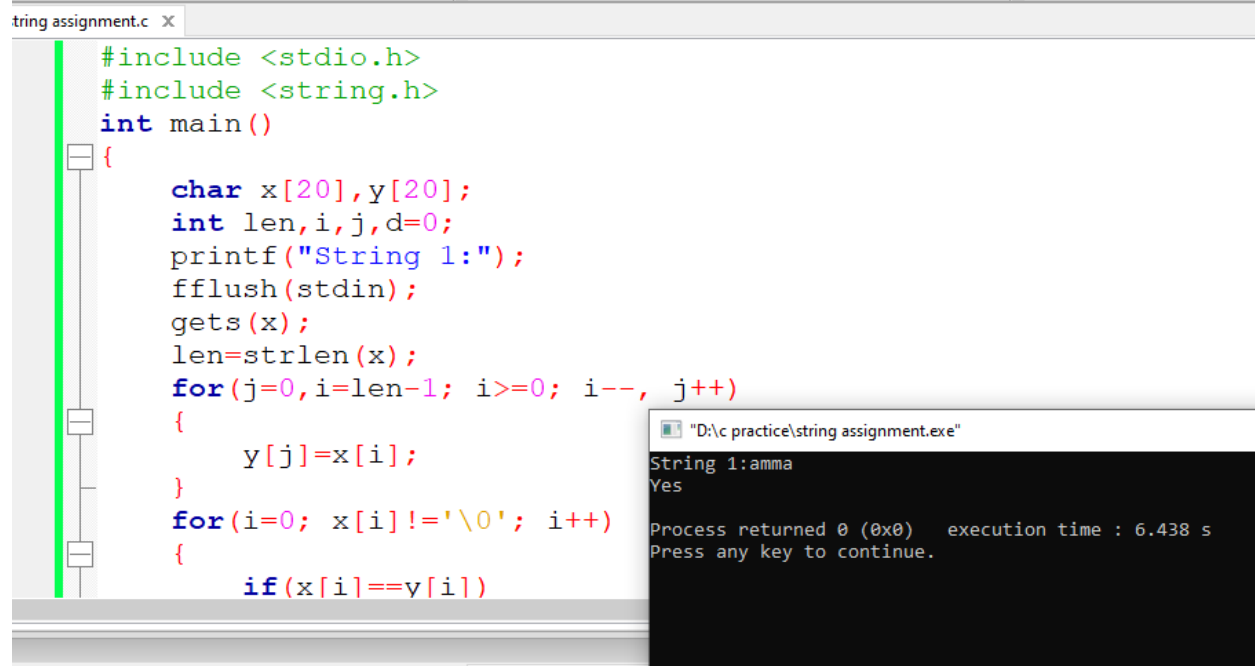
Below the code editor, a console window titled "D:\c practice\string assignment.exe" displays the program's output:

```
String 1: Premier League
Character to be Searched: r
Character r has been occurred 2 times

Process returned 0 (0x0)   execution time : 4.359 s
Press any key to continue.
```

### 13. Write a c program to find whether a given string is palindrome or not.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char x[50],y[50];
    int len,i,j,d=0;
    fflush(stdin);
    gets(x);
    len=strlen(x);
    for(j=0,i=len-1; i>=0; i--, j++)
    {
        y[j]=x[i];
    }
    for(i=0; x[i]!='\0'; i++)
    {
        if(x[i]==y[i])
            d=1;
    }
    if(d==0)
        printf("No\n");
    else
        printf("Yes\n");
    return 0;
}
```



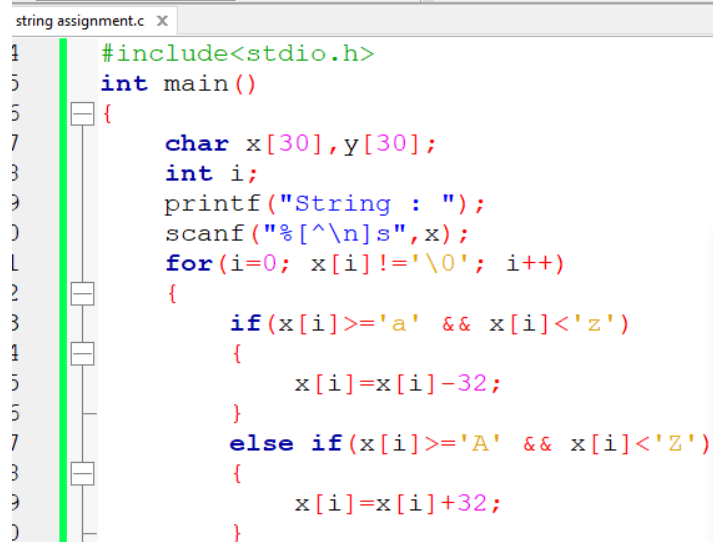
```
#include <stdio.h>
#include <string.h>
int main()
{
    char x[20],y[20];
    int len,i,j,d=0;
    printf("String 1:");
    fflush(stdin);
    gets(x);
    len=strlen(x);
    for(j=0,i=len-1; i>=0; i--, j++)
    {
        y[j]=x[i];
    }
    for(i=0; x[i]!='\0'; i++)
    {
        if(x[i]==y[i])

```

```
"D:\c practice\string assignment.exe"
String 1: amma
Yes
Process returned 0 (0x0)   execution time : 6.438 s
Press any key to continue.
```

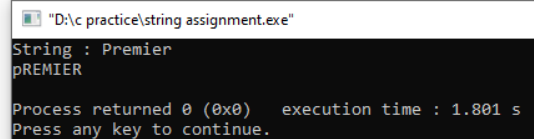
#### 14. Write a c program that will convert a lowercase string to an uppercase string.

```
#include<stdio.h>
int main()
{
    char x[30],y[30];
    int i;
    printf("String : ");
    scanf("%[^\\n]s",x);
    for(i=0; x[i]!='\\0'; i++)
    {
        if(x[i]>='a' && x[i]<'z')
        {
            x[i]=x[i]-32;
        }
        else if(x[i]>='A' && x[i]<'Z')
        {
            x[i]=x[i]+32;
        }
    }
    printf("%s\\n",x);
    return 0;
}
```



The screenshot shows a code editor window titled "string assignment.c". The code is as follows:

```
1  #include<stdio.h>
5  int main()
5  {
7      char x[30],y[30];
3      int i;
9      printf("String : ");
0      scanf("%[^\\n]s",x);
L     for(i=0; x[i]!='\\0'; i++)
2     {
3         if(x[i]>='a' && x[i]<'z')
4         {
5             x[i]=x[i]-32;
5         }
7         else if(x[i]>='A' && x[i]<'Z')
3         {
9             x[i]=x[i]+32;
0         }
    }
```



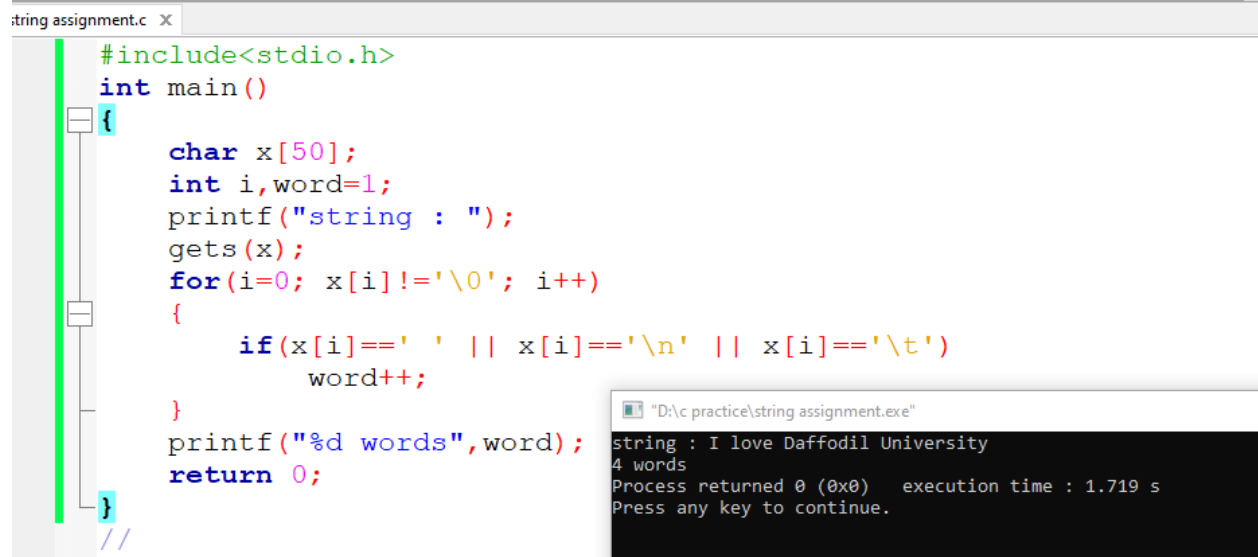
The screenshot shows a command prompt window titled "D:\c practice\string assignment.exe". The output is as follows:

```
String : Premier
pREMIER

Process returned 0 (0x0)   execution time : 1.801 s
Press any key to continue.
```

### 15. Write a c program to find the number of words in a given string

```
#include<stdio.h>
int main()
{
    char x[50];
    int i,word=1;
    printf("string : ");
    gets(x);
    for(i=0; x[i]!='\0'; i++)
    {
        if(x[i]==' ' || x[i]=='\n' || x[i]=='\t')
            word++;
    }
    printf("%d words",word);
    return 0;
}
```



The image shows a screenshot of a C program in a text editor and its execution output in a command prompt window.

The C program, named `string assignment.c`, is as follows:

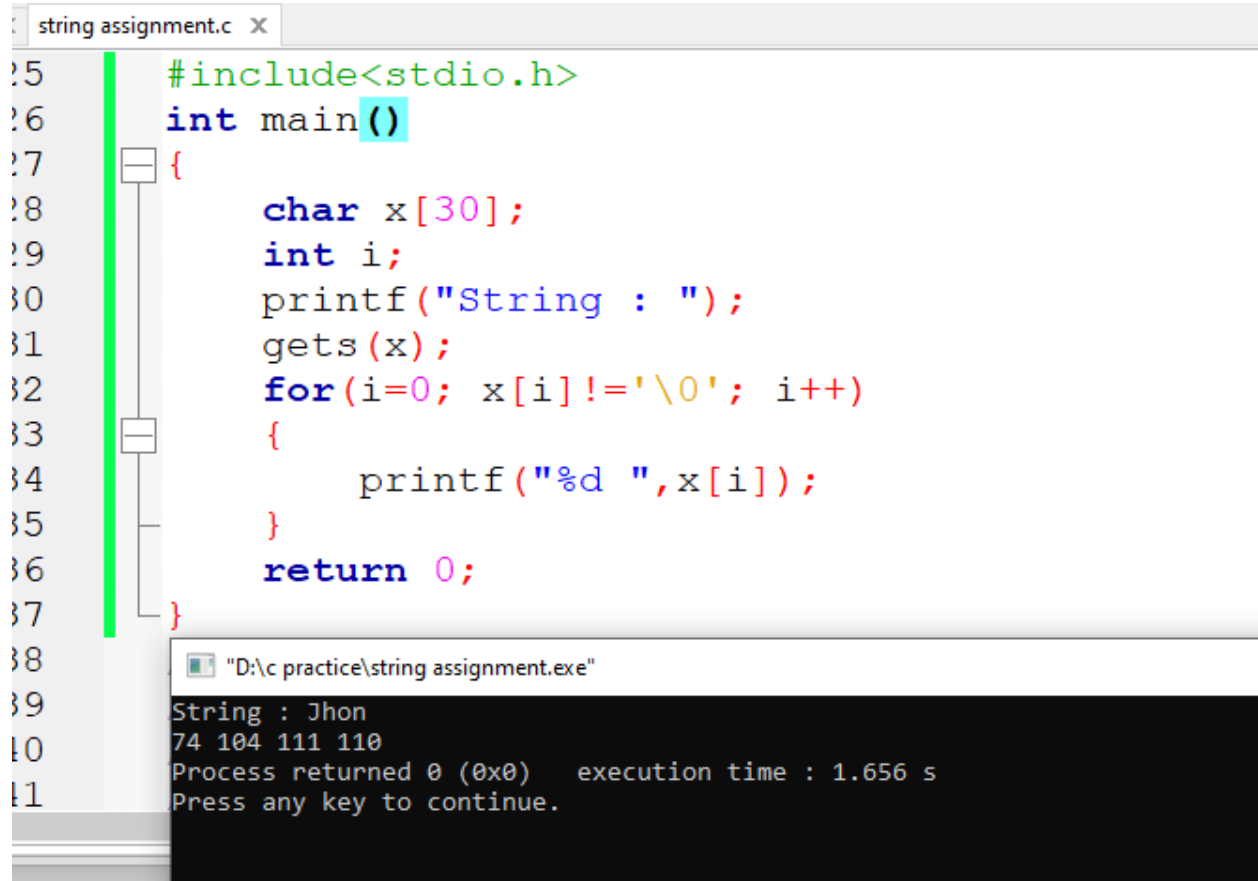
```
#include<stdio.h>
int main()
{
    char x[50];
    int i,word=1;
    printf("string : ");
    gets(x);
    for(i=0; x[i]!='\0'; i++)
    {
        if(x[i]==' ' || x[i]=='\n' || x[i]=='\t')
            word++;
    }
    printf("%d words",word);
    return 0;
}
```

The program is compiled and executed, resulting in a command prompt window titled `"D:\c practice\string assignment.exe"`. The output of the program is:

```
string : I love Daffodil University
4 words
Process returned 0 (0x0)   execution time : 1.719 s
Press any key to continue.
```

**16. Write a program, which reads your name from the keyboard and outputs a list of ASCII codes, which represent your name.**

```
#include<stdio.h>
int main()
{
    char x[30];
    int i;
    printf("String : ");
    gets(x);
    for(i=0; x[i]!='\0'; i++)
    {
        printf("%d ",x[i]);
    }
    return 0;
}
```



The image shows a screenshot of a C program and its execution. The program is named 'string assignment.c' and is located in the directory 'D:\c practice\'. The code is as follows:

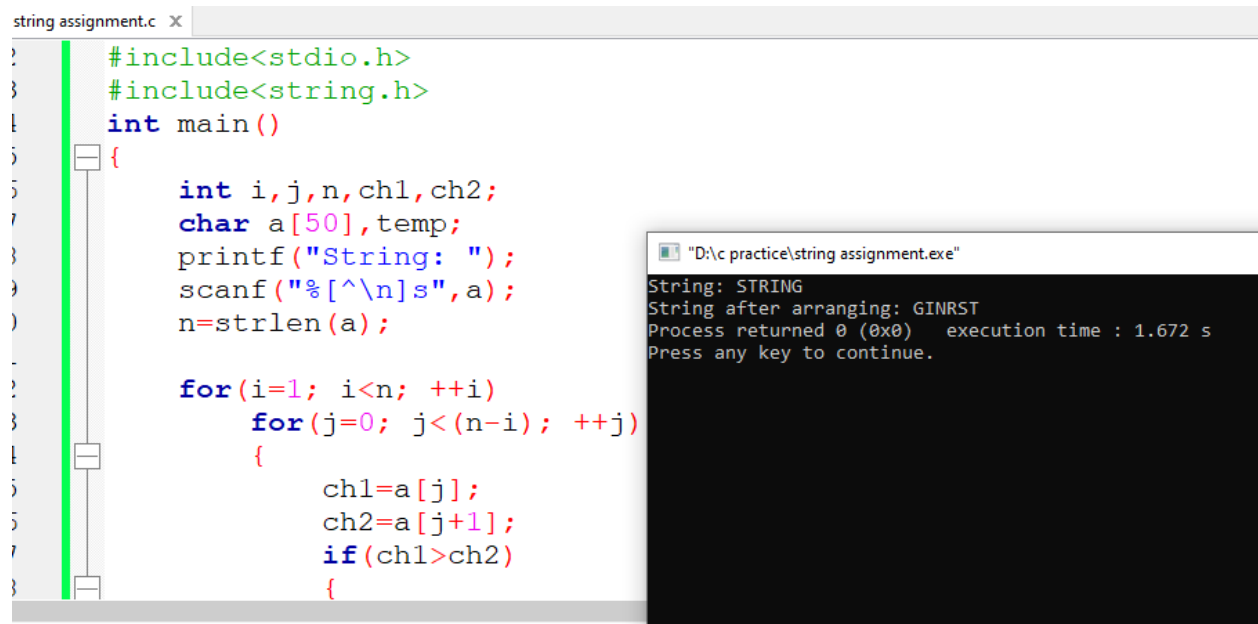
```
15 #include<stdio.h>
16 int main()
17 {
18     char x[30];
19     int i;
20     printf("String : ");
21     gets(x);
22     for(i=0; x[i]!='\0'; i++)
23     {
24         printf("%d ",x[i]);
25     }
26     return 0;
27 }
```

The execution output is shown in a window titled 'D:\c practice\string assignment.exe'. It displays the input 'Jhon' and the corresponding ASCII codes '74 104 111 110'. The process returned 0 (0x0) and the execution time was 1.656 s. The prompt 'Press any key to continue.' is also visible.



## 17. Write a program which will read a string and rewrite it in the alphabetical order.

```
#include<stdio.h>
#include<string.h>
int main()
{
    int i,j,n,ch1,ch2;
    char a[50],temp;
    printf("String: ");
    scanf("%[^\\n]s",a);
    n=strlen(a);
    for(i=1; i<n; ++i)
        for(j=0; j<(n-i); ++j)
        {
            ch1=a[j];
            ch2=a[j+1];
            if(ch1>ch2)
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    printf("String after arranging: %s",a);
    return 0;
}
```



```
string assignment.c x
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     int i,j,n,ch1,ch2;
6     char a[50],temp;
7     printf("String: ");
8     scanf("%[^\\n]s",a);
9     n=strlen(a);
10
11     for(i=1; i<n; ++i)
12         for(j=0; j<(n-i); ++j)
13         {
14             ch1=a[j];
15             ch2=a[j+1];
16             if(ch1>ch2)
17             {
18                 temp=a[j];
19                 a[j]=a[j+1];
20                 a[j+1]=temp;
21             }
22         }
23     printf("String after arranging: %s",a);
24     return 0;
25 }
```

"D:\c practice\string assignment.exe"

String: STRING  
String after arranging: GINRST  
Process returned 0 (0x0) execution time : 1.672 s  
Press any key to continue.