

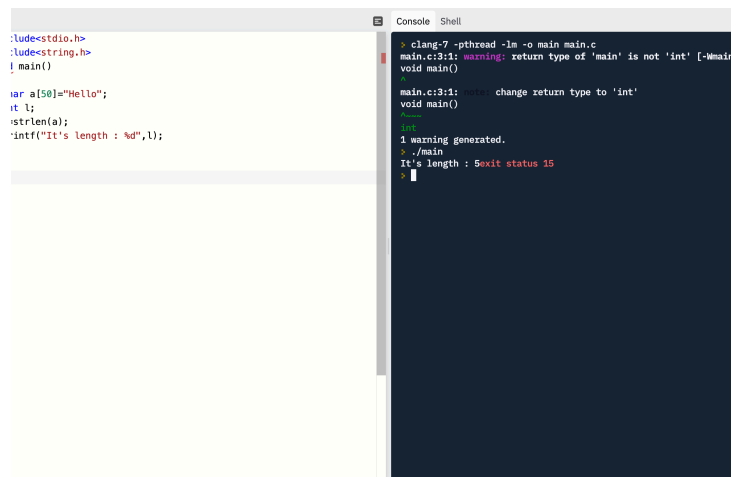
STRING HANDLING FUNCTIONS

- **C** language supports a large number of string handling functions that can be used to carry out many of the string manipulations.
- These functions are packaged in string.h library.
- Hence we must include string.h header file in our program to use these functions.
- The following are the most commonly used string handling function.

- 1) strlen()
- 2) strcpy()
- 3) strcat()
- 4) strcmp()
- 5) strrev()
- 6) strlwr()
- 7)strupr()

1. strlen()

The function calculates the length of a given string. The `strlen()` function takes a string as an argument and returns its length. The returned value is of type `size_t` (the unsigned integer type). It is defined in the `<string.h>` header file.



```
.:lude<stdio.h>
.:lude<string.h>
| main()

| var a[50]="Hello";
| int l;
| strlen(a);
| printf("It's length : %d",l);
```

```
> clang-7 -pthread -lm -o main main.c
main.c:3:1: warning: return type of 'main' is not 'int' [-Wmain]
void main()
^
main.c:3:1: note: change return type to 'int'
int
1 warning generated.
./main
It's length : 5; exit status 15
```

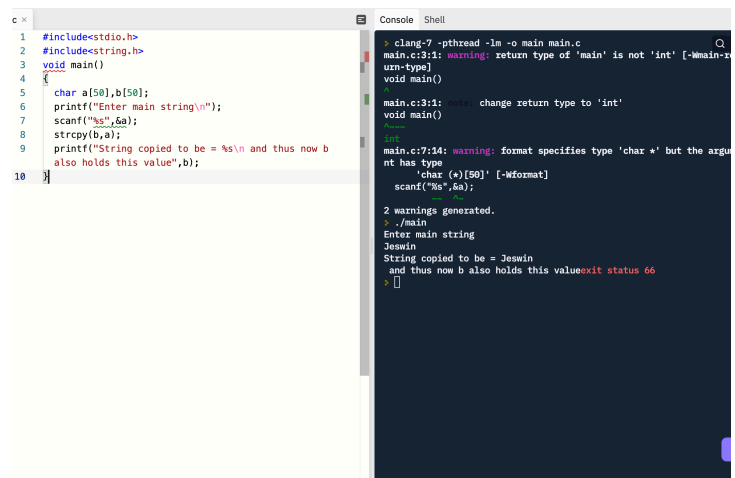
strlen()

2. strcpy()

strcpy() is a standard library function in C/C++ and is used to copy one string to Another.In C it is present in string.

Syntax:

char*strcpy(char*dest,const char*src);



The screenshot shows a code editor on the left and a console window on the right. The code in the editor is as follows:

```
1 #include<stdio.h>
2 #include<string.h>
3 void main()
4 {
5     char a[50],b[50];
6     printf("Enter main string\n");
7     scanf("%s",&a);
8     strcpy(b,a);
9     printf("String copied to be = %s\n and thus now b
    also holds this value",b);
10 }
```

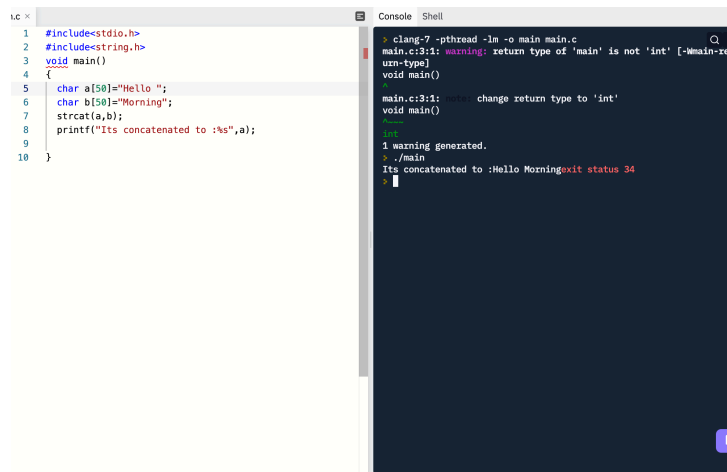
The console window shows the output of the program, including compiler warnings and the execution results:

```
> clang-7 -pthread -lm -o main main.c
main.c:3:1: warning: return type of 'main' is not 'int' [-Wmain-type]
void main()
^
main.c:3:1: note: change return type to 'int'
void main()
^
main.c:7:14: warning: format specifies type 'char *' but the argu
nt has type
      'char (*)[50]' [-Wformat]
      scanf("%s",&a);
      ^
2 warnings generated.
./main
Enter main string
Jeswin
String copied to be = Jeswin
and thus now b also holds this value
exit status 66
```

strcpy()

3. strcat()

The `strcat()` function concatenates `string2` to `string1` and ends the resulting string with the null character. The `strcat()` function operates on null-ended strings.



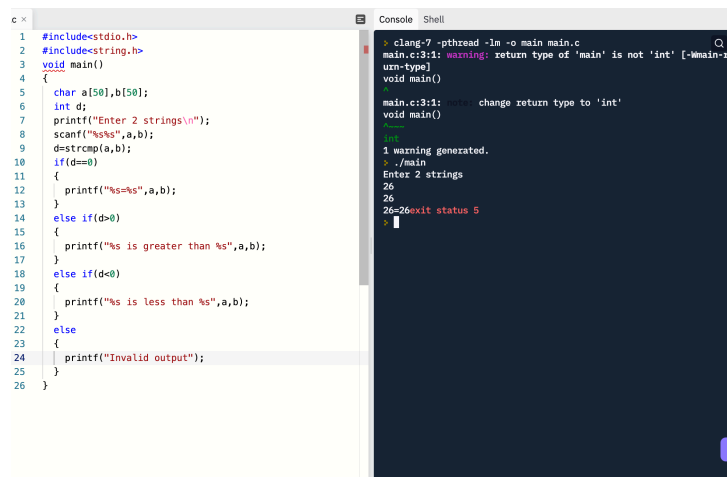
```
1 #include<stdio.h>
2 #include<string.h>
3 void main()
4 {
5     char a[50]="Hello ";
6     char b[50]="Morning";
7     strcat(a,b);
8     printf("Its concatenated to :%s",a);
9 }
10
```

```
> clang-7 -pthread -lm -o main main.c
main.c:3:1: warning: return type of 'main' is not 'int' [-Wmain-se
urn-type]
void main()
^
main.c:3:1: note: change return type to 'int'
void main()
^
1 warning generated.
> ./main
Its concatenated to :Hello Morningexit status 34
```

`strcat()`

4. strcmp()

This function is used to compare the string arguments. It compares strings lexicographically which means it compares both the strings character by character. It starts comparing the very first character of strings until the characters of both strings are equal or Null character is found.



```
1 #include<stdio.h>
2 #include<string.h>
3 void main()
4 {
5     char a[50],b[50];
6     int d;
7     printf("Enter 2 strings\n");
8     scanf("%s%s",a,b);
9     d=strcmp(a,b);
10    if(d==0)
11    {
12        printf("%s==%s",a,b);
13    }
14    else if(d>0)
15    {
16        printf("%s is greater than %s",a,b);
17    }
18    else if(d<0)
19    {
20        printf("%s is less than %s",a,b);
21    }
22    else
23    {
24        printf("Invalid output");
25    }
26 }
```

```
1 clang-7 -pthread -lm -o main main.c
main.c:3:1: warning: return type of 'main' is not 'int' [-Wmain-rtn-type]
void main()
^
main.c:3:1: note: change return type to 'int'
void main()
^
1 warning generated.
./main
Enter 2 strings
20
26=26\xit status 5
```

strcmp()

5. strrev()

The strrev() function is used to reverse the given string.

Syntax:

char*strrev(char*str);Parameter:str:

The given string which is needed to be reversed.

```
main.c ×
1  #include<stdio.h>
2  #include<string.h>
3  void main()
4  {
5      char a[50]="Hello";
6      printf("Before =%s\n",a);
7      printf("After : %s",strrev(a));
8
9  }
```

strrev()

The above output is : olleH

6. strlwr()

The strlwr() function is a built-in function in C and is used to convert a given string into lowercase.

Syntax:

char*strlwr(char*str);Parameter:str:

This represents the given string which we want to convert into lowercase.

```

#include<stdio.h>
#include<string.h>
void main()
{
    char a[50]="JeSwIn"
    printf("In lowercase:\n%s", strlwr(a));
}

```

strlwr()

The above output is : jeswin

7. strupr()

The strupr() function is used to convert a given string to uppercase.

Syntax:

char*strupr(char*str);Parameter:str:

This represents the given string which we want to convert into uppercase.

```
include<stdio.h>
include<string.h>
id main()

char a[50]="jeswin"
printf("In uppercase:\n%s", strupr(a));
```

strupr()

The above output is : JESWIN

created by:

Jeswin Joji

Roll No:22

BCA 1st Yr

