



Fundamentals of Programming: String Functions in C++

Abdul Haseeb

Agenda

- Different ways to create strings
- String header file
- Using predefined string functions from string header

What is a string?

- ▶ Combination of two or more than two characters (alphanumeric)
- ▶ String is also called a character array
- ▶ “Java is much better than C++ version @11”

Libraries for strings in C++

4

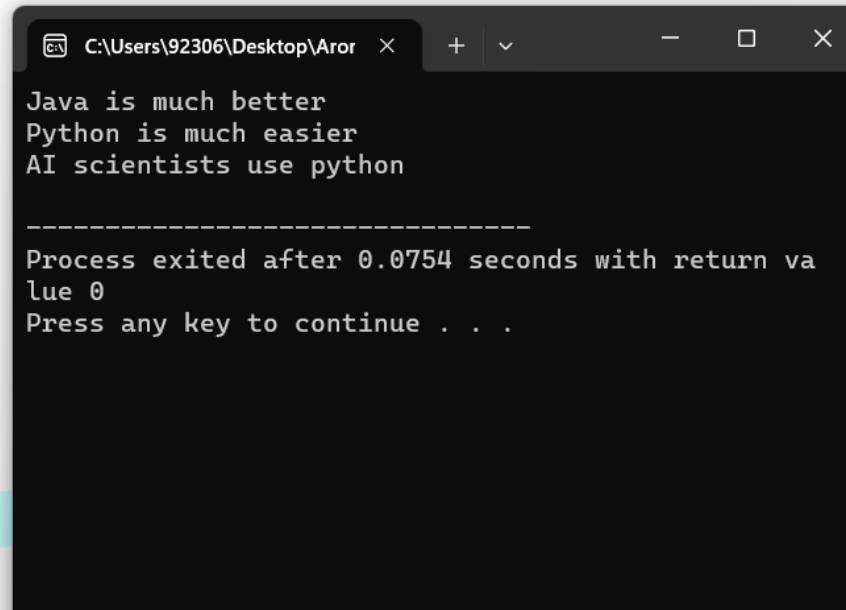
`#include<string>` used
to manipulate C++
strings

`#include<cstring>` used
to manipulate C style
strings

```
string s1("Java is much better");  
cout<<s1<<endl;
```

```
char s2[]="Python is much easier";  
cout<<s2<<endl;
```

```
string s3="AI scientists use python";  
cout<<s3<<endl;
```



```
C:\Users\92306\Desktop\Aror x + - □ x  
Java is much better  
Python is much easier  
AI scientists use python  
  
-----  
Process exited after 0.0754 seconds with return va  
lue 0  
Press any key to continue . . .
```

Different ways to create a string:

```
char greeting[] = "Hello";
```

Following is the memory presentation of above defined string in C/C++ –

| Index | 0 | 1 | 2 | 3 | 4 | 5 |
|----------|---------|---------|---------|---------|---------|---------|
| Variable | H | e | l | l | o | \0 |
| Address | 0x23451 | 0x23452 | 0x23453 | 0x23454 | 0x23455 | 0x23456 |

Actually, you do not place the null character at the end of a string constant. The C++ compiler automatically places the '\0' at the end of the string when it initializes the array. Let us try to print above-mentioned string –

C STYLE STRINGS

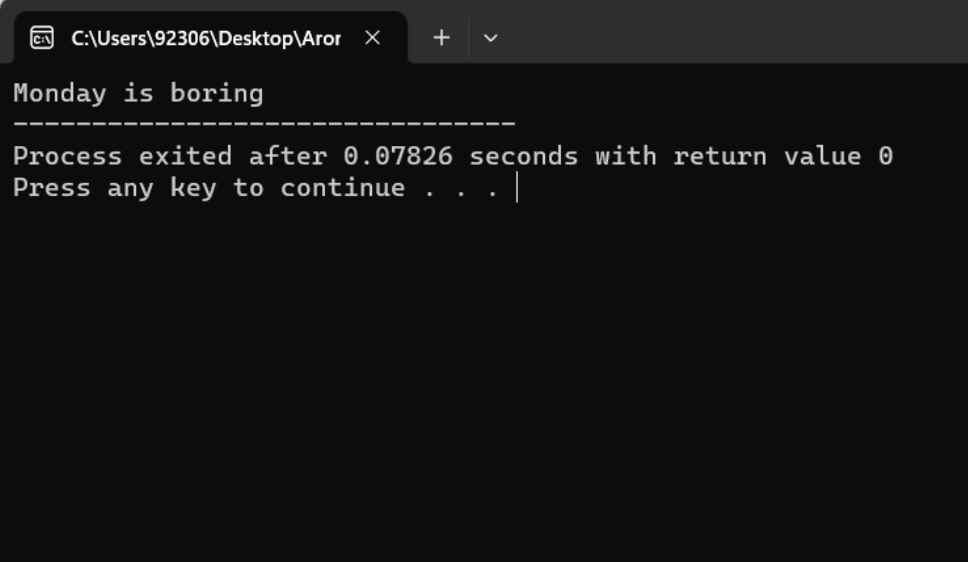
1. strcpy()

- ▶ String copy function copies a string from source location to destination location
- ▶ `strcpy(destination, source)`, available in `cstring`
- ▶ destination and source both must be character array strings.

```
#include<string>
#include<cstring>
using namespace std;

int main(){
char src_str[17]="Monday is boring";
char dst_str[17];
strcpy(dst_str,src_str);
cout<<dst_str;

return 0;
}
```

A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\92306\Desktop\Aror". The window content displays the output of the program: "Monday is boring" followed by a separator line of dashes. Below this, it shows "Process exited after 0.07826 seconds with return value 0" and "Press any key to continue . . . |".

```
C:\Users\92306\Desktop\Aror >
Monday is boring
-----
Process exited after 0.07826 seconds with return value 0
Press any key to continue . . . |
```

Example strcpy(), Why 17 characters

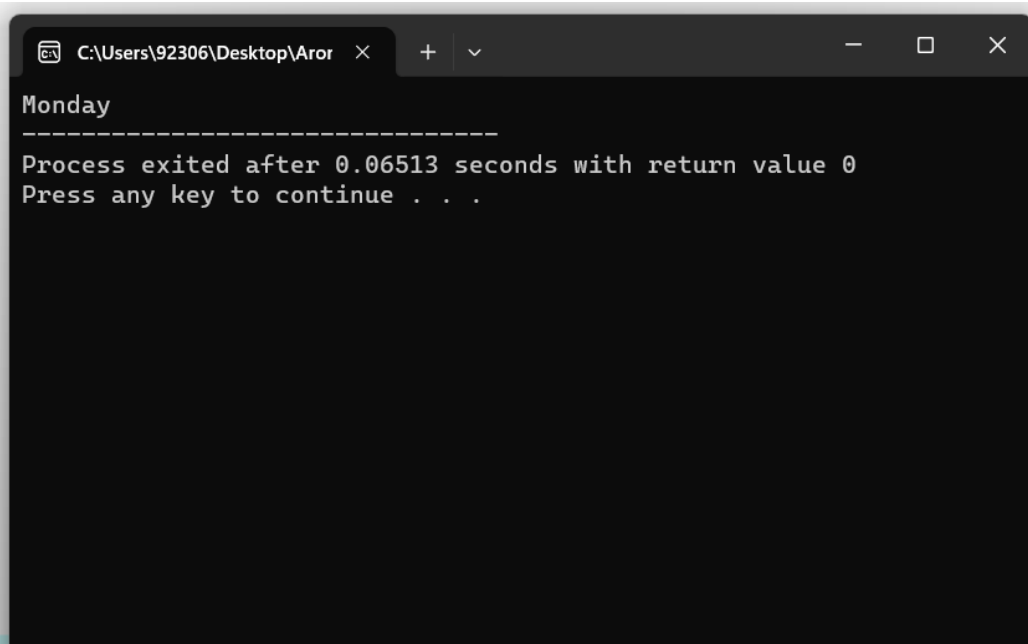
2. strncpy()

9

- ▶ Copies specified number of characters from source to destination
- ▶ `strncpy(Destination, source, number)`
- ▶ Available in `cstring`

```
#include<cstring>
using namespace std;

int main(){
char str1[17]="Monday is boring";
char str2[7];
strncpy(str2,str1,7);
cout<<str2;
return 0;
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\Users\92306\Desktop\Aror". The window displays the output of the program: "Monday" followed by a line of dashes, and then "Process exited after 0.06513 seconds with return value 0" and "Press any key to continue . . .".

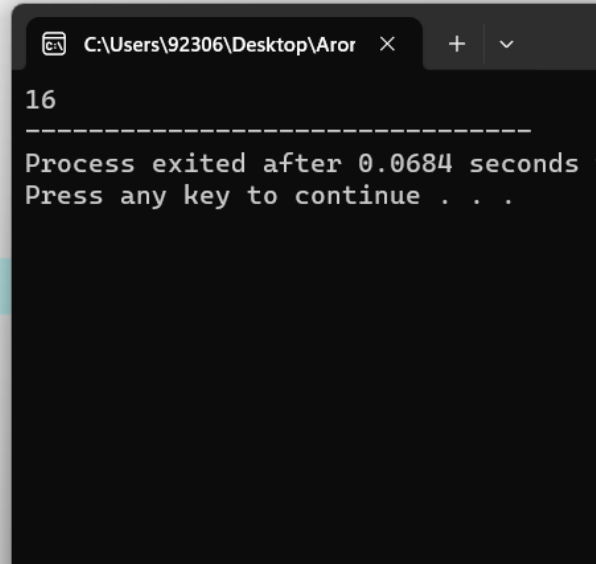
Example strncpy()

3. strlen()

- ▶ It finds the length (number of characters of a string)
- ▶ Available in cstring

```
#include<iostream>
#include<cstring>
using namespace std;

int main(){
char str1[17]="Monday is boring";
cout<<strlen(str1);
return 0;
}
```



```
C:\Users\92306\Desktop\Aror >
16
-----
Process exited after 0.0684 seconds with return code 0
Press any key to continue . . .
```

Example strlen()

4. strcmp

- ▶ Compares two strings lexicographically (from left to right)
- ▶ Which string is greater alphabetically?
 - ▶ string a="Hi";
 - ▶ string b="He";

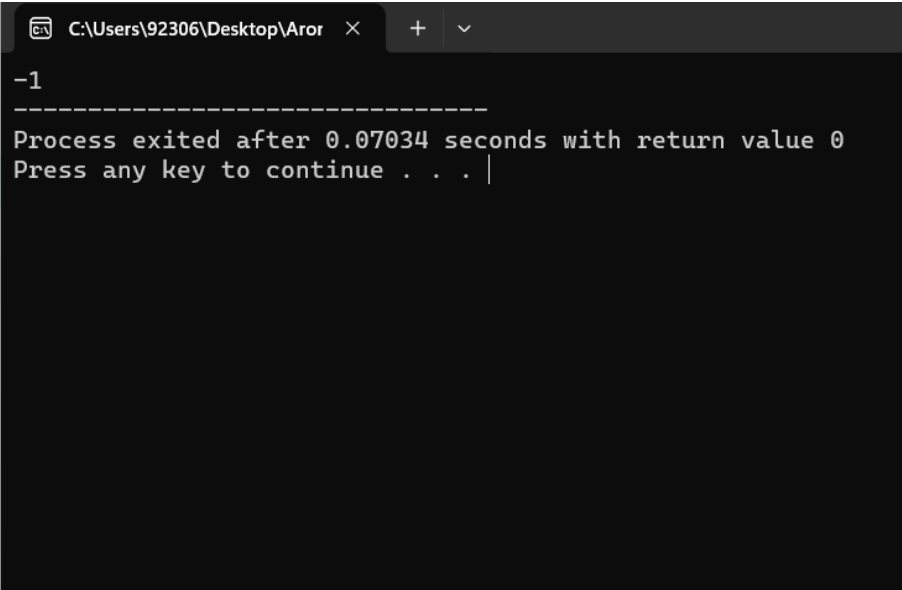
4. strcmp

14

- ▶ strcmp(lhs,rhs)
 - ▶ returns 1 if the first differing character in lhs is greater than rhs
 - ▶ returns -1 if the first differing character in lhs is smaller than rhs
 - ▶ returns 0 if both strings are equal

```
#include<iostream>
#include<cstring>
using namespace std;

int main(){
char str1[3]="Me";
char str2[3]="Mo";
cout<<strcmp(str1,str2);
return 0;
}
```



```
C:\Users\92306\Desktop\Aror  x  +  v
-1
-----
Process exited after 0.07034 seconds with return value 0
Press any key to continue . . . |
```

Example strcmp()

5.strncmp

16

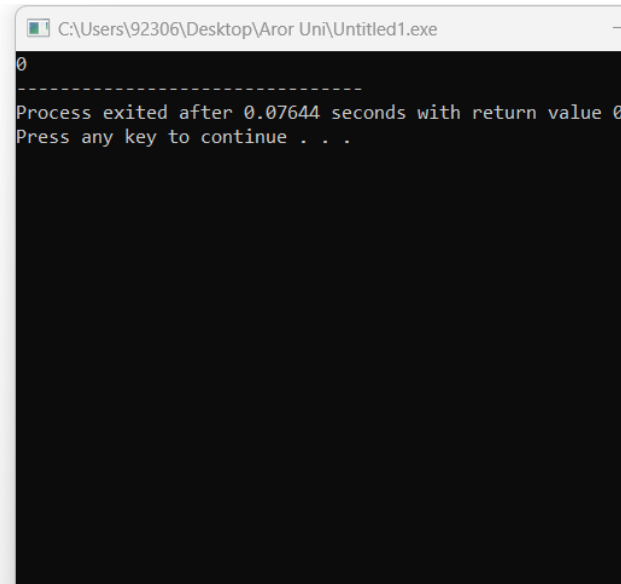
- ▶ Compares n characters from lhs and rhs
- ▶ Other working is same as strcmp()
- ▶ Available in cstring


```
#include<iostream>
#include<cstring>
using namespace std;

int main(){

char s1[7]="Monday";
char s2[5]="More";

cout<<strncmp(s1,s2,2);
```



```
C:\Users\92306\Desktop\Aror Uni\Untitled1.exe
0
-----
Process exited after 0.07644 seconds with return value 0
Press any key to continue . . .
```

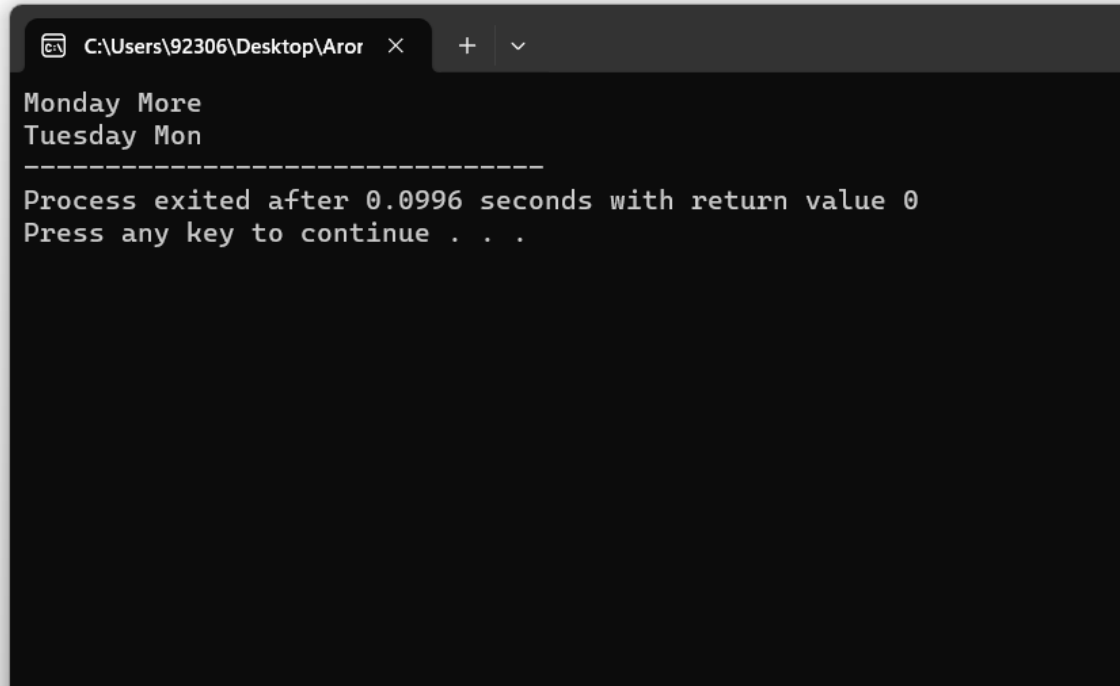
Example

6. strcat() and strncat()

18

- ▶ strcat() Appends the contents of source string into destination string
- ▶ strncat() Appends the n characters from source string to destination string
- ▶ Available in cstring

```
int main(){  
  
    char s1[16]="Monday ";  
    char s2[5]="More";  
    char s3[12]="Tuesday ";  
    strcat(s1,s2);  
    strncat(s3,s1,3);  
    cout<<s1<<endl;  
    cout<<s3;  
  
    return 0;  
}
```



```
C:\Users\92306\Desktop\Aror > C:\Users\92306\Desktop\Aror\Aror.exe  
Monday More  
Tuesday Mon  
-----  
Process exited after 0.0996 seconds with return value 0  
Press any key to continue . . .
```

Example

7. strtok()

20

- ▶ Tokenizes a string
- ▶ It takes a string and a delimiter
- ▶ Available in cstring

```
#include <cstring>
#include <iostream>
using namespace std;

int main() {
    char str[] = "Remember me when you look at the moon!";
    char delim[] = " ";

    cout << "The tokens are:" << endl;

    // tokenize str in accordance with delim
    char *token = strtok(str,delim);

    // loop until strtok() returns NULL
    while (token) {

        // print token
        cout << token << endl;

        // take subsequent tokens
        token = strtok(NULL,delim);
    }

    return 0;
}
```

Code for printing all the tokens

8. strstr()

22

- ▶ Searches for first occurrence of source string in the destination string
- ▶ It takes two strings as parameters
- ▶ Available in cstring

```
#include <cstring>
#include <iostream>

using namespace std;

int main()
{
    char str[] = "Use your brain and heart";
    char target[] = "brain";
    char *p = strstr(str, target);

    if (p)
        cout << "\"" << target << " is present in \"" << str << "\" at position " << p-str;
    else
        cout << target << " is not present \"" << str << "\"";

    return 0;
}
```

Example

string

24

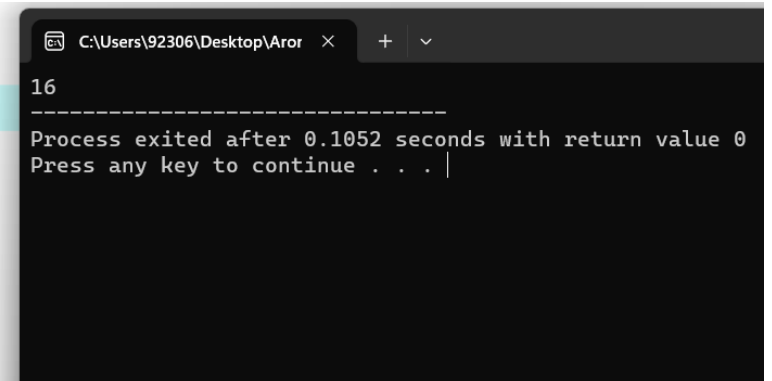
- ▶ String library allows us to work with C++ style strings

C++ string functions

1. size() function

- ▶ size() function is used to find the length (number of characters) of a string
- ▶ Available in string header file

```
string my_str("");  
my_str="Monday is boring";  
int length=my_str.size();  
cout<<length;
```



The screenshot shows a Windows command prompt window with the title bar 'C:\Users\92306\Desktop\Aror'. The output of the program is displayed as follows:

```
16  
-----  
Process exited after 0.1052 seconds with return value 0  
Press any key to continue . . . |
```

2. at() function

- ▶ It gives the index of a character in the string variable

```
#include<iostream>
#include<string>
using namespace std;

int main(){

string s1("TechDev");
cout<<s1.at(5);

return 0;
}
```

```
e
-----
Process exited after 0.0665 seconds with return value 0
Press any key to continue . . .
```

3. insert() function

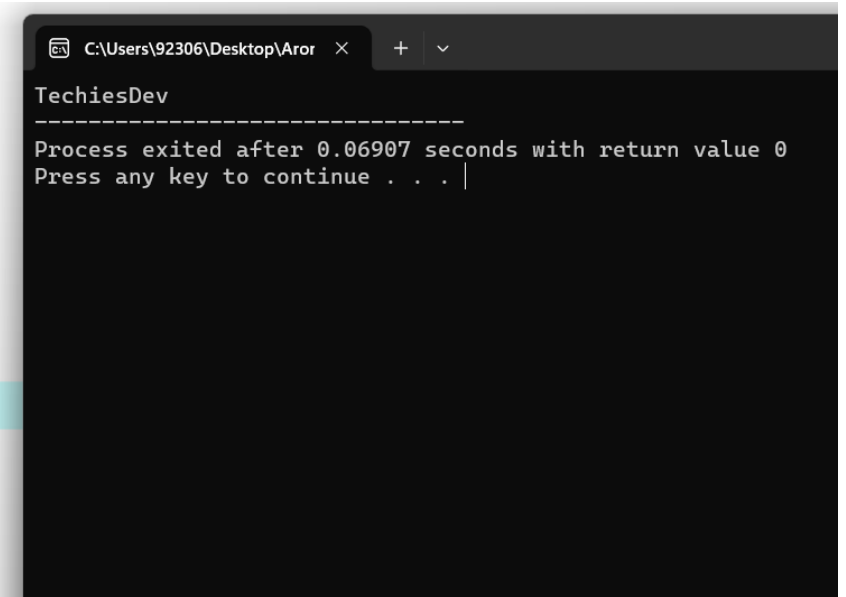
- ▶ Inserts a string value at particular position in a string variable

```
#include<iostream>
#include<string>
using namespace std;

int main(){

    string s1("TechDev");
    s1.insert(4,"ies");
    cout<<s1;

    return 0;
}
```

A screenshot of a Windows command prompt window titled "C:\Users\92306\Desktop\Aror". The window shows the output of a C++ program. It displays "TechiesDev" followed by a separator line, then "Process exited after 0.06907 seconds with return value 0", and finally "Press any key to continue . . .".

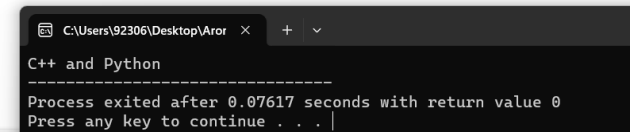
```
C:\Users\92306\Desktop\Aror x + v
TechiesDev
-----
Process exited after 0.06907 seconds with return value 0
Press any key to continue . . . |
```

4. replace()

- ▶ It replaces a substring of a given string, with another string

```
#include<string>
using namespace std;

int main(){
    string string1 = "C++ and Java";
    string string2 = "Python";
    //first number is position
    //second number is the length of replacing string
    string1.replace(8, 6, string2);
    cout<<string1;
    return 0;
}
```

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Users\92306\Desktop\Aror'. The window content displays the output of the C++ program: 'C++ and Python' followed by a separator line '-----'. Below the separator, it says 'Process exited after 0.07617 seconds with return value 0' and 'Press any key to continue . . .'.

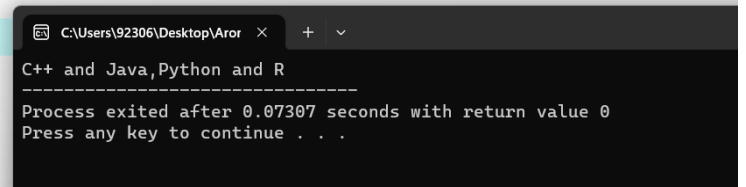
```
C:\Users\92306\Desktop\Aror  x  +  v
C++ and Python
-----
Process exited after 0.07617 seconds with return value 0
Press any key to continue . . .
```

5. append()

- ▶ It appends a string at the end of a given string

```
#include<iostream>
#include<string>
using namespace std;
```

```
int main(){
    string string1 = "C++ and Java,";
    string string2 = "Python and R";
    string1.append(string2);
    cout<<string1;
return 0;
}
```



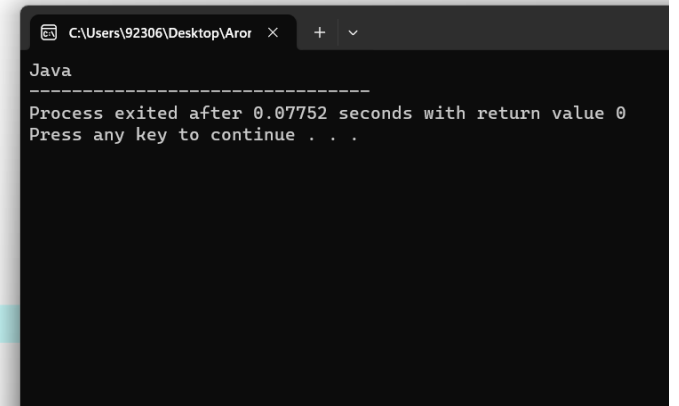
A screenshot of a terminal window showing the output of a C++ program. The window title is "C:\Users\92306\Desktop\Aror". The output displays the concatenated string "C++ and Java,Python and R" followed by a separator line, the execution time, and the return value.

```
C:\Users\92306\Desktop\Aror  x  +  v
C++ and Java,Python and R
-----
Process exited after 0.07307 seconds with return value 0
Press any key to continue . . .
```

6. copy()

```
#include<iostream>
#include<string>
using namespace std;

int main(){
    string string1 = "C++ and Java,";
    char string2[5];
    //string 1 is source
    string1.copy(string2,4,8);
    cout<<string2;
    return 0;
}
```

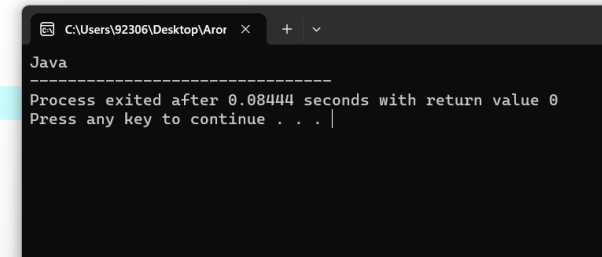


7. substr()

- ▶ Extracts a substring from the given string, we specify the position from where the extraction will start, and length of the substring to be extracted.

```
#include<iostream>
#include<string>
using namespace std;

int main(){
    string string1 = "C++ and Java,";
    //start from position 8 and length 4
    string string2=string1.substr(8,4);
    cout<<string2;
    return 0;
}
```

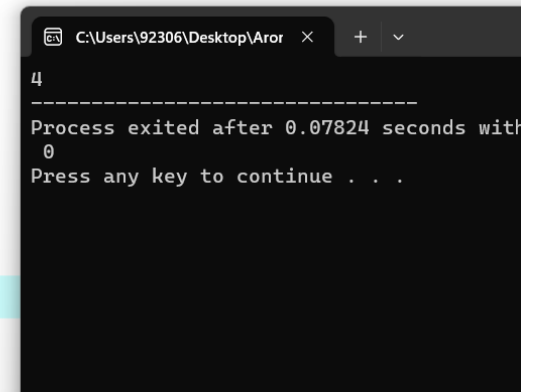


8. find()

- Finds the first occurrence of a given character in the string

```
#include<iostream>
#include<string>
using namespace std;
```

```
int main(){
    string string1 = "C++ and Java,";
    cout<<string1.find('a');
    return 0;
}
```

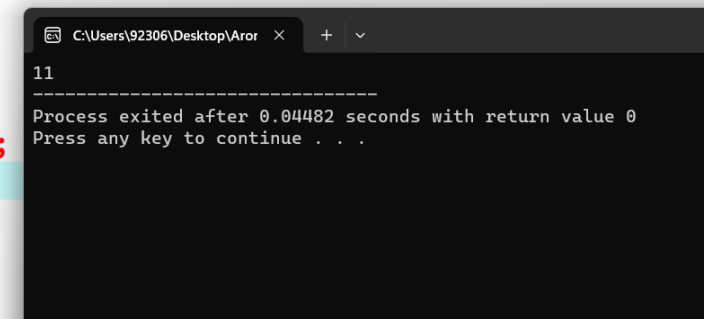


9. rfind()

- Finds the last occurrence of a given character in the string

```
#include<iostream>
#include<string>
using namespace std;

int main(){
    string string1 = "C++ and Java,";
    cout<<string1.rfind('a');
    return 0;
}
```



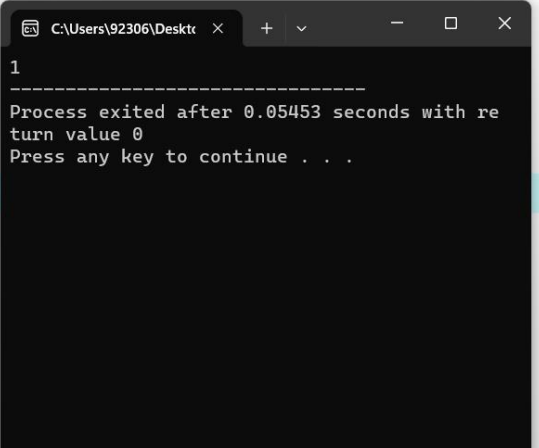
The screenshot shows a Windows command prompt window with the title bar 'C:\Users\92306\Desktop\Aror'. The output of the program is displayed as follows:

```
11
-----
Process exited after 0.04482 seconds with return value 0
Press any key to continue . . .
```

10. compare()

```
#include<iostream>
#include<string>
using namespace std;
```

```
int main(){
    //Returns 1 because string1 is greater
    string string1 = "C++ and Java,";
    string string2= "B-- & Java";
    cout<<string1.compare(string2);
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\92306\Desktop" and standard window controls. The command prompt displays the output of the program: a line with the number "1", followed by a separator line of dashes, and then the message "Process exited after 0.05453 seconds with return value 0" and "Press any key to continue . . .".

10. compare(), for comparing two substrings

```
#include<iostream>
#include<string>
using namespace std;

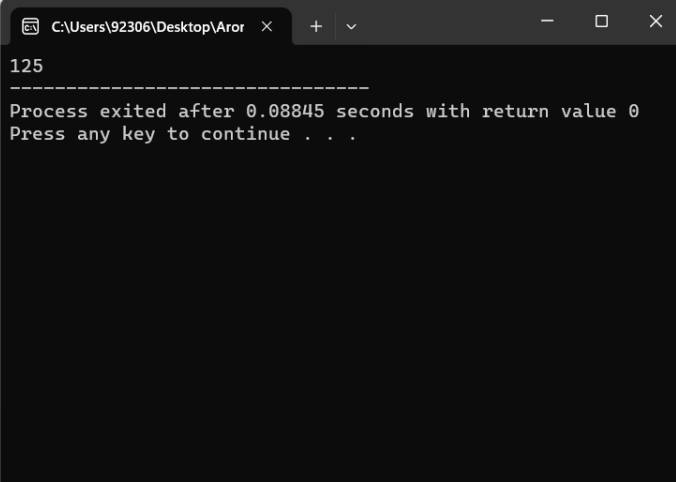
int main(){
    //Returns 0 because both substrings are equal
    string string1 = "C++ and Java";
    string string2= "B-- & Java";
    //start from 8 in string1 and take 4 characters onward
    //start from 6 in string2 and take 4 characters onward
    cout<<string1.compare(8,4,string2,6,4);
    return 0;
}
```

11. stoi

- ▶ Converts string to Integer, it only accepts a numeric string.

```
#include<iostream>
#include<string>
using namespace std;

int main(){
    string s1="123";
    int converted=stoi(s1);
    cout<<converted+2;
    return 0;
}
```

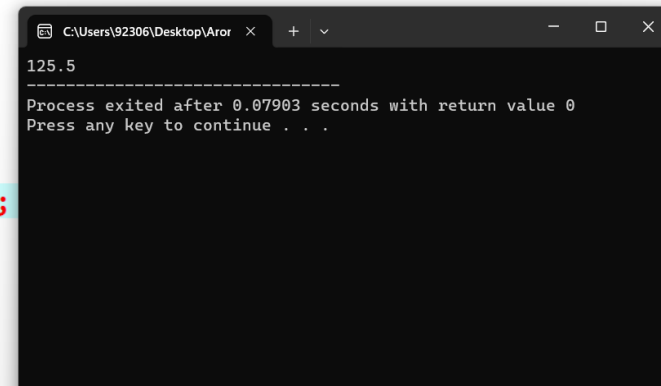


A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\92306\Desktop\Aror" and standard window controls. The command prompt displays the output "125" followed by a separator line "-----". Below the separator, it shows the message "Process exited after 0.08845 seconds with return value 0" and "Press any key to continue . . .".

12. stof,
stod, conert
string to
float, and
double
respectively

```
#include<iostream>
#include<string>
using namespace std;

int main(){
    string s1="123.5";
    float converted=stof(s1);
    cout<<converted+2;
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\92306\Desktop\Aror". The window displays the output of the C++ program: "125.5" followed by a separator line "-----". Below the separator, it says "Process exited after 0.07903 seconds with return value 0" and "Press any key to continue . . .".

13.

to_string()

- Converts various types in their string representation.

38

```
int num = 42;
double pi = 3.14159;

string numAsString = std::to_string(num);
string piAsString = std::to_string(pi);

cout << "Integer as string: " << numAsString << std::endl;
cout << "Double as string: " << piAsString << std::endl;
```

Activity

- ▶ Define a function named `array_sizes`, which takes a string array as its parameter, and prints the length of each array element on the console.