



String as Arrays



Arrays: Review

Now we have data in the `stu[220]` float array, we want to print **Passed** or **Failed** based on student performance.

Condition is: If the marks are greater than 15 print **marks and write Passed** Otherwise **Failed**.

How to do it ?

19	stu[0]
23.5	stu[1]
28.5	stu[2]
15	stu[3]
25	stu[4]
⋮	
5	stu[219]

Arrays: Review

```
#include <iostream>
using namespace std;
string result(float number)
{
    string grade;
    if (number > 15){
        grade = "Passed";
    }
    else{
        grade = "Failed";
    }
    return grade;
}

main(){
    float stu[220];
    for(int x = 0; x < 220; x = x+1){
        cin >> stu[x];
    }
    for(int idx = 0; idx < 220; idx = idx + 1){
        string grade = result(stu[idx]);
        cout << stu[idx] << ": " << grade << endl;
    }
}
```

Working Example: Updated

Now the requirement is to check if a **specific character** is present in the **string** entered by the user or not.

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
Enter a Word: programming
```

```
Enter the character you want to find: m
```

```
m is found in programming
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
Enter a Word: programming
```

```
Enter the character you want to find: l
```

```
l is not found in programming
```

Using the previous knowledge, can we take the input in **string datatype** or we have to use some array?

| String

Before moving to the solution, lets see a **simple** program.

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";

}
```

| String

Can we do **something** like this?

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";
    cout << word[0];

}
```

String

Can we do **something** like this? Lets see the **Output**.

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";
    cout << word[0];

}
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
C
```



|String

Can we do **something** like this? Lets see the **Output**.

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";
    cout << word[2];

}
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
+
```



|String

Can we do **something** like this? Lets see the **Output**.

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";
    cout << word[3];

}
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```



|String: Char Array

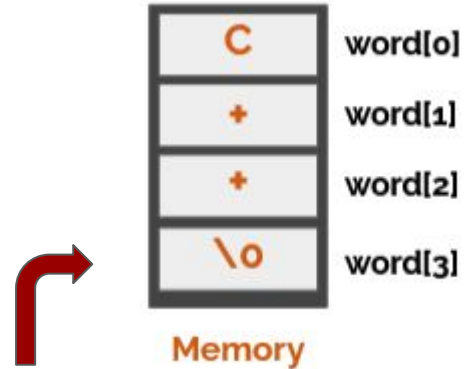
It means, C++ considers **string** as **Character Array**.

```
#include <iostream>
using namespace std;
main(){

    string word = "C++";
    cout << word[3];

}
```

Although "C++" has 3 characters, the null character '\0' is added to the end of the string **automatically**.

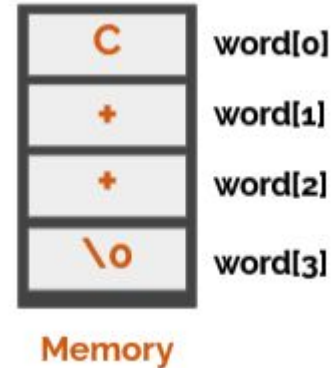


Char arrays are terminated with a **null character**, that is '\0'.

String: Char Array (Initialization)

Different ways of initializing **string** or **character array**.

```
string word = "C++";
```

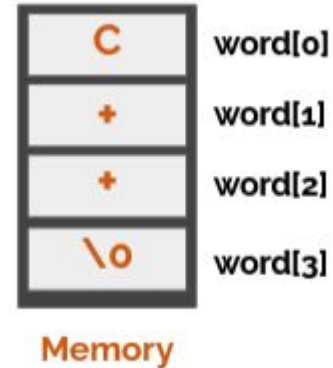


String: Char Array (Initialization)

Different ways of initializing **string** or **character array**.

```
string word = "C++";
```

```
char word[4] = "C++";
```



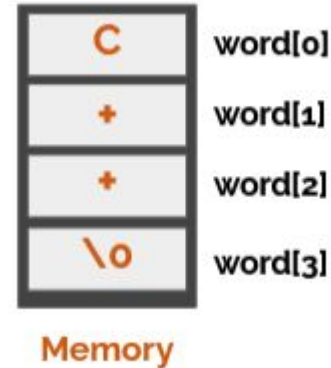
String: Char Array (Initialization)

Different ways of initializing **string** or **character array**.

```
string word = "C++";
```

```
char word[4] = "C++";
```

```
char word[] = "C++";
```



String: Char Array (Initialization)

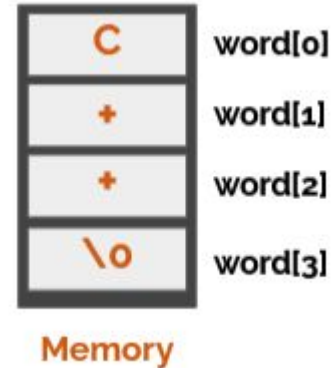
Different ways of initializing string or character array.

```
string word = "C++";
```

```
char word[4] = "C++";
```

```
char word[] = "C++";
```

```
char word[] = {'C', '+', '+', '\\0'};
```



String: Char Array (Initialization)

Different ways of initializing string or character array.

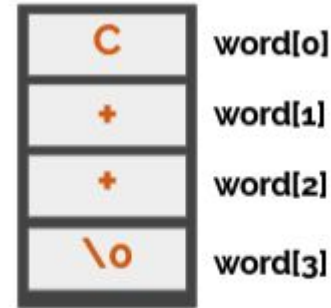
```
string word = "C++";
```

```
char word[4] = "C++";
```

```
char word[] = "C++";
```

```
char word[] = {'C', '+', '+', '\0'};
```

```
char word[4] = {'C', '+', '+', '\0'};
```



Memory

String: Char Array (Initialization)

Different ways of initializing string or character array.

```
string word = "C++";
```

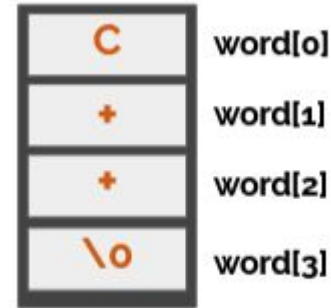
```
char word[4] = "C++";
```

```
char word[] = "C++";
```

```
char word[] = {'C', '+', '+', '\0'};
```

```
char word[4] = {'C', '+', '+', '\0'};
```

```
char word[4] = {'C', '+', '+'};
```



Memory

String: Char Array (Initialization)

Different ways of initializing string or character array.

```
string word = "C++";
```

```
char word[4] = "C++";
```

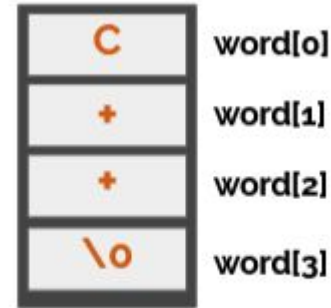
```
char word[] = "C++";
```

```
char word[] = {'C', '+', '+', '\0'};
```

```
char word[4] = {'C', '+', '+', '\0'};
```

```
char word[4] = {'C', '+', '+'};
```

```
char word[100] = "C++";
```



Memory

String: Char Array (Initialization)

Different ways of initializing string or character array.

➡ `string word = "C++";`

➡ `char word[4] = "C++";`

Easy ones.

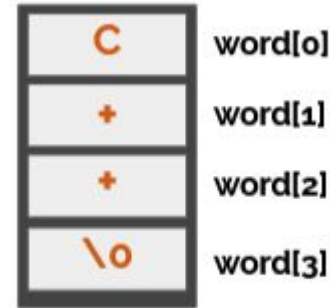
`char word[] = "C++";`

`char word[] = {'C', '+', '+', '\0'};`

`char word[4] = {'C', '+', '+', '\0'};`

`char word[4] = {'C', '+', '+'};`

`char word[100] = "C++";`



Memory

String: Char Array (Output)

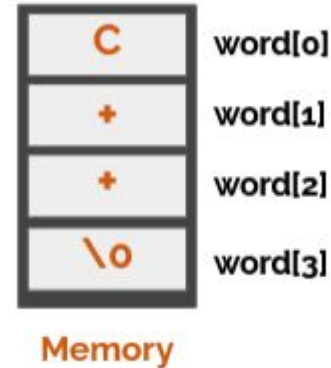
Different ways of displaying Output on the console using **string** or **character array**.

Method 1:

```
cout << word;
```

Method 2:

```
for (int idx = 0; word[idx] != '\0'; idx = idx + 1)
{
    cout << word[idx];
}
```



String: Char Array (Output)

Different ways of displaying Output on the console using **string** or **character array**.

Method 1:

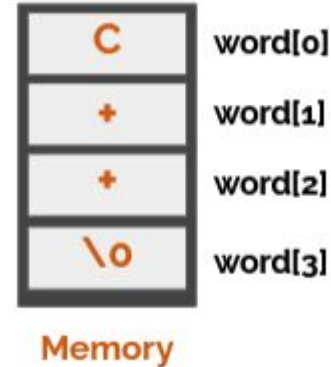
```
cout << word;
```

Method 2:

```
for (int idx = 0; word[idx] != '\0'; idx = idx + 1)
{
    cout << word[idx];
}
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe  
C++
```



String: Char Array (Input)

Simply take input in the `string` or `char array` as before.

→ `string word`

→ `char word[100]`

```
cin >> word;
```

Back to Working Example

Now the requirement is to check if a **specific character** is present in the **string** entered by the user or not.

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
Enter a Word: programming
```

```
Enter the character you want to find: m
```

```
m is found in programming
```

```
C:\C++>c++ example.cpp -o example.exe
```

```
C:\C++>example.exe
```

```
Enter a Word: programming
```

```
Enter the character you want to find: l
```

```
l is not found in programming
```

Using the previous knowledge, can we take the input in **string datatype** or we have to use some array?

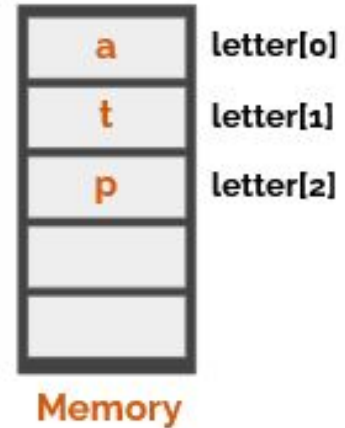
```
#include <iostream>
using namespace std;
bool check(string word, char letter)
{
    bool isFound = false;
    for(int idx = 0; word[idx] != '\0'; idx = idx + 1){
        if (word[idx] == letter){
            isFound = true;
            break;
        }
    }
    if (isFound == true){
        return 1;
    }
    else{
        return 0;
    }
}
```

```
main(){
    string word;
    char letter;
    cout << "Enter a Word: ";
    cin >> word;
    cout << "Enter the character you
want to find: ";
    cin >> letter;
    if(check(word, letter)){
        cout << letter << " is found in
" << word;
    }
    else{
        cout << letter << " is not
found in " << word;
    }
}
```

Character Arrays in C++

We know Character values are stored using **single quotes** as shown below.

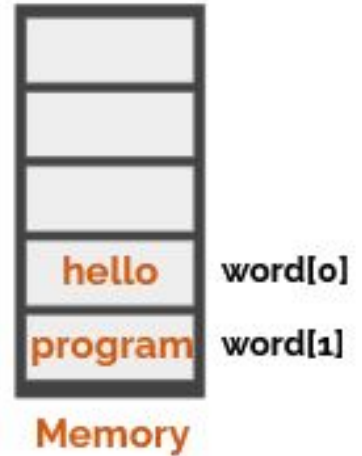
```
char letter[3] = {'a','t','p'};
```



String Arrays in C++

Similarly, String values are stored using **double quotes** as shown below.

```
string word[2] = {"hello", "programming"};
```



String Input: With Spaces

Previously, we encountered the error when we take **input in string with a space** then it gives us the error.

```
string sentence;  
cin >> sentence;
```

String Input: With Spaces

We can resolve this error by using the function `getline`.

```
string sentence;  
getline(cin, sentence);
```

Learning Objective

Declare, initialize and use char arrays to solve real world problems that needs relatively large amount of data.



Conclusion

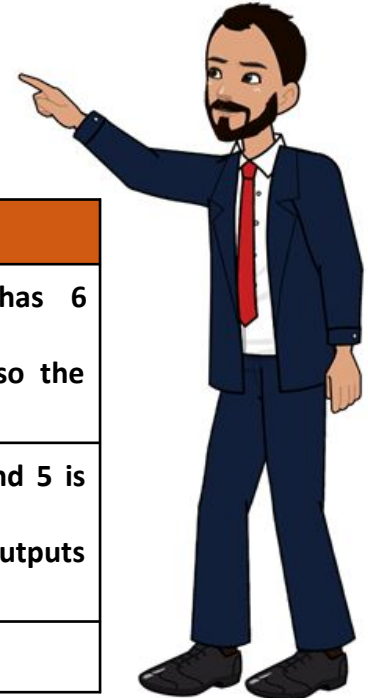
- **Char** arrays are a little different from **int** and **float** arrays.
- **Char** arrays are terminated with a **null character**, that is **'\0'**.
- In C++, character arrays are also called **strings**.

Self Assessment

Write a C++ program that is given a string as input, it displays **true if its length is even** and **false if the length is odd**.

Test Cases:

Input	Output	Explanation
Enter a String: apples	true	// The word "apples" has 6 characters. // 6 is an even number, so the program outputs true.
Enter a String: pears	false	// "pears" has 5 letters, and 5 is odd. // Therefore the program outputs false.
Enter a String: cherry	true	



Self Assessment

Create a C++ program that takes a string (a random name). If the last character of the name is an 'n', it displays **true**, otherwise returns false.

Test Cases:

Input	Output
Enter a String: Aiden	true
Enter a String: Piet	false
Enter a String: Bert	false

