

Programming 4kids

Char Arrays

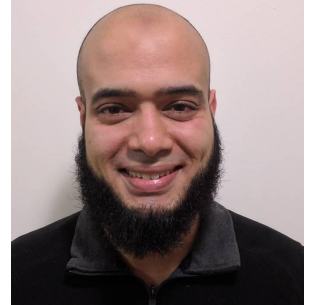
Mostafa Saad Ibrahim

Computer Vision Researcher @ Huawei Canada

PhD - Simon Fraser University

Bachelor / Msc - FCI Cairo University

Ex-(Software Engineer / Teaching Assistant)



Let's refresh about characters

11_01.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     char ch1 = 'A';
6     int ch_value = ch1;
7
8     cout<<ch_value<<"\n";
9     cout<<(int)'A'<<"\n";
10    cout<<(int)'B'<<"\n";
11    cout<<(int)'C'<<"\n";
12    cout<<(int)'Z'<<"\n";
13    cout<<(int)'A' + 26 - 1<<"\n";
14
15    char ch2 = 90;
16    cout<<ch2<<"\n";
17
18    cout<<"***\n";
19
20    cout<<(int)'a'<<"\n";
21    cout<<(int)'b'<<"\n";
22    cout<<(int)'c'<<"\n";
23    cout<<(int)'z'<<"\n";
24    cout<<(int)'a' + 26 - 1<<"\n";
25
26    cout<<('A' < 'a')<<"\n";
27    return 0;
28 }
```

<terminat

```
65
65
66
67
90
90
Z
***
97
98
99
122
122
1
|
```

Check and Convert Chars

```
11_02.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     char ch1 = 'D';
6
7     if ('A' <= ch1 && ch1 <= 'Z') {
8         cout << ch1 << " is an upper case\n";
9         ch1 = ch1 - 'A' + 'a';
10        cout << ch1 << " now is a lower case\n";
11    } else if ('a' <= ch1 && ch1 <= 'z')
12        cout << ch1 << " is already a lower case\n";
13    else if ('0' <= ch1 && ch1 <= '9')
14        cout << ch1 << " is a digit\n";
15    else
16        cout << ch1 << " is neither a digit nor a letter\n";
17
18    return 0;
19 }
```

Problems Console Tasks Properties Call Graph Search

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cp
D is an upper case
d now is a lower case

- Always remember they are just numbers internally
- If we have letter 'A'
 - Subtract 'A'
 - Now this is zero
 - Now add 'a'
 - This shifts to 'a'
 - And so on
 - If 'D' \Rightarrow 'D' - 'A' = 3

Accessing string array

11_03.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     string name = "Hany";
6
7     int sz = name.size(); // called function/method
8     cout << sz << "\n"; // 4
9
10    cout << name << "\n";
11
12    for (int i = 0; i < sz; ++i)
13        cout << name[i]; // internally array
14
15    return 0;
16 }
17
```

Problems Console Tasks Properties Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/ecl

4

Hany

Hany|

- We used to read/write string as an item
- String is sequence of characters
- So actually internally an array

Let's create char array

```
11_03.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int numbers[5] = { 1, 2, 3, 4, 5 };
6
7     char name1[5] = { 'H', 'a', 'n', 'i' }; // 5 not 4
8     char name2[5] = "Hani";
9
10    string name3 = "Hani";
11
12    cout << name1 << "\n";
13    cout << name2 << "\n";
14    cout << name3 << "\n";
15
16    return 0;
17 }
```

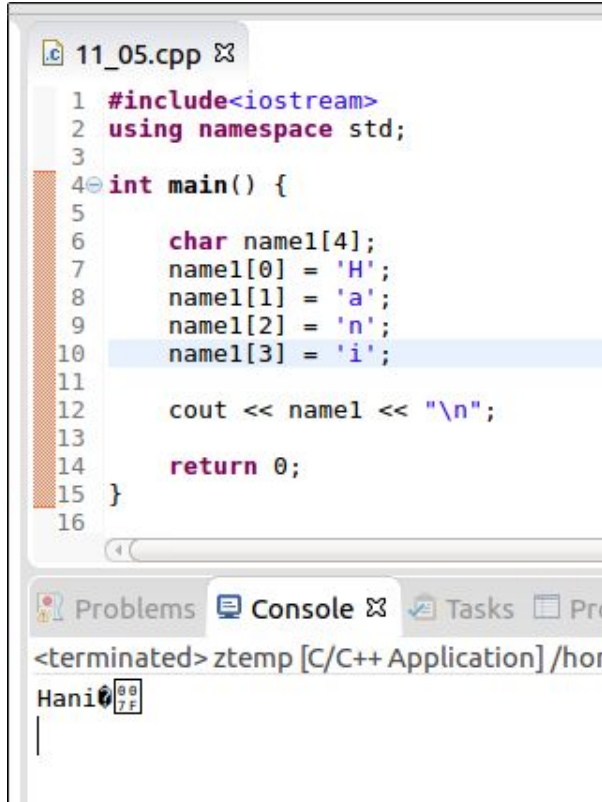
- We can create array of integers or doubles!
- Let's create array of chars
- Length must be 1 + intended length
- Usually, you will use string as easier way
 - Internally has char array

Problems Console Tasks Properties 1010 0101 Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/e

Hani
Hani
Hani

The null char



```
11_05.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     char name1[4];
7     name1[0] = 'H';
8     name1[1] = 'a';
9     name1[2] = 'n';
10    name1[3] = '\0';
11
12    cout << name1 << "\n";
13
14    return 0;
15 }
16
```

Problems Console Tasks Pro

<terminated> ztemp [C/C++ Application] /hor

Hani

- For internal reasons, C++ wants you to tell when the string ends (for easy print)
- We add extra char to do so
- On left, wrong way

The null char

- The right way
 - 1) 1+size (4 here)
 - 2) Null char: `\0`

11_06.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     char name1[5];
7     name1[0] = 'H';
8     name1[1] = 'a';
9     name1[2] = 'n';
10    name1[3] = 'i';
11    name1[4] = '\0';    // Null character
12
13    cout << name1 << "\n";
14
15    return 0;
16 }
```

Problems Console Tasks Properties 1010 0101 Call C

<terminated> ztemp [C/C++ Application] /home/moustafa/wo
Hani

The null char

```
11_07.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     char name1[5];
6     name1[0] = 'H';
7     name1[1] = '\0';
8     name1[2] = 'n';
9     name1[3] = 'i';
10    name1[4] = '\0';    // Null character
11
12    cout << name1 << "\n";
13
14    return 0;
15 }
16
```

Problems Console Tasks Properties

<terminated> ztemp [C/C++ Application] /home/moustafa/v

H

- Print stops once sees the null char
 - Letters after it won't be printed

Reading string with spaces

- Based on the data type, you can read in 2 ways
- Again, we typically use string

11_07.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     string name1;
7     getline(cin, name1);
8     cout<<name1<<"\n";
9
10    char name2[50];
11    cin.getline(name2, 50);
12    cout<<name2<<"\n";
13
14
15    return 0;
16 }
17
```

Problems Console Tasks

<terminated> ztemp [C/C++ Application]

```
I am mostafa
I am mostafa
I am from Egypt
I am from Egypt
```

Array of strings

11_09.cpp

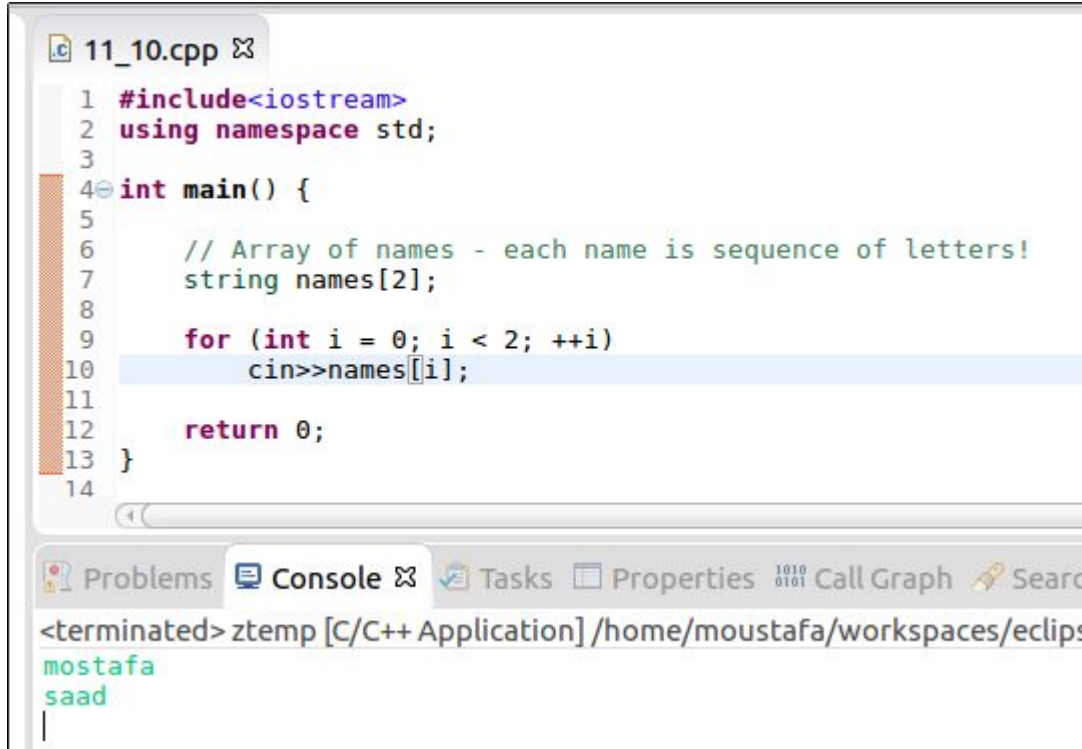
```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     // Array of names - each name is sequence of letters!
7     string names[5] = {"Mostafa Saad", "Never Ever", "Hello world"};
8
9     for (int i = 0; i < 5; ++i)
10         cout<<names[i]<<"\n";
11
12     return 0;
13 }
14
```

Problems Console Tasks Properties Call Graph Search

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cpp/ztemp
Mostafa Saad
Never Ever
Hello world

- We can make array of strings!

We can read them



The screenshot shows an IDE window with a C++ file named 11_10.cpp. The code defines a main function that reads two strings into an array. The console output shows the program has terminated and displays the input strings 'mostafa' and 'saad' on separate lines.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     // Array of names - each name is sequence of letters!
7     string names[2];
8
9     for (int i = 0; i < 2; ++i)
10         cin>>names[i];
11
12     return 0;
13 }
14
```

Problems Console Tasks Properties Call Graph Search

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse

mostafa
saad
|

- Using cin
- Or use getline for spaces

Escape characters

```
11_11.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     // Escape characters
7     cout<<"hello\tworld\n";
8     cout<<"\0";
9     cout<<"Let's print a double quote \" ";
10
11     return 0;
12 }
13
14
```

Problems Console Tasks Properties 1010 0101 Call G

```
<terminated> ztemp [C/C++ Application] /home/moustafa/wor
hello    world
Let's print a double quote " |
```

- Starts with \
 - \n = new line
 - \t = tab
 - \0 = null
 - \" = “

Definitions

- Let say we have string aaabcdefgg
- **Prefix:** Any string starts from the first character (n prefixes)
 - a, aa, aaa, aaab, aaabc, aaabcdefgg
- **Suffix:** Any string sends at the last character (n suffixes)
 - g, gg, fgg, efgg, Aaabcdefgg
- **Substring:** Starts wherever and end wherever, but **consecutive**
 - E.g. of length 3: aaa, aab, abc, cde, def, efg, fgg. Same as *subarray*.
- **Sub-sequence:** Not **consecutive** but must be in order
 - In order: Next letter must has bigger index
 - adef, bgg, aeg, cdgg
 - aeg indices: 0 5 8
 - But not: gga, ed, aca

Practice Like an array

- Read a string, and print its reverse
- Read a string and print YES if [palindrome](#)
- Read a string and count the [frequency](#) of each digit
- ...
- They are solved the same as a normal array. Just access the array
 - You may need to change the letter
 - E.g. convert digit char '8' to integer number 8
 - E.g. `int digit = char - '0'`

Practice: Count Words

- Read a string that is a statement, e.g I am mostafa saad
- Count how many words here. Words can be separated with spaces/tabs
- Output : 4
- Solution:
 - You can read using getline. Then carefully skip spaces and tabs
 - Either: Just while(cin>>str) count++;
 - Keep reading strings and count them
 - cin>>str by definition skips spaces and tabs

Practice: Conc Strings

- Read two strings S and T. Print a new string that contains the following:
 - First letter of the string S followed by the First letter of the string T.
 - Second letter of the string S followed by the Second letter of the string T.
 - and so on...
- Don't create new strings. Don't change input content
- Input \Rightarrow Output
 - abc defghi \Rightarrow adbecfghi
 - AM CICPC \Rightarrow ACMICPC
- Stop video and code

Practice: Conc Strings

11_12.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string first, second;
6
7     cin >> first >> second;
8
9     int mx_sz = first.size();
10
11     if (mx_sz < second.size())
12         mx_sz = second.size();
13
14     for (int i = 0; i < mx_sz; ++i) {
15         if (i < first.size())
16             cout << first[i];
17
18         if (i < second.size())
19             cout << second[i];
20     }
21
22     cout << "\n";
23     return 0;
24 }
25
```

Practice: Letters Frequency

- Read a string of lower/upper letters. Ignore upper letters and compute the frequency of lower letters. Print ones that exists as below.
- Input: bAAaazzz
- Output
 - a 3
 - b 1
 - z 2
- Stop the video and code it

Practice: Letters Frequency

11_13.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string str;
6
7      cin >> str;
8
9      int frequency[150] = {0}; // initialize to zeros
10
11     for (int i = 0; i < str.size(); ++i)
12         frequency[str[i]]++; // Use char as frequency
13
14     for (int i = 'a'; i <= 'z'; ++i) {
15         if (frequency[i])
16             cout << (char) i << " " << frequency[i] << "\n";
17     }
18
19     return 0;
20 }
21
```

- We can use array of 26 and do operations.
- Here is easier code:
- We know both 'z' and 'Z' are small values. Array of 150 is more than we need. Use bigger array
- Iterate only 'a' to 'z'

Practice: Advanced String Mapping

- Read a string and do the following conversions for its letters
 - If it is an upper letter, don't change
 - If it is lower letter, use this map of 26 letters:
 - abcdefghijklmnopqrstuvwxyz
 - YZIMNESTODUAPWXHQFBRJKCGVL
 - E.g. a \Rightarrow Y and z \Rightarrow L
 - If it is digit, use this map of 10 letters:
 - 0123456789
 - !@#\$%^&*()
- Input \Rightarrow Output
 - acMNmn39 \Rightarrow YIMNPW\$)
 - vwXYZ0123 \Rightarrow KCXYZ!@#\$

Practice: Advanced String Mapping

11_14.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string from = "abcdefghijklmnopqrstuvwxyz0123456789";
6     string to   = "YZIMNESTODUAPWXHQFBRJKCGVL!@#$$%^&*()";
7
8     char letter_map[150] = {0};
9
10    for (int i = 0; i < from.size(); ++i)
11        letter_map[from[i]] = to[i];
12
13    string str;
14    cin >> str;
15
16    for (int i = 0; i < str.size(); ++i) {
17        if ('A' <= str[i] && str[i] <= 'Z')
18            continue;
19
20        str[i] = letter_map[str[i]];
21    }
22    cout << str;
23
24    return 0;
25 }
26
```

Homework 1: Is Prefix?

- Read 2 strings input and str. Print YES if the given str is a prefix for the string. Otherwise, print NO
- Input \Rightarrow Outputs
 - ABCDEFG ABCD \Rightarrow YES
 - ABCDEFG ABCDEF \Rightarrow YES
 - ABCDEFG EFG \Rightarrow NO
 - ABCDEFG DEFG \Rightarrow NO
 - ABCDEFG BCD \Rightarrow NO
 - ABCDEFG DEF \Rightarrow NO
 - ABCDEFG ACEG \Rightarrow NO
 - ABCDEFG DG \Rightarrow NO
 - ABCDEFG GD \Rightarrow NO
 - ABCDEFG ABCDEFG \Rightarrow YES

Homework 2: Is Suffix?

- Read 2 strings input and str. Print YES if the given str is a suffix for the string. Otherwise, print NO
- Input \Rightarrow Outputs
 - ABCDEFG ABCD \Rightarrow NO
 - ABCDEFG ABCDEF \Rightarrow NO
 - ABCDEFG EFG \Rightarrow YES
 - ABCDEFG DEFG \Rightarrow YES
 - ABCDEFG BCD \Rightarrow NO
 - ABCDEFG DEF \Rightarrow NO
 - ABCDEFG ACEG \Rightarrow NO
 - ABCDEFG DG \Rightarrow NO
 - ABCDEFG GD \Rightarrow NO
 - ABCDEFG ABCDEFG \Rightarrow YES

Homework 3: Is Substring?

- Read 2 strings input and str. Print YES if the given str is a substring for the string. Otherwise, print NO
- Input \Rightarrow Outputs
 - ABCDEFG ABCD \Rightarrow YES
 - ABCDEFG ABCDEF \Rightarrow YES
 - ABCDEFG EFG \Rightarrow YES
 - ABCDEFG DEFG \Rightarrow YES
 - ABCDEFG BCD \Rightarrow YES
 - ABCDEFG DEF \Rightarrow YES
 - ABCDEFG ACEG \Rightarrow NO
 - ABCDEFG DG \Rightarrow NO
 - ABCDEFG GD \Rightarrow NO
 - ABCDEFG ABCDEFG \Rightarrow YES

Homework 4: Is Subsequence?

- Read 2 strings input and str. Print YES if the given str is a subsequence for the string. Otherwise, print NO
- Input \Rightarrow Outputs
 - ABCDEFG ABCD \Rightarrow YES
 - ABCDEFG ABCDEF \Rightarrow YES
 - ABCDEFG EFG \Rightarrow YES
 - ABCDEFG DEFG \Rightarrow YES
 - ABCDEFG BCD \Rightarrow YES
 - ABCDEFG DEF \Rightarrow YES
 - ABCDEFG ACEG \Rightarrow YES
 - ABCDEFG DG \Rightarrow YES
 - ABCDEFG GD \Rightarrow NO
 - ABCDEFG ABCDEFG \Rightarrow YES

Homework 5: Convert to number

- Read a string of 6 letters maximum, convert it to int then print
 - the number, the number multiplied by 3
- Input \Rightarrow output
 - "100" \Rightarrow 100 300
 - "0200" \Rightarrow 200 600

Homework 6: Grouping

- Read a string, then divide it to consecutive groups of same letter. Print each group
- Input \Rightarrow outputs.
 - “111222aabb” \Rightarrow 111 222 aa bb
 - HHHH \Rightarrow HHHH

Homework 7: Compressing

- Read a string of letters and then compress each group of same letter
 - E.g. if the sub-string is cccc \Rightarrow c4
 - Use _ between each group
- Input \Rightarrow Outputs
 - ccccDDDxxxxxxxxxe \Rightarrow c4_D3_x8_e1
 - Xxxxxxxx \Rightarrow x8

Homework 8: Compare strings

- Read 2 strings, then output YES if the first is smaller than or equal to the second. Otherwise, output NO
 - Don't use < operator to compare strings. Use loops
- Input \Rightarrow Output
 - aaa aaa \Rightarrow YES
 - aaaaa aa \Rightarrow NO
 - abc d \Rightarrow YES
 - ddddddddddddddd xyz \Rightarrow YES
 - azzzzzzzz za \Rightarrow YES
 - za azzzzzzzz \Rightarrow NO

Homework 9: Add 5555

- [illegible]

Homework 10: Employee Program (v1)

- Help our factory in managing his employees. Create a program that does the following:
 - Display the following choices:
 - Enter your choice:
 - 1) Add new employee
 - 2) Print all employees
 - 3) Delete by age
 - 4) Update Salary by name
 - You will keep the program running forever. Display the choices and user input from 1 to 4

Homework 10: Employee Program (v1)

- For choice 1: Allow the manager to enter information of an employee
 - Ask user to input: Name, Age, Salary and Gender letter
 - Add the information to your database
- For choice 2: Print all employees. Line per employee
- For choice 3: User enter to values, start age and end age
 - Find all employees with: $\text{start_age} \leq \text{age} \leq \text{end_age}$ and remove them
- For choice 4: User enter name, then salary
 - Find the employee and update his salary
- See screenshots

Homework 10: Employee Program (v1)

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
1
Enter name: mostafa
Enter age: 33
Enter salary: 12345
Enter gender (M/F): M
```

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
1
Enter name: Mona
Enter age: 28
Enter salary: 3333
Enter gender (M/F): F
```

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
*****
mostafa 33 12345 M
Mona 28 3333 F
```

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
4
Enter the name and salary mostafa 505
```

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
*****
mostafa 33 505 M
Mona 28 3333 F
```

```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
3
Enter start and end age|
20 30

Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
*****
mostafa 33 505 M
```

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً