

Computer Programming

Dr. Deepak B Phatak

Dr. Supratik Chakraborty

Department of Computer Science and Engineering
IIT Bombay

Session: More on strings

Quick Recap



- Strings are represented using arrays of type char
- C++ provides useful library functions to handle strings

Overview



- In this session, we will discuss some more problems requiring string processing

Two Problems



- Read a string containing first-name and last-name of a person, separated by one or more blank spaces. Separate these two and store each part in a different array
- Read an input line containing multiple words, separated by one or more blank spaces. Separate each word, storing it in appropriate array

First Name – Last Name



Assume that we supply input names as follows:

Nandlal Sardar

Varsha Apte

Dasthaghir

- Note that the last line contains only one name (first or last)

Program design



- We will read and store the given line in a char array `namestr[60]`
- We will use two char arrays to store the two parts of a name, as `firstname[60]` and `lastname[60]`
- Our strategy is to scan the `namestr` array using an index `i`
 - `i` will start with 0, and can go up to length of `namestr`
- First we will assemble and print `firstname`, using an index `j`
 - Search till a non-blank character
- We then skip any following blanks, then assemble and print the `lastname`, again reusing index `j`

A program to find first name and last name



```
#include <iostream>
#include <cstring>
#include <cstdio>
using namespace std;
int main () {
    char namestr[60], firstname[60], lastname[60];
    int nchar, lengthf, lengthl, i, j;
    gets(namestr); nchar = strlen(namestr);
    i=0;
```

Program ... (assemble first name)

```
//Assemble and print first name
for(j=0; i<nchar; i++, j++){
    if (namestr[i] != ' '){
        firstname[j] = namestr[i];
    }
    else break;
}
```


Program ... (print first name)



```
lengthf = j;  
firstname[lengthf] = '\0';  
cout << "first name has " << lengthf << " characters: ";  
cout << firstname << endl;
```

Program ... (Skip consecutive blanks)

```
// skip consecutive blanks, if any
while(namestr[i]!=' ') i++;
if (i == nchar){
    // reached end of namestr, no more names
    cout << "Given string contains only one name" << endl;
    return -1;
}
```

Program .. (Assemble second name)



```
//Assemble and print last name
for(j=0; i<nchar; i++, j++){
    if (namestr[i] != ' '){
        lastname[j] = namestr[i];
    }
    else break;
}
```

Program ... (Print second name)

```
lengthl = j;  
lastname[lengthl] = '\\0';  
cout << "last name has "<< lengthl << " characters: ";  
cout << lastname << endl;  
return 0;  
}
```

Is this a good approach?

- We were told that there are two parts of the name
- What if there are multiple parts or words?
 - As in the second problem
- It will be clumsy to keep defining different arrays to hold different words
- Why not use a two dimensional array (a matrix)
 - Each row can contain one word
 - Have many rows, as required to accommodate all words

Summary



- We studied how to separate two words in a string
- We look forward to a generalized approach