

Introduction to the C programming language

Sorting Algorithms

Giuseppe Lipari
<http://retis.sssup.it/~lipari>

Scuola Superiore Sant'Anna – Pisa

February 16, 2010

Outline

- 1 Root finding
- 2 Triangles
- 3 Recursive problems

Finding the root of an equation

- **Problem:** Given a function $f(x)$, find the roots of the equation $f(x) = 0$
- We have to write a C program that finds the root of a polynomial
 - How to represent a polynomial function?

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

- Iterative methods
 - Find the root starting from points close to it
 - **Secant method:** start from x_0 and x_1
 - **Newton-Raphson method:** start from one point, and from the derivative

Representation

- A polynomial can be represented by an array containing the coefficients
- The program should
 - Read the polynomial from the keyboard or from a file
 - Read the starting point of the iterations
 - How to read an array from the keyboard?
 - How to read an array from a file?
 - Compute the root
 - Ask if the user wants to find another root, else exit

Reading from command line

- It is possible to pass parameters to a C program via a command line
- This can be done by using a special version of the main function

```
int main(int argc, char *argv[])  
{  
    ...  
}
```

Number of parameters on the command line: it includes the program name

List of parameters on the command line: the number of the elements is argc

Example

- Lists all parameters

roots/argcexample.cpp

```
bool isnumber(char str[]) //  
{  
    int i = 0;  
    int len = strlen(str);  
  
    while (i < len) {  
        if (isdigit(str[i])) return true;  
        else if (isspace(str[i])) i++;  
        else if (str[i] == '-' ||  
                 str[i] == '+' ||  
                 str[i] == '.') i++;  
        else return false;  
    }  
}  
  
int main(int argc, char *argv[])  
{  
    int i;  
    // Prints number of arguments  
    cout << "Number of arguments: " << argc << endl;  
  
    //Prints all arguments  
    for (i=0; i<argc; i++) {  
        cout << "argv[" << i << "] = " << argv[i];  
        cout << " --> is a number? "  
            << (isnumber(argv[i]) ? "Yes" : "No")  
            << endl;  
    }  
}
```

Function that returns true if the string contains a number

Prints all arguments
For each one, tests if it is a number.

Find the maximum sum

- This problem has been taken from <http://projecteuler.net/index.php?section=problems&i>
- By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

```
3
7 4
2 4 6
8 5 9 3
```

- Find the maximum sum of a **larger triangle**.

Fibonacci Numbers

- Fibonacci numbers can be defined as follows:
 - $F_0 = 0, F_1 = 1$
 - $F_n = F_{n-1} + F_{n-2}$
- The first numbers in the sequence are:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,

- Well known since ancient times

Fibonacci numbers

- Write two algorithms to generate the n-th Fibonacci number
 - An iterative algorithm
 - A recursive algorithm

The tower of Hanoi

- The tower of hanoi is a simple game
- The goal is to move all disks from the left stick to the right stick
- Rule: it is not possible to put a large disk on top of a small disk

