Problem Solving & C
M. Tech. (CS) 1st year, 2014

Arijit Bishnu
arijit@isical.ac.in

Indian Statistical Institute, India.

July 24, 2014

Outline

Outline

A brief introduction

- C is an imperative language.

A brief introduction

- C is an imperative language.
- Imperative programs consist of

A brief introduction

- C is an imperative language.
- Imperative programs consist of
  - a program state – how is it encoded?

A brief introduction

- C is an imperative language.
- Imperative programs consist of
  - a program state – how is it encoded?
  - instructions/commands that change the program state – what are they? (assignment statements, conditionals, loop, procedures)

A brief introduction

- C is an imperative language.
- Imperative programs consist of
  - a program state – how is it encoded?
  - instructions/commands that change the program state – what are they? (assignment statements, conditionals, loop, procedures)
- Instructions in an imperative language are similar to the native machine instructions of traditional computer hardware.

A brief introduction

- C is an imperative language.
- Imperative programs consist of
  - a program state – how is it encoded?
  - instructions/commands that change the program state – what are they? (assignment statements, conditionals, loop, procedures)
- Instructions in an imperative language are similar to the native machine instructions of traditional computer hardware.
- A bit of history – John von Neumann's *stored program computers* and Eckert & Mauchly's contribution. The main idea is that machine can store both data and instructions indistinguishably.

Outline

A unary operator – sizeof

```
#include<stdio.h>
int main(void){
printf("\nSize of char, unsigned char:> %u, %u",
sizeof(char),sizeof(unsigned char));
printf("\nSize of int, short int, long int:>%u, %u, %u",
sizeof(int),sizeof(short int),sizeof(long int));
printf("\nSize of float:> %u",sizeof(float));
printf("\n Size of double, long double:> %u, %u",
sizeof(double),sizeof(long double));
printf("\nSize of uint:> %u",sizeof(unsigned int));
return 0;
}
```

Exercise

Exercise

Write a program to determine the ranges of `char`, `int`, `float`, `double` both for unsigned and signed cases; also consider `short` and `long` data types, wherever possible.

Outline

Swapping two variables

Problem

Swap the contents of the two variables without using any extra variable.

Swapping two variables

Problem

Swap the contents of the two variables without using any extra variable.

The program

```
#include<stdio.h>
int main(void){
  int a = 6, b = 9;
  a = a - b;
  b = a + b;
  a = b - a;
  printf("\n a = %d, b = %d \n",a,b);
  return 0;
}
```

Swapping two variables

Problem

Swap the contents of the two variables without using any extra variable.

The program

```
#include<stdio.h>
int main(void){
  int a = 6, b = 9;
  a = a - b;
  b = a + b;
  a = b - a;
  printf("\n a = %d, b = %d \n",a,b);
  return 0;
}
```

Any problem?

Can you detect any problem here?

Another way of swapping

Taking recourse to Boolean algebra

XOR is a boolean operation on two Boolean variables x and y, denoted as $x \oplus y$. $x \oplus y = x\bar{y} + \bar{x}y$. Find out what are $y \oplus (x \oplus y)$ and $x \oplus (x \oplus y)$?

Another way of swapping

Taking recourse to Boolean algebra

XOR is a boolean operation on two Boolean variables x and y, denoted as $x \oplus y$. $x \oplus y = x\bar{y} + \bar{x}y$. Find out what are $y \oplus (x \oplus y)$ and $x \oplus (x \oplus y)$?

Exercise

Taking recourse to the above, can you rewrite the program using the `bitwise operator` XOR ^ in `C`?

Another way of swapping

Taking recourse to Boolean algebra

XOR is a boolean operation on two Boolean variables x and y, denoted as $x \oplus y$. $x \oplus y = x\bar{y} + \bar{x}y$. Find out what are $y \oplus (x \oplus y)$ and $x \oplus (x \oplus y)$?

Exercise

Taking recourse to the above, can you rewrite the program using the `bitwise operator` XOR ^ in `C`?

Other bitwise operators

There are other `bitwise operators` & (AND), | (OR), $<<$ (left shift), $>>$ (right shift) and the unary one's complement $\sim$ in `C`.

Outline

Some problems to solve

GCD

The greatest common divisor gcd(a, b) of two positive integers $a >= b$ is the largest natural number of which both a and b are integral multiples. Write a iterative program to compute gcd(a, b).

Some problems to solve

GCD

The greatest common divisor $\gcd(a, b)$ of two positive integers $a >= b$ is the largest natural number of which both a and b are integral multiples. Write a iterative program to compute $\gcd(a, b)$.

Fibonacci sequence

Given a positive integer n, find $F(n)$, where $F(n)$ denotes the Fibonacci sequence. Do not use recurrence.

$$\begin{aligned} F(n) &= 0 \text{ if } n = 0 \\ &= 1 \text{ if } n = 1 \\ &= F(n-1) + F(n-2) \text{ if } n \geq 2. \end{aligned}$$

Another problem

Integer exponentiation

The problem is to raise a real number x to the n-th power, where n is a non-negative integer. First, write a non-recursive program to compute x^n. This method requires $n-1$ multiplications. Now, we will try to do better. Let $m = \lfloor n/2 \rfloor$, and suppose we know how to compute x^m. Then, we have two cases: if n is even, then $x^n = (x^m)^2$, otherwise, $x^n = x \times (x^m)^2$. Now, write a recursive program to compute x^n.

Outline