Input and Output

Inst. Nguyễn Minh Huy

Contents

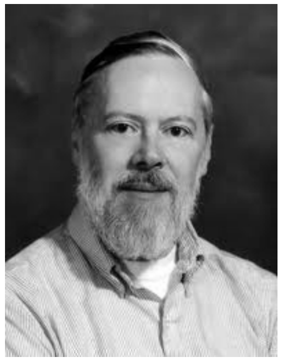
- C/C++ overview.
- C/C++ basic.
- IO stream.

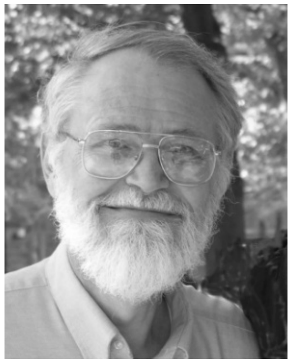
Contents

- **C/C++ overview.**
- C/C++ basic.
- IO stream.

C/C++ overview

- Origin:
  - C (1972):
    - Dennis Ritchie, Brian Kernighan.
    - Simple and minimal.
    - Fast, low-level memory access.
    - Used to write UNIX operating system.
    - Standard: C89 -> C23.
  - C++ (1983):
    - Bjarne Stroustup.
    - Add object oriented programming.
    - Mostly compatible with C.
    - Standard: C++98 -> C++23.


Dennis Ritchie


Brian Kernighan

Bjarne Stroustrup

C/C++ overview

- Basic program structure:

Sections	**Descriptions**

#include <stdio.h>
#include <math.h>
	1. Library declarations.

int   a, b, c;
void sum();
	2. Global variables, function declarations.

int **main**()
{
}
	3. Program entry point.

void sum()
{
}
	4. Function implementations.

C/C++ overview

- Statement:
  - A single command to perform a task.
  - End with semi-colon ‘;’.
  - Compiler ignores spaces, blank lines.
  - Block:
    - A sequence of related statements.
    - Enclosed in curly braces ‘{‘, ‘}’.
  - Compound statement:
    - Contain nested statements or blocks.
    - main() is a giant compound statement.

```
int main()
{
    int   a, b, c;

    a = 100;
    b =
        a /
        2;
    {
        b = b + 5;
        c = a *
            b;
    }
    printf("%d", a, b);
}
```

C/C++ overview

- Comment:
  - Explanation added anywhere in program.
  - Compiler ignores comments.
  - C comment: enclose in /* */.
  - C++ comment: like C or start with //.

```
/* Program to compute
   ampere current.
*/
int main()
{
   double   U, I, R;

   // Formula.
   I = U / R;
}
```

C/C++ overview

- Standard library:
  - KISS principle - **K**eep **I**t **S**imple and **S**tupid.
  - Use library: **#include** <library name>

C libraries	Utilities	C++ libraries
<stdio.h>	Input and output streams.	<iostream>, <fstream>
<math.h>	Math computations and functions.	
<string.h>	String manipulations.	<string>, <algorithm>
<stdlib.h>	Memory management, random numbers.	<vector>, <memory>
<time.h>	Time operations.	<chrono>
<ctype.h>	Character checks and convertions.	
<float.h>	Floating-point numbers.	<limits>
<stdbool.h>	Boolean type.	
…	…	…

C/C++ overview

- Standard library:
  - Math library <math.h>:

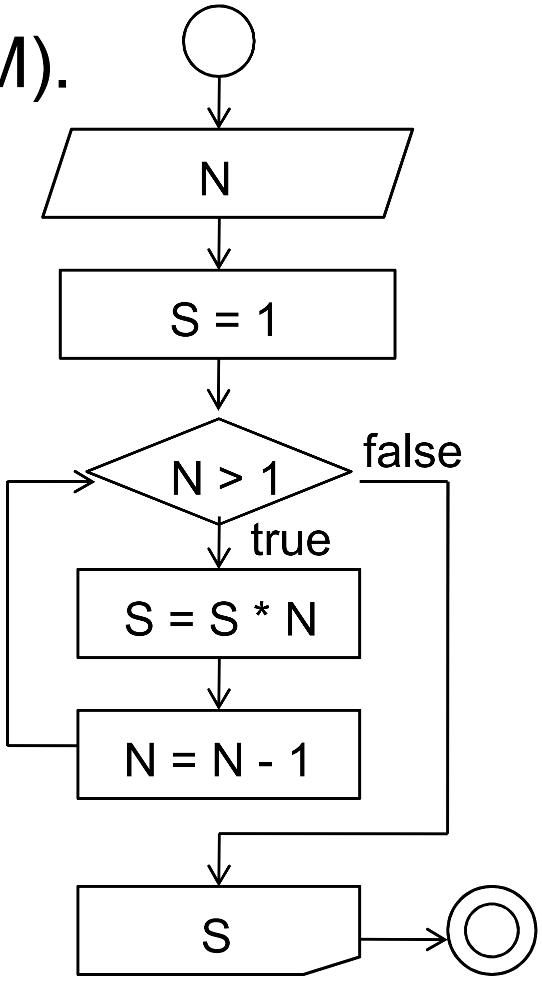
Functions	Descriptions	Examples
sin, cos, tan, atan	Trigonometry	float x = sin(30 * 3.14 / 180);
log, log10, exp	Logarithm	float y = log(exp(5.0));
sqrt, cbrt	Square/cube root	float z = sqrt(2.0);
pow	Exponentiation	float a = pow(2.0, 5);
floor, ceil, round	Round	float b = ceil(2.4);
abs, fabs	Absolute	float c = fabs(a);
…		

Contents

- C/C++ overview.
- **C/C++ basic.**
- IO stream.

C/C++ basic

- Variable and constant:
  - Program stores values during execution.
  - Values are stored in memory units (RAM).
  - Each memory unit is given a name:
    - Variable: changeable value.
    - Constant: unchangeable value.



C/C++ basic

- Variable and constant:
  - Declaration: give name to memory unit before use.
    - Variable: **<data type> <name> [= <value>]**;

```
int     age;
float  gpa = 5.0;
```

    - Constant:

#define <name> <value
const <data type> <name> = <value>; // C++

```
#define PI  3.1416
const int MAX = 1000;                    // C++
```

C/C++ basic

- Variable and constant:
  - Naming rules:
    - Allowed characters: **A-Z, a-z, 0-9, _** (underscore).
    - The **first** character is **not digit**.
    - Avoid **keywords**:
      - C: int, float, char, void, if, else, while, do, for, return, …
      - C++: C keywords, bool, true, false, new, delete, class, …
    - Should be meaningful.

```
int     a, b;                    // Valid, not meaningful.
float  literature, math;         // Valid, meaningful.
char  _letter_123;               // Valid, meaningful.
int     123so, new;              // Invalid.
```

C/C++ basic

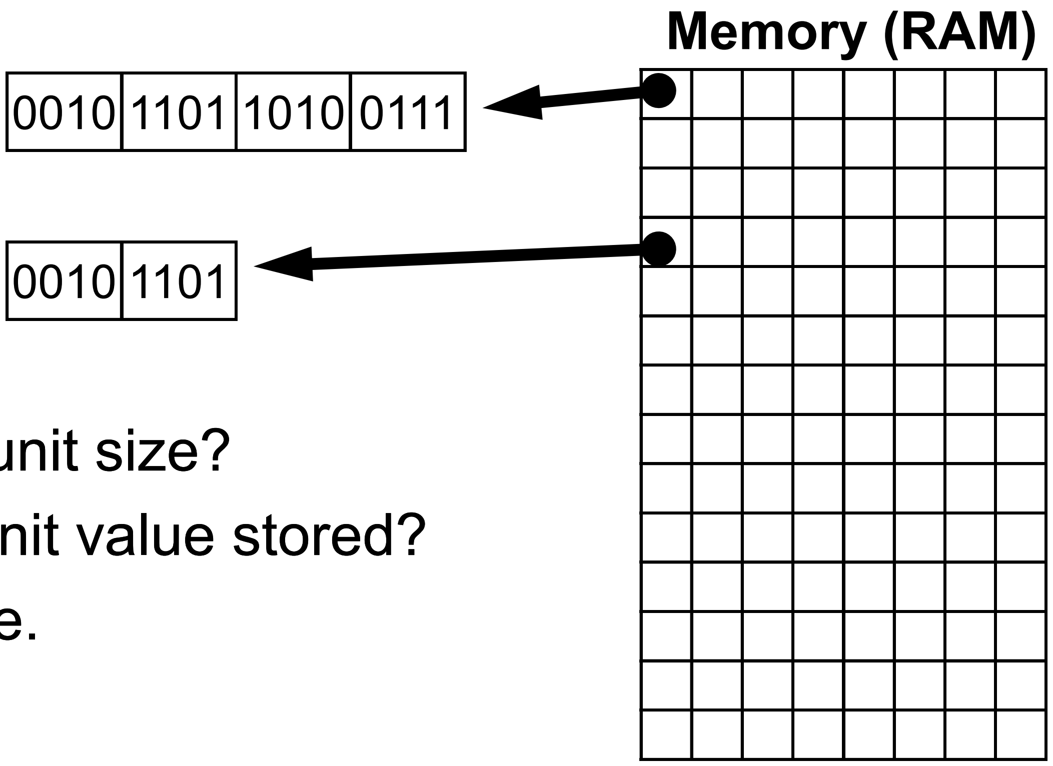
- Data type:
  - What happen when declaring a variable/constant?
    - A memory unit is given a name.

      int a;

      → Memory unit of a:

      char b;

      → Memory unit of b:



    - What is the memory unit size?
    - How is the memory unit value stored?

    ➔ Decided by data type.

C/C++ basic

- Basic data types:

Data types	Descriptions	Size	Range
int, long unsigned int	Integers (normal size)	4 bytes	-2147483648.. 2147483647 0.. 4,294,967,295
long long unsigned long long	Integers (large size)	8 bytes	–9,223,372,036,854,775,808.. 9,223,372,036,854,775,807 0..18,446,744,073,709,551,615
short unsigned short	Integers (small size)	2 bytes	-32768..32767
float	Real numbers (single precision)	4 bytes	3.4E +- 10^{38} (7 digits)
double	Real numbers (double precision)	8 bytes	1.7E +- 10^{308} (15 digits)
char	Character	1 byte	-128..127
bool (C++)	Logic	1 byte	true, false

C/C++ basic

- Expression:
  - Finite sequence of operators and operants.

    a + b – d * c / e
    (x >> (p + 1 - n)) & ~(~0 << n)
  - Operand: variable or constant.
  - Operator:
    - Unary: **<operator> a** → !a, ~b, ++c.
    - Binary: **a <operator> b** → a + b, x >= y.
    - Ternary condition: <condition> **?** <true value> **:** <false value>;
  - Result value is a number.

C/C++ basic

- Expression:
  - Arithmetic:
    - +, -, *, /, %.
    - / depends on operands.
    - % for integer only.

```
int    a = 5 / 3;    // Integral div
float b = 5.0 / 3; // Float div
float c = 5 % 3.0; // Wrong
```

  - Comparisons:
    - >, <, >=, <=, ==, !=.
    - Value: 1 (true), 0 (false).

```
int  a = 5 > 3;        // 1 (true)
int  b = 5 == 3;       // 0 (false)
```

  - Logic:
    - ! (not), && (and), || (or).
    - Connect comparisons.
    - Value: 1 (true), 0 (false).

```
int  a = (5 > 3) && (4 > 7); // 0 (false)
int  b = (5 > 3) ||  !(4 > 7); // 1 (true)
```

C/C++ basic

- Expression:
  - Bitwise:
    - & (and), | (or), ^ (xor).
    - ~ (complement).
    - >> (shift left), << (shift right).
  - Increase/decrease:
    - ++, --.
    - Prefix: before expression.
    - Postfix: after expression.
  - Assignment:
    - =, <operator>=

    → a **=** a **<operator>** b;
    - <operator>: arithmetic, bitwise.

```
short  a = 5 & 6;   // 0101 and 0110
short  b = 5 | 6;   // 0101 or   0110
short  c = 10 >> 1;// 1010 >> 1
```

```
int  a = 5++;        // Wrong
int  b = 5;
int  c = ++b * 4;    // c = 24
int  c = b++ * 4;    // c = 20
```

```
int  a = 5;
int  b, c;
c = b = a;   // b = a, c = b
a *= b + c; // a = a * (b + c)
```

C/C++ basic

- Expression:

Prior.	Descriptions	Operators
1	Unary arithmetic	++, --, +, -
2	Unary logic	!, ~
3	Binary arithmetic	*, /, %
4	Binary arithmetic	+, -
5	Binary bitwise shift	<<, >>
6	Binary comparisons	>, <, ==, !=
7	Binary bitwise logic	&, ^, |
8	Binary logic	&&, ||
9	Ternary condition	? :
10	Assignments	=, +=, -=, …

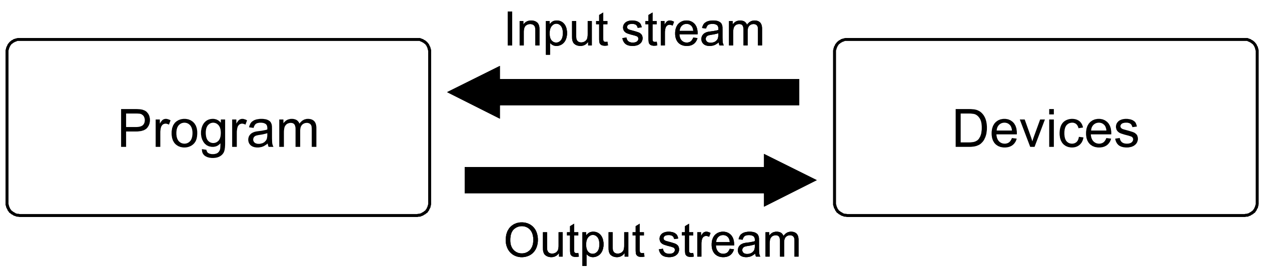
```
int main()
{
    int  a = 1;
    int  b = 2;
    int  c = 3;

    int  d = c++ + - a * ++b;
    int  e = a + b * c >= c + d * a;
    int  f = a - 1 && b + 2 != c >> 1;
}
```

Contents

- C/C++ overview.
- C/C++ basic.
- **IO stream.**

IO stream

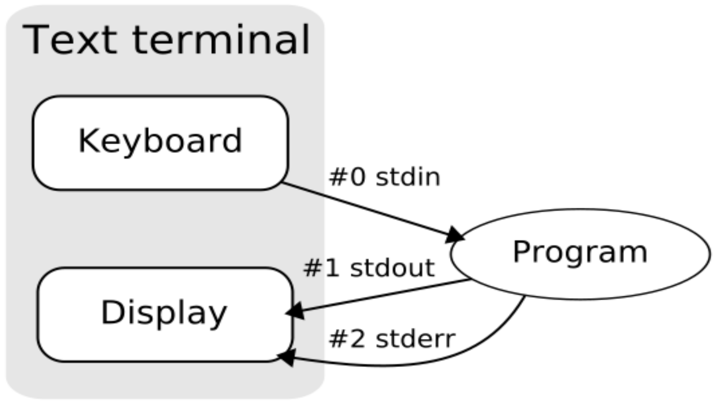
- IO device and stream:
  - Input → Program → Output.
  - IO device: hardware communicating with program.
    - Input devices: keyboard, mouse, file, …
    - Output devices: screen, printer, file, …
  - Stream: connection between program and devices.
    - Input stream: connect to input device to read data.
    - Output stream: connect to output device to write result.



IO stream

- Standard stream:
  - Pre-defined stream in computer program.
    - C: stdin, stdout, stderr (<stdio.h>).
    - C++: std::cin, std::cout, std::cerr (<iostream>).
  - Can be used by default or redirected to other devices.
  - Redirect program standard stream:

```
program.exe < input.txt > output.txt
```



IO stream

- Read/write stream in C:
  - Syntax:
    - Read: **scanf**("<Format>"[, &var1, &var2, …]);
    - Write: **printf**("<Format>"[, var1, var2, …]);
  - Read/write format can be:
    - Text to read or write.
    - Control characters: \n, \t, \\, \", …
    - Type format of variables:

Type format	Descriptions
%d, %lld	Integers: short/int/long, long long
%f, %Lf	Floats: float/double, long double
%c	Character: char
%s	String: char [ ], char *

IO stream

- Read/write stream in C:

```
#include <stdio.h>

int main()
{
        char  name[30];
        float  literature, math;

        printf("Enter name = ");
        scanf("%s", &name);                                       // Read string.
        printf("Enter literature, math = ");
        scanf("%f %f", &literature, &math);                       // Read 2 floats.

        printf("Hello %s.\n", name);                              // Write string.
        printf("Your GPA is %f.", (literature + math) / 2); // Write float.
}
```

IO stream

- Read/write stream in C++:
  - Syntax:
    - Read: **std::cin >>** var1 [**>>** var2 …];
    - Write: **std::cout <<** <format> [**<<** <format> …];
  - Write format can be:
    - Text or variable to write.
    - Control characters: \n, \t, \\, \”, …
    - Type format is not needed!

IO stream

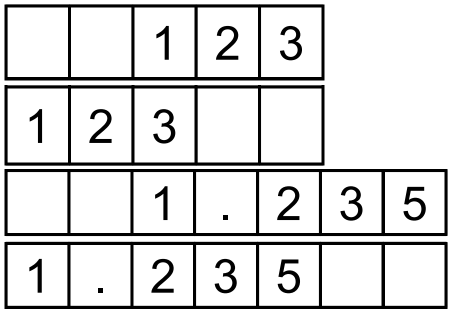
- Read/write stream in C++:

```
#include <iostream>

int main()
{
        char  name[30];
        float  literature, math;

        std::cout << “Enter name = “;
        std::cin >> name;                                          // Read string.
        std::cout << “Enter literature, math = “;
        std::cin >> literature >> math;                            // Read 2 floats.

        std::cout << “Hello “ << name << “.\n”;                    // Write string.
        std::cout << “Your GPA is ” << (literature + math) / 2; // Write float.
}
```

IO stream

- Read/write format:
  - C syntax: % **[-] [n] . [k]** [type format]
    - -: left alignment, n: write width, k: float precision.

```
int    a = 123;
float  x = 1.2345;
printf("a = %5d", a);     // Write a =
printf("a = %-5d", a);    // Write a =
printf("x = %7.3f", x);   // Write x =
printf("x = %-7.3f", x);  // Write x =
```



  - C++ syntax: library <iomanip>
    - Alignment: **std::left**, **std::right**.
    - Write width: **std::setw(n)**.
    - Float precision: **std::setprecision(k)**.
    - C++20: **std::format**(<format>) (library <format>).

Summary

- C/C++ overview:
  - Dennis Ritchie, Brian Kernighan, Bjarne Stroustrup.
  - Statement: end with ;
  - Comment: /* */ or //.
  - Standard library: #include <library name>
- C/C++ basic:
  - Variable/constant: named unit to store value.
  - Data type: size and format of stored value.
  - Expression:
    - Sequence of operators and operands.
    - Evaluated in number.

Summary

- IO stream:
  - Connection between program and devices.
  - Standard stream:
    - Pre-define stream in program.
    - Can be redirected.
    - C: stdin,stdout.
    - C++: std::cin, std::cout.
  - Read/write standard stream:
    - C: scanf, printf.
    - C++: std::cin >>, std::cout <<.

Practice

- Practice 2.1:

Write C/C++ program as follow cho phép:

- Enter person name and birth-year.
- Compute person age and print result.

Notes: enter string with spaces:

- C: scanf("%[^\n]"), fgets.
- C++: std::cin.getline.

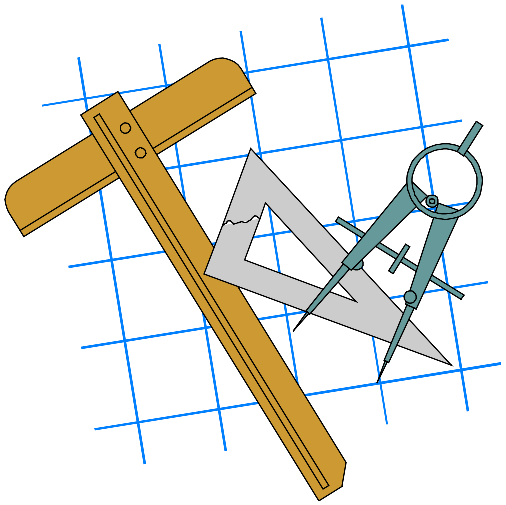
Input format:

Name = Nguyen Van A

Birth-year = 2005

Output format:

*Hello **Nguyen Van A**, now you are **19** years old.*



Practice

- Practice 2.2:

Write C/C++ program to compute lucky number of a car as follow:

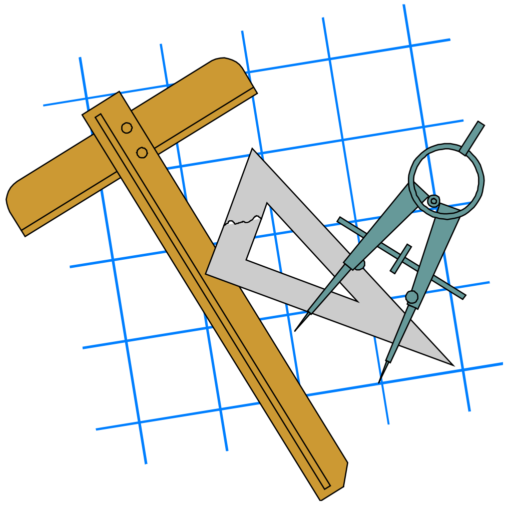
- Enter car registration number (a 5-digit positive integer).
- Compute and print the lucky number.

Input format:

Registration number = 17950

Output format:

Lucky number = 2



Practice

- Practice 2.3:

Write C/C++ program to convert temperature degree as follow:

- Enter temperature degree in Celsius.
- Convert to Fahrenheit and Kelvin degrees, and print results.

Notes:

- Fahrenheit = Celsius * 1.8 + 32.
- Kelvin = Celsius + 273.

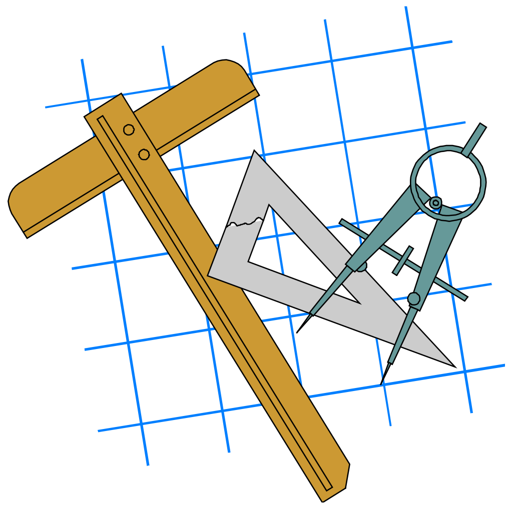
Input format:

Celsius = 25.5

Output format:

Fahrenheit = 77.9

Kelvin = 298.5



Practice

- Practice 2.4:

Write C/C++ program to compute distance of 2 points of time in a day:

- Enter 2 points of time in a day T1 and T2 (hour, minute, second).
- Compute distance (seconds) and print result.

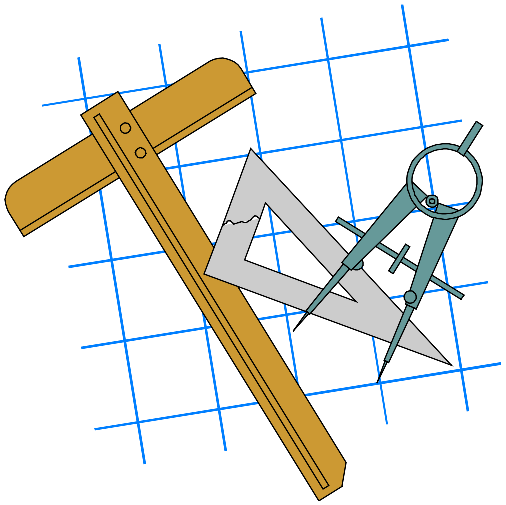
Input format:

T1 (h m s) = 11 3 26

T2 (h m s) = 14 25 18

Output format:

Distance = 12112



Practice

- Practice 2.5:

Cubic equation $x^3 + p^2x + q = 0$ has only one solution:

$$x = \sqrt[3]{\sqrt{\frac{p^6}{27} + \frac{q^2}{4}} - \frac{q}{2}} - \sqrt[3]{\sqrt{\frac{p^6}{27} + \frac{q^2}{4}} + \frac{q}{2}}$$

Write C/C++ program compute the solution of the above equation:

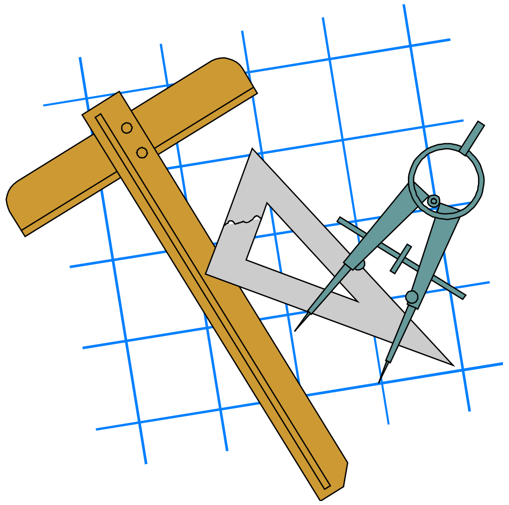
- Enter coefficients of the equation.
- Compute the solution and print result.

Input format:

Enter p, q = 2 3

Output format:

Solution x = -0.673593



Practice

- Practice 2.6:

Write C/C++ program to exchange money as follow:

- Enter an amount of money.
- Print to screen how to exchange the money with:
  + The least necessary money notes.
  + Available money notes: 1000, 5000, 10000, 20000.

Input format:

Exchange money = 94500

Output format:

Note 20000: 4

Note 10000: 1

Note 5000: 0

Note 1000: 4

Remain money = 500