CS1010E: Programming Methodology

Tutorial 02: I/O & Math

30 Jan 2017 - 03 Feb 2017

1. Discussion Questions

(a) [Standard Input] Using `scanf`

i. Write a code to read **one (1)** `integer` number — i. ____________

ii. Write a code to read **one (1)** `real` number — ii. ____________

iii. Write a code to read **two (2)** numbers: the first is `integer` and the second is `real` — iii. ____________

iv. Write a code to read **one (1)** number that can *either* be an `integer` or a `real` number — iv. ____________

(b) [Standard Output] Using `printf`

i. Which of the following code fragment(s) print a *different* output (*if any*)?

A. `printf("Hello\nWorld!\n");`

B. `printf("Hello\n"); printf("World!\n");`

C. `printf("Hello"); printf("\nWorld!"); printf(""); printf("\n");`

ii. Write a code to print a `double` value in 3 decimal places (hint: *use non-scientific notation*)

iii. Write a code to print a `double` value in 3 significant figures (hint: *use scientific notation*)

(c) [Mathematics] Math Library (*using* `#include <math.h>`)

i. Which **two (2)** functions compute the square root of a number? — i. ____________

ii. Which function computes the *absolute* value of a number? — ii. ____________

2. Program Analysis

(a) [Operation Precedence] What is/are the output of code fragments below?

i.
```
int a = 3, b = 4, ans;
    ans = sqrt(a*a+b*b);
    //  = sqrt(3*3 + 4*4)
    //  = sqrt(9   + 16 )
    //  = sqrt(25       )
    //  = 5.0 = 5 ("by truncation")
printf("%d", ans);
```

i. ______ 5 ______

ii.
```
int x1  = 0, y1 = 0, x2 = 0, y2 = 4, x3 = 3, y3 = 0, ans;
    ans = sqrt( pow(sqrt(pow(x1-x2, 2) + pow(y1-y2, 2)), 2) +
                pow(sqrt(pow(x1-x3, 2) + pow(y1-y3, 2)), 2)   );
    // :: pow(x1-x2, 2) = pow(0-0, 2) = 0.0
    // :: pow(y1-y2, 2) = pow(0-4, 2) = 16.0
    // :: pow(x1-x3, 2) = pow(0-3, 2) = 9.0
```

```
        // :: pow(y1-y3, 2) = pow(0-0, 2) = 0.0
        // ans = sqrt( pow(sqrt(0.0) + sqrt(16.0), 2) +
        //             pow(sqrt(9.0) + sqrt(0.0 ), 2)   )
        // :: pow(sqrt(0.0) + sqrt(16.0), 2) = pow(0.0 + 4.0, 2) = 16.0
        // :: pow(sqrt(9.0) + sqrt(0.0 ), 2) = pow(3.0 + 0.0, 2) = 9.0
        // ans = sqrt( 16.0 + 9.0 )
        //     = sqrt( 25.0       )
        //     = 5.0 = 5 ("by truncation")
    printf("%d", ans);
```

ii. 5

iii.

```
int a = 15, b = 25, ans;
ans = sqrt(++a) + sqrt(b++);
//  = sqrt(16 ) + sqrt(25 )   [a = 16, b = 26] ("by increment")
//  = 4.0 + 5.0
//  = 9.0 = 9 ("by truncation")
printf("%d %d %d", a, b, ans);
```

iii. 16 26 9

iv.

```
int CS = 1010; printf("CS%dE", CS);
```

iv. CS1010E

3. Designing a Solution

(a) [Mathematics; Standard Output] Given a triangle described as the length of its three sides S1, S2, and S3, you can compute the total area of the triangle using the Heron's formula given in Equation 1.

$$Area = \sqrt{p \times (p - S1) \times (p - S2) \times (p - S3)} \tag{1}$$

where,

$$p = \frac{S1 + S2 + S3}{2} \tag{2}$$

Write a program to read **three (3)** **real** numbers from the user corresponding to the length of the three sides of the triangle and print the area of triangle rounded to 3 decimal places. Write your program below:

```
int main() {
  double S1, S2, S3, area, p;
  /* Read Input */
  scanf("%lf %lf %lf", &S1, &S2, &S3);

  /* Compute Area */
  p    = (S1 + S2 + S3) / 2;
  area = sqrt(p * (p-S1) * (p-S2) * (p-S3));

  /* Print Area */
  printf("%.3f", area);

  return 0;
}
```

(b) [Mathematics] We revisit the compound interest from last week. Equipped with math library, we can now calculate the balance at the end of each year by using the formula for compound interest given in Equation 3.

$$Final\_Balance = Initial\_Balance \times \left(1 + \frac{Rate}{100}\right)^{Year} \tag{3}$$

However, instead of being given the number of years, you are given the number of months. In this question, you may assume that the interest is only calculated at the end of each year (*thus, ignoring partial year*) and that no external funds is added to the balance.

Given all these information (*as user input*), print the value of the balance at the end of the given *month* rounded to 2 decimal places. Write your program below:

```
int main() {
  double balance, rate; int month;
```

```
   /* Read Input */
   scanf("%lf %lf %d", &balance, &rate, &month)
```

```
   /* Compute Balance */
   int year = month / 12;
   balance *= pow(1.0 + (rate / 100), year);
```

```
   /* Print Balance */
   printf("%.2f", balance)
```

```
  return 0;
}
```

4. Challenge

(a) [Mathematics] Mortgage payment is a tricky business, and a good understanding of that may save your future pension funds. Here, we are trying to slightly help you by creating a program to calculate the *monthly payment* for a mortgage loan (*also called a monthly amortized loan*).

A characteristic of a monthly amortized load is that the interest is *amortized*, that is it have a smooth monthly payment until the loan has been paid off. Often, this type of loan is compounded monthly. Luckily, it can be calculated (*assuming you do not miss any payment*) using the formula in Equation 4:

$$P = \frac{L \times i}{1 - e^{-n \times \ln(1+i)}} \tag{4}$$

where:

- P = monthly payment
- L = principal amount (i.e. *amount borrowed*)
- i = monthly interest rate
- n = number of payments

Write a program that read the user input on L, i, and n, and print the monthly payment rate P rounded to 2 decimal places. Write your program below:

```
int main() {
  double P, L, i; int n;
  /* Your Solution Here */
  scanf("%lf %lf %d", &L, &i, &n);
  P = (L * i) / (1.0 - exp(-1.0 * n * log(1 + i)));
  printf("$%.2f", P); // We are printing the dollar sign for aesthetic



  return 0;
}
```