

To students:

This paper is to be used in week 3 tutorial. Due to time constraint, not all the questions will be discussed in class. Your tutor has the discretion to choose the questions to discuss (or you may request your tutor to discuss certain questions). Please do go through those left-over questions after class.

Please cooperate with your tutor towards a fruitful and enriching learning experience.

Please be reminded that the submission deadline for Problem Set 1 is 9 Feb 2018 6pm!

You just need to submit 10 exercises (including Ex #22) before the deadline and can continue working on the rest after that.

Take note that copy and paste of programs for submission of problem sets is considered plagiarism and offenders will be subject to the disciplinary action of the school.

I. C Basics and Tracing

1. For each of the following question, compose a correct C expression to calculate the desired result.

- (a) Convert degrees Celsius to degrees Fahrenheit, given the formula

$$F = \frac{9}{5}C + 32$$

```
double celsius, fahrenheit;  
  
//Assume celsius is read from keyboard  
  
fahrenheit =
```

- (b) Find out the ratio of length versus width of a rectangle.

```
int length, width;  
  
//Assume length and width are read from user  
  
double ratio =
```

- (c) What did you learn from the above two questions?

2. Being able to follow the execution of a program **without actually running it** is known as **program tracing**. By tracing a program, you will gain a stronger understanding of the execution flow and improve your own coding skill.

Trace the following program fragment, using memory box to represent the value stored in each variable. What is the output?

```
int prev1 = 1, prev2 = 2;

int cur = prev1 + prev2;
printf("cur is %d\n", cur);

prev1 = prev2;
prev2 = cur;
cur = prev1 + prev2;
printf("Now cur is %d\n", cur);
```

3. Trace the program below manually. Determine the values of the variables **a**, **b**, and **c** in the **main** function, and the parameters **a**, **b**, and **c** in the **confuse** function at every step. What is the output of this program?

```
#include <stdio.h>

int confuse(int a, int b, int c);

int main(void) {

    int a = 6, b = 2, c = 5;

    a = confuse(a, b, c);

    printf("a = %d; b = %d; c = %d\n", a, b, c);

    return 0;
}

int confuse(int a, int b, int c) {

    a = b + c;
    c = a * b;

    return c - a + b;
}
```

4. Trace the following program fragment and write down its output.

```
#include <stdio.h>

int main(void) {

    int a = 1, b, c = 1, d = 2;

    a++;
    b = a + 10;
    printf("a = %d; b = %d\n", a, b);

    --a;
    c += a;
    printf("a = %d; c = %d\n", a, c);

    d *= 2 + d;
    printf("d = %d\n", d);

    return 0;
}
```

II. Hands-on Session

5. You may download the program **week3_q5.c** from CS1010E account by issuing the following UNIX command:

```
cp ~cs1010e/tutorial/week3_q5.c .
```



Note the dot at the end of the command. It denotes the *current directory* which is your destination directory to store the copy of the program. The dot must be present, or the **cp** command will not work. If you want to copy the file into a different directory, replace the dot with the directory name accordingly.

The program is replicated below. Compile it without and with the **-Wall** option. What do you observe? What is the output of the program?

```
#include <stdio.h>

int main(void) {

    int i, j = 1;
    double f;

    printf("i = %d; j = %d; f = %f\n", i, j, f);

    return 0;
}
```

Initialize variables!

6. Download the program **week3_q6.c** as follows:

```
cp ~cs1010e/tutorial/week3_q6.c .
```

The program works but is very badly written. Explain how you would improve it.

```
#include <stdio.h>

int main(void) {

    // declare the int variables num1, num2 and num3
    float num1,          num2, num3;

    // ask user to enter two values into num1 and num2
    printf("Enter two real numbers: ");
    scanf("%f %f", &num1, &num2);

    // divide num1 by num2, then multiply the result by num2
    // and then assign the result to num3
    num3 = (num1/num2) * num2;
    printf("num3 = %f\n", num3); // display value in num3
    return 0; }
```

Good typographical style is desired.

Tip: Do you know how to do global indentation of your C program in vim?
The magic vim command is **gg=G**.

7. Compile the program in Q6 above and test it with the following inputs

7 2

- (a) Does the program give correct output?

- (b) Does the program produce correct output, if the following inputs are given?

123.1 2.0

Not all real numbers can be represented accurately in computers.

- (c) What if you declare **num1**, **num2** and **num3** as **double** variables?

8. Freezer Temperature

Write a program **freezer.c** that estimates the temperature (in degree Celsius) in a freezer given the elapsed time (in hours) since a power failure, using the following formula:

$$temperature = \frac{4 * hours^2}{hours + 2} - 20$$

User input will be two integers (separated by a space) indicating the *hours* and *minutes* elapsed since the power failure. You need to convert the inputs into hours before applying the formula.

For example, if the user entered 2 30 (meaning 2 hours 30 minutes), you need to convert it to 2.5 hours before applying the above formula.

Correct your output of real number in two decimal places.

Two sample runs are shown below, with user input highlighted in **bold**.

Sample run #1:

```
Enter hours and minutes since power failure: 0 0  
Temperature in freezer = -20.00
```

Sample run #2:

```
Enter hours and minutes since power failure: 2 30  
Temperature in freezer = -14.44
```