

## CS100 Tutorial 1 (fall semester 2018)

1. Read the following C program and answer the questions below it.

```
#include <stdio.h>
int main() {
    printf("%c ", 'A');
    printf("%c ", 65);
    printf("%c ", 0x41);
    printf("%c ", 0101);
    return 0;
}
```

- (a) What will the above program output? (Look up an ASCII table.)
- (b) What will happen if the conversion specifier of the second printf is changed to **%d**?
- (c) What will be the result if **0x** in the third printf is removed?
- (d) What if the first **0** in the fourth printf is deleted?

**Answer:**

- (a) The output will be: **A A A A**

The constants in all the printf are the various forms of the character 'A' (i.e. the character itself, the decimal form, the hexadecimal form and the octal form of the ASCII code for 'A' respectively).

- (b) If the conversion specifier of the second printf is changed to **%d**, the output will be: **A 65 A A**
- (c) If **0x** in the third printf is removed, the '(' will be printed instead, because 41 (decimal) is the ASCII code for the right parenthesis. So the output will be: **A A ) A**
- (d) If the 0 of the fourth printf is deleted, the output will be: **A A A e**, because value of 101 will be interpreted as a decimal number for ASCII code and it is the ASCII code for character 'e'.

2. Given the following declarations and initial assignments:

```
int i, j, m, n;
float f, g;
i = j = 2;
m = n = 5;
f = 1.2;
g = 3.4;
```

evaluate the following expressions independently (i.e. all variables start with the same set of initial values given above). Show any conversions that take place and the data type of result.

- (a)  $m * j / j$
- (b)  $m / j * j$
- (c)  $(f + 10) * 20$
- (d)  $(i++) * n$
- (e)  $i++ * n$
- (f)  $m = n = --j;$
- (g)  $(int) g * 10$
- (h)  $(int) (g * 10)$
- (i)  $j = i + f$

**Answer:**

- (a)  $m*j/j = 5*2/2 = 10/2 = 5$  (integer)
- (b)  $m/j*j = 5/2*2 = (5/2)*2 = 2*2 = 4$  (integer). Here,  $(5/2)$  has value 2, instead of 2.5, because the division operator for integers results in an integer.
- (c)  $(f + 10) * 20 = (1.2 + 10.0) * 20 = 11.2 * 20.0 = 224.0$  (float). Here, integer 20 is converted to float 20.0 by arithmetic conversion.
- (d)  $(i++) * n = 2 * 5 = 10$  (integer); after this,  $i$  takes value of 3.
- (e) same as (d).
- (f)  $m = n = --j \rightarrow m = (n = (--j)) \rightarrow m = (n = 1) \rightarrow m = 1$ . As a result, each of  $m$ ,  $n$ , and  $j$  is equal to 1.
- (g)  $(int) g * 10 = 3 * 10 = 30$  (integer). Here,  $g$  is converted from float to int by explicit conversion.
- (h)  $(int) (g * 10) = (int) (3.4 * 10.0) = (int) 34.0 = 34$  (integer). Here, 10 is converted from int to float with the arithmetic conversion when multiplying with 3.4; then, 34.0 is converted from float to int by the explicit conversion.
- (i)  $j = i + f = (float)i + f = 2.0 + 1.2 = 3.2 \rightarrow j = (int)3.2 = 3$  (integer). Here,  $i$  is converted from int to float with the arithmetic conversion to a higher ranking, because  $f$  is of type float. Then, 3.2 is converted from float to int through assignment conversion because the left-hand side is  $j$ , which has the type of integer.

3. Write a C program that reads in several lines of non-negative integer numbers, computes the average for each line and prints out that average. A sample input and output session is given below (the italic and underscored numbers are users' input):

Enter the number of input lines: 2

Enter input line 1: 2 4 6 8 -1

Average: 5.000000  
Enter input line 2: 2 6 -1  
Average: 4.000000  
<Program terminates>

The value -1 in each line of user's input is used to indicate the end of input for that line and it is not counted in the average.

**Suggested code:**

```
#include <stdio.h>
int main(void) {
    int total, count, lines, input;
    int i;

    printf("Enter the number of input lines: ");
    scanf("%d", &lines);
    getchar();
    for (i = 0; i < lines; i++) {
        total=0; count=0;
        printf("Enter line %d: ", i);
        scanf("%d", &input);
        while (input != -1) {
            total += input;
            count++;
            scanf("%d", &input);
        }
        double average = (double)total/(double)count;
        printf("Average: %f\n", average);
    }
    return 0;
}
```

4. Write a C program that allows user to interact with the computer to guess whether the next card is higher or lower (called HiLo game). In this game, the first card will be opened first. Then, the user is required to guess the next four cards. If the user guesses all correctly, then he/she will win the game. Otherwise, he/she will lose the game.

A sample input and output session is given below (where the user's input is in red):

```
The first card (between 1 and 13): 11
Guess Higher (H) or Lower (L)
L
The next card (between 1 and 13): 4
Guess Higher (H) or Lower (L)
H
The next card (between 1 and 13): 11
Guess Higher (H) or Lower (L)
L
The next card (between 1 and 13): 8
Guess Higher (H) or Lower (L)
H
The next card (between 1 and 13): 5
You lose
```

**Suggested code:**

```
#include <stdio.h>
#include <time.h>
#define MAXNUM 13
#define MINNUM 1
int main()
{
    int card1 = 0, card2 = 0;
    int i;
    char reply;

    srand(time(NULL));
    card1 = (rand()%MAXNUM ) + MINNUM;
    printf("The first card (between 1 and 13): %d\n", card1);
    for (i=0; i < 4; i++) {
        printf("Guess Higher(H) or Lower(L)\n");
        scanf("%c", &reply);
        getchar();
        srand(time(NULL));
        card2 = (rand()%MAXNUM) + MINNUM;
        printf("The next card (between 1 and 13): %d\n", card2);
        if ((card1 > card2 && reply == 'H') ||
            (card1 < card2 && reply == 'L'))
```

```
        break;
    card1 = card2;
}
if (i == 4)
    printf("You win\n");
else
    printf("You lose\n");
return 0
}
```