



Data Structures and Program Design Using C

SCS 1301

AY 22 - Semester 1



Exercise II

Dr Manjusri Ishwara

1. What would be the output for each print statement in code given below. You are required to dry run first and then execute the code to verify the results.

```
#include <stdio.h>

int main()
{
    int x = 2, y, z;

    x *= 3 + 2;
    printf("%d\n", x);

    x *= y = z = 4;
    printf("%d\n", x);

    x = y = z;
    printf("%d\n", x);

    x = x && (y = z);
    printf("%d\n", x);
    return 0;
}
```

2. What would be the output for each print statement in code given below. You are required to dry run first and then execute the code to verify the results.

```
#include <stdio.h>

int main()
{
    int x, y, z;
    x = 03; y = 02; z = 01;
    printf("%d\n", x | y & z );
    printf("%d\n", x | y & ~z);
    printf("%d\n", x ^ y & ~z);
    printf("%d\n", x & y && z);
}
```

```

x = 1; y = -1;
printf("%d\n", !x | x);
printf("%d\n", ~x | x);
printf("%d\n", x ^ x);

x <= 3;
printf("%d\n", x );
y <= 3;
printf("%d\n", y );
y >= 3;
printf("%d\n", y );

return 0;
}

```

3. What would be the output for each print statement in code given below. You are required to dry run first and then execute the code to verify the results.

```

#include <stdio.h>
int main()
{
    int x, y, z;
    x = y = z = 1;
    ++x || ++y && ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);

    x = y = z = 1;
    ++x && ++y || ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);

    x = y = z = 1;
    ++x && ++y && ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);

    x = y = z = -1;
    ++x && ++y || ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);

    x = y = z = -1;
    ++x || ++y && ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);

    x = y = z = -1;
    ++x && ++y && ++z;
    printf("X = %d\t, Y = %d\t, Z = %d\n", x, y, z);
}

```

4. State which of the following statements are TRUE or FALSE. If FALSE, explain the reason.
- (a) `printf(...)` always begins printing at a newline.
 - (b) All variables in C language must be defined before they are used.
 - (c) It is not mandatory to have a data type for a variable.
 - (d) The remainder operator (%) can be only used with integer operands.
 - (e) Multiplication (*), Addition (+), Subtraction (-), and Division (/) have the same operator precedence.
 - (f) All operators in the C language have left-to-right associativity.
 - (g) To print five (5) lines of output in the standard output, five `printf(...)` statements are required.
 - (h) Integer division is a special name used when dividing integer variables.
 - (i) C programming language considers the variable names `ThisIsAVariable` and `thisisavariab` as the same.
 - (j) In order to use the `printf(...)` function, the programmer has to include the standard input output header file (`<stdio.h>`).
5. Consider three integer variables `a`, `b`, and `c` that is initialized to any value. Write C programs using **only** arithmetic, relational and logic operators to do the following:
- (a) To assign the value one (1) to the maximum valued variable.
 - (b) To assign the value one (1) to minimum valued variable.
6. Write C Programs to achieve the following tasks. Your program should **only** use arithmetic, relational and logic operators for the computation. **No user input is required**. The input values are to be hard coded as a variable.
- (a) Given Celsius ($^{\circ}\text{C}$) value as a variable, find the corresponding Fahrenheit ($^{\circ}\text{F}$) value. Equation for the conversion is $^{\circ}\text{F} = \frac{9}{5} \times ^{\circ}\text{C} + 32$.
 - (b) American Heart Association (AHA) states that the maximum heart rate of an individual is calculated by subtracting an individual's age from 220. The target heart rate for an individual when working out is 50% - 85% of the maximum heart rate. The program should output the maximum and minimum target heart rate.
 - (c) Logic is the backbone of programming. Write a program to find the logical truth of ten (10) statements. Once you've done that, compute and output the final logical truth of all ten statements in the console.
 - (d) Write a program to find the maximum and the minimum possible number of integer, short, and long variables and perform the following bitwise operators. To calculate the maximum and minimum value, use `<limits.h>` header.
 - i. Complement (\sim)
 - ii. left-shift (\ll) by two
 - iii. right-shift (\gg) by two
 - iv. `&` and `|` operators for the maximum and the minimum value of the same data type.
 - (e) Write a program to compute the following. User input is not required.
 - i. Convert an angle given in degrees to radians.

- ii. Given two sides and angles of a triangle, compute the remaining side and the angle.
 - iii. Given two sides and a single angle, compute the remaining angles and the sides.
 - iv. Given a floating point variable, find the floating point remainder.
- (f) For all exercises (a) to (e), compute the time it takes to do the processing in processor time. (Any graphing tool can be used to plot graphs. e.g., MS-Excel, Open Office - Calc)
- i. Compute the processor time for all programs in seconds (S).
 - ii. Run the `clock()` function multiple times to record the value for each execution.
 - A. Plot the values against the execution and draw a regression line through each plot.
 - B. Comment on the plots and the order of the regression line.
 - C. What is `clock_t`?
- (g) Write a program to calculate the amount of tax a person has to pay to the Inland Revenue Department, given the monthly salary income and monthly additional income as a double variable. The tax brackets are as follows: **[This exercise has to be done without using flow control and conditional statements.]**
- i. There is no tax for the salary income, the first 100,000 LKR.
 - ii. For the salary income, each subsequent increase of 50,000 LKR, the tax rate is incremented at 6% blocks to a maximum of 36%.
 - iii. For additional income, the first 100,000 LKR is taxed at 10%, and it is increased by 10% for each subsequent 50,000 LKR to a maximum of 40%.
- iv. **Required Output**
- A. Monthly take-home salary.
 - B. Monthly take-home total income.
 - C. Total annual income.
 - D. Monthly and annual salary tax.
 - E. Monthly and annual additional income tax.
 - F. Total annual income tax to be paid.
7. Take the program written in 6(g), and do the following:
- (a) Introduce an error by changing the code.
 - (b) Observe the compiler messages produced for the error and record them.
 - (c) Fix the error
 - (d) Repeat the steps from (i) to (iii) by introducing new errors to the program. Discover all possible compiler messages that can be produced for the program.

8. SELF-STUDY TASKS

- (a) What is the `puts(...)` function, and how to use it in a C program?
- (b) What similarities does the `puts(...)` function have with the `printf(...)` function?
- (c) What constants are defined in the `<math.h>` header?
