Section 14

PDS Lab **Lab - 1** **08.12.2021**

Instructions:

Give sufficient comment against each statement in your program.
You should save each program with the file name as specified against each problem.
There is a partial credit even if your program does not run successfully for all the test cases as mentioned.
You should upload all the programs (with *.zip*) before leaving the lab session. No Moddle submission will attract zero credit in the evaluation.

1. A *C*-program is given to you, which is given below.

```
/* This program reads two characters using scanf and print the
characters on the screen. */

#include <stdio.h>

int main()
{
      char c1, c2;   // Two variables to store any character
      scanf("%c%c", &c1, &c2);  // Read any two characters
      printf("%c%c", c1, c2);   // Print the characters
      return 0;
}
```

(a) Use an editor to type the program. Save the program as **L1_1.c**
(b) Compile the program so that the program is successfully compiled and a *.out* file is created.
(c) Run the program. To test the program execution, run your program with the following test cases:
Case 1: Type two characters without any blank space between the characters. For example, ab
Case 2: Type two characters with blank space between the characters. For example, 2‘ ’3
Case 3: Type a space followed by *. That is ‘ ‘*
Case 4: Type any 3-digit number, say 123

[Time: 20 Minutes] [4+4+4× 3=20]

2. Write a program to do the following. (Save your program as **L1_2.c.)**
(a) Declare three variables as per the following requirements. You should declare all these variables in **global declaration section**.
- To store a **positive** integer.
- To store any character which you can type through your computer keyboard.
- To store **a very large** floating point number.

(b) Read three values enter through keyboard for each of the variable you have declared.
(c) Print the values on the screen read by your programs with the following specifications.
- Print the integer value with “%2d” format.

- Print the character value with "%d" format.
- Print the floating point value with floating point format. For example, 6.023×10^{23} (say, it is Avogadro's Number)

[Time: 20 Minutes] [6+6+(2+3+3) = 20]

3. A program is given to you which is as follow.

```
/* This program reads a number and print the reciprocal of it. */

#include <stdio.h>
int main ()
{
     int n;
     scanf("%d",&n);        // Read a number, e.g, 5, 0.5, etc.
     printf("%f\n",1/n);
     return 0;
}
```

(a) Edit the program (save the program as **L1_3a.c**) and compile the program.
(b) Run the program with following test cases
   i. Case 1: n = 5
   ii. Case 2: n = .5
   iii. Case 3: n = 0
   iv. Case 4: n = (a very large number, say 2147483647)

(c) Modify the program **L1_3a.c**), so that your modified program (save as **L1_3b.c**) passes most of the test cases.

[Time: 30 Minutes] [4+8+8 = 20]

4. Following is an expression T_p.

$$T_p = T_s \sqrt{\frac{R_s \sqrt{\frac{1-\alpha}{\sigma}}}{2D}}$$

where α = 0.306, T_s = 6.96 × 10^8 m, R_s = 6.96 × 10^8 m, D = 1.496 × 10^{11} m and σ = 1.2.

Write a program to evaluate the value of and then print the evaluated value. Your program should read the value from the keyboard while you run the program. Save your program as **L1_4.c**

[Time: 20 Minutes] [10+10]

5. A ball of mass 125 gram is released from a height of Y meters. Each time it bounces on the floor, its velocity becomes halved. Write a program, which reads the value of Y and prints the total distance

traversed by the ball when it touches the ground for the *n*-th time. Assume that the value of acceleration due to gravity, g is 978 cm/sec^2.

(a) Write a C-program to solve the problem. Save your program as **L1_5.c**
(b) Run your program, with the following input:
   i. Case 1: Y = 100 meter, n=1
   ii. Case 2: Y = 1000 meters, n=100
   iii. Case 3: Y = 50 meters, n= (a very large number)

[Time: 40 Minutes] [12+(2+3+3) = 20]

---*---

Submission instruction

Keep all your program files in a folder, say Lab**1**. Zip the folder. Upload your zip file into the Moodle server. Submission deadline: 11:30 AM