Section 2

PDS Lab **Lab-2** **13.08.2024**

Instructions:

- This lab is based on the topics: IO, Variables, Assignment and Expressions.
- You should save each program with the file name as specified against each problem as <Lab#>-<Assignment#>-<Roll#>.c. For example, **02-01-22CS10006.c** to save Program to 1st assignment in Lab 2 with Roll Number 22CS10006
- You should upload each program to the Moodle system. Also, copy+paste your programs to the text window on the test page.
- You must use the Code::Blocks to write, compile and run your programs.
- There are three problems and the maximum time allowed is 180 minutes.
- **Do not use if-then-else, loop, and arrays.**

1. **Write a program to:**
   a. **Read an integer from the user in an integer variable *n*.**
   b. **Compute the sum of the following series in an integer variable *S1*:**

$$\sum_{1}^{n} n = 1 + 2 + \cdots + n$$

   c. **Compute the sum of the following series in a long variable *S2*:**

$$\sum_{i}^{n} n^2 = 1^2 + 2^2 + \cdots + n^2$$

   d. **Compute the sum of the digits of *n* in an integer variable *S3*.**
   e. **Print the value of *S1*, *S2* and *S3* as the output.**

(Assume: [100 <= *n* <= 999] i.e. *n* is a 3-digit number.)

Test cases:

#	INPUT	OUTPUT
1	n = 110	S1 = 6105 S2 = 449735 S3 = 2
2	n = 309	S1 = 47895 S2 = 9882335 S3 = 12
3	n = 500	S1 = 125250 S2 = 41791750 S3 = 5
4	n = 100	S1 = 5050 S2 = 338350 S3 = 1

```c
// C Program to compute S1,S2,S3 where S1 is the sum of first n natural numbers, S2 is the sum of squares of first n natural numbers, S3 is the sum of digits of n
// Code creator: Nishkal Prakash (nishkal@iitkgp.ac.in)

#include <stdio.h>
int main()
{
    int n;       // n is a three digit number
    int S1 = 0;  // S1 is the sum of first n natural numbers
    long S2 = 0; // S2 is the sum of squares of first n natural numbers
    int S3 = 0;  // S3 is the sum of digits of n
    printf("Enter the value of n: ");
    scanf("%d", &n);                        // Input n
    S1 = (n * (n + 1)) / 2;                 // Cumputing S1
    S2 = (n * (n + 1) * (2 * n + 1)) / 6;   // Computing S2
    S3 = n % 10 + (n / 10) % 10 + n / 100; // Computing S3
    printf("S1 = %d\n", S1);                // Printing S1
    printf("S2 = %ld\n", S2);               // Printing S2
    printf("S3 = %d\n", S3);                // Printing S3
    return 0;
}
```

2. **Write a program to define the following variables:**
   $\alpha = 0.306$, $\sigma = 1.2$, $R_s = 6.96 \times 10^8$ m, $T_s = 1.3654 \times 10^{-11}$ m, $D = 1.496 \times 10^{11}$ m
   Your program should:
   a. **Read the above values from the user.**
   b. **Calculate the value of T_p which is defined by the following expression:**

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

   c. **Print the value of T_p as the output.**

   Compare the output of the above with the case when the values are predefined in the program (i.e. no user input).
   Write your findings and observations as a comment below the end of the program.

Test cases:

#	INPUT	OUTPUT
1	alpha = 0.306 rho = 1.2 Rs = 6.96e8 Ts = 1.3654e-11 D = 1.496e11	Tp = ??
2	No User Input	Tp = ??

```c
// C Program to Tp = Ts * sqrt((Rs*sqrt((1-alpha)/rho))/ (2*D))
// Code creator: Nishkal Prakash (nishkal@iitkgp.ac.in)

#include <stdio.h>
#include <math.h>
int main()
{
    // double alpha = 0.306;
    // double rho = 1.2;
    // double Rs = 6.96e8;
    // double Ts = 1.3654e-11;
    // double D = 1.496e11;
    // double Tp;
    double Ts, Rs, D, alpha, rho, Tp;
    printf("Enter the value of alpha: ");
    scanf("%lf", &alpha); // Input alpha
    printf("Enter the value of rho: ");
    scanf("%lf", &rho); // Input rho
    printf("Enter the value of Rs: ");
    scanf("%le", &Rs); // Input Rs
    printf("Enter the value of Ts: ");
    scanf("%le", &Ts); // Input Ts
    printf("Enter the value of D: ");
    scanf("%le", &D);                                   // Input D
    Tp = Ts * sqrt((Rs * sqrt((1 - alpha) / rho)) / (2 * D)); // Computing Tp
    printf("Tp = %le\n", Tp);                           // Printing Tp
    printf("alpha = %.3lf\n", alpha);                   // Printing alpha
    printf("rho = %.3lf\n", rho);                       // Printing rho
    printf("Rs = %.3le\n", Rs);                         // Printing Rs
    printf("Ts = %.3le\n", Ts);                         // Printing Ts
    printf("D = %.3le\n", D);                           // Printing D
```

```
    return 0;
}
/*
User input Tp = 5.742861e-013
Predefined input Tp = 5.742861e-013
Observation: There is no change in the outputs
*/
```

3. **Assume a particle moves in a two-dimensional xy-plane.**
   Let, the particle start from a point P(x_0, y_0)
   with an initial speed *u*
   along a line inclined to θ degree with the x-axis and
   with a uniform acceleration α.
   The particle reaches a point *Q(x_t, y_t)* after *t* seconds.

   Write a program to:
   a. **Read x_0, y_0, *u*, θ, α from the user.**
   b. **Compute the position *Q(x_t, y_t)***
   c. **Print *Q(x_t, y_t)* on the terminal.**

   The values of *P(x_0, y_0)*, u, θ, α, and t are real and supplied by the user during the execution of your program.

Test cases:

#	INPUT	OUTPUT
1	x0 = 0 y0 = 0 theta = 45 u = 1.41421356237 a = 0 t = 1	Q(Xt,Yt) = Q(1.00,1.00)
2	x0 = 10 y0 = 10 theta = 90 u = 10 a = 0 t = 10	Q(Xt,Yt) = Q(10.00,110.00)
3	x0 = 0 y0 = 0 theta = 0 u = 10 a = 10 t = 1	Q(Xt,Yt) = Q(15.00,0.00)
4	x0 = 0 y0 = 0 theta = 45 u = 10 a = 10 t = 10	Q(Xt,Yt) = Q(424.26,424.26)

```c
// C Program to compute the coordinates of a particle moving in xy-plane with constant acceleration
using the formula d = u*t + 0.5*a*t^2
// Code creator = Nishkal Prakash (nishkal@iitkgp.ac.in)

#include <stdio.h>
#include <math.h>
int main()
{
    double x0, y0, theta, u, a, t;
    double xt, yt;
    double d;
    printf("x0 = ");
```

```
    scanf("%lf", &x0); // Input x0
    printf("y0 = ");
    scanf("%lf", &y0); // Input y0
    printf("theta = ");
    scanf("%lf", &theta); // Input theta as degrees
    // Convert theta to radians
    theta = theta * M_PI / 180;
    printf("u = ");
    scanf("%lf", &u); // Input u
    printf("a = ");
    scanf("%lf", &a); // Input a
    printf("t = ");
    scanf("%lf", &t);                          // Input t
    d = u * t + 0.5 * a * pow(t, 2);           // Computing d
    xt = x0 + d * cos(theta);                  // Computing xt
    yt = y0 + d * sin(theta);                  // Computing yt
    printf("Q(Xt,Yt) = Q(%.2lf,%.2lf)", xt, yt); // Printing the coordinates
}
```

---*---