Section 14

PDS Lab **Lab - 9** **18.01.2023**

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.

1. Write a C program to add two complex numbers by passing structure to a function

   - Declare a structure named complex:

```
typedef struct complex {
     float real;
     float imag;
} complex;
```

   - Read 2 complex numbers Z_1 and Z_2 from the user and store them using the above structure.
   - Create an `add()` function that returns the sum of Z_1 and Z_2

```
complex add(complex z1, complex z2) {...}
```

   - Finally, print the sum of complex numbers in the `main()` function.

For example:

Input:

Z1 = 2.1 -2.3
Z2 = 5.6 23.2

Output:

Z = 7.7 + 20.9i

#	INPUT	OUTPUT
1	Z1: 1 1 Z2: 2 2	Z = 3 + 3i
2	Z1: 1 1 Z2: -1 -1	Z = 0 + 0i
3	Z1: 2.1 -2.3 Z2: 5.6 23.2	Z m = 7.7 + 20.9i
4	Z1: 2 2 Z2: -5 -5	Z = -3 - 3i

[Time: 25 Minutes] [20]

2. Write a program to find if two circles intersect.

- First create a struct named `circle` to represent a circle. It should contain
  - `int x, int y`, to represent the x and y coordinate of the centre
  - `float r`, to represent the radius of the circle

- Now define two circles (two variables of type struct circle), and take input of measurements of two circles.

- Declare a function to check whether the two circles intersect or not.

```
int IsIntersectingCircle(struct circle, struct circle)
```

Hint: Two circles intersect if the distance between their centers is less than the sum of their radii.

#	INPUT	OUTPUT
1	x y r 0 0 5 6 6 4	Circles Intersect
2	x y r 0 0 5 9 9 2	Circles DON'T Intersect

[Time: 25 Minutes] [20]

3. A rectangle is said to be axis-parallel if its sides are parallel to the x and y-axis. An axis parallel rectangle can be fully defined by two points – its top left corner point and bottom right corner point

- Define a C structure named `rectangle` to store an axis-parallel rectangle
  Hint: Store the x-y coordinates of its bottom left and top right corners

- Write a function that determines the area of the rectangle.

#	INPUT	OUTPUT
1	`Rect: (0 0) (2 2)`	`Area: 4`
2	`Rect: (0 0) (3 3)`	`Area: 9`

[Time: 25 Minutes] [20]

4. Create a structure to store details of an employee so as to compute the net salary using the following rule:

a. basic_pay BP = wages*days
b. If BP < 1000, HRA=12% of BP, else it is 20% of BP
c. If number of days present is greater than 19, TA=10% of BP, else TA=0
d. Finally calculate net salary=BP+HRA+TA

Input:

- a positive integer n denoting the number of employees
- Following details of n employees
  - employee name
  - wages per day
  - number of days' present

Output:

- Using the function `netPay(...)` calculate the net pay and print the name and net pay of each employee.

Example:

Input:

- Enter the total number of employee(s) = 1
- Enter name of employee [1] = abc
- Enter wages/day = 150
- Number of days present for employee [1] = 21

Output:

- Name: abc
  Netpay: 4095.00

#	INPUT	OUTPUT
1	1 Name wage days: abc 150 21	Name: abc Netpay: 4095.00
2	2 Name wage days: abc 150 21 xyz 50 10	Name: abc Netpay: 4095.00 Name: xyz Netpay: 560.00

[Time: 30 Minutes] [20]

5. Any date in English calendar can be represented with integers
DD (01-31), MM (01-12) and YYYY (2000 - 2022)

Define structure say `Date` to store such a date. With the `Date` structure definition, define the following functions:

- `struct Date ReadDate( );` //To read a date from the keyboard
- `void PrintDate(struct Date x);`
  //To print a date in the form of dd/mm/yyyy
- `void FindDays (struct Date x,y);`
  // To find and print the number of days between x and y

Write the `main()` function to do the following:

- Call `ReadDate()` 2 times to input 2 dates, say X and Y
  (Assume valid date will be provided as input)
- Call `PrintDate()` to print two dates X & Y
- Call `FindDays()` and print the number of days between X and Y

#	INPUT	OUTPUT
1	X = 23 11 2022 Y = 24 11 2022	X = 23/11/2022 Y = 24/11/2022 Days = 1
2	X = 23 11 2020 Y = 24 11 2022	X = 23/11/2020 Y = 24/11/2022 Days = 731
3	X = 1 1 2000 Y = 31 12 2022	X = 01/01/2000 Y = 31/12/2022 Days = 8400
4	X = 31 1 2020 Y = 1 1 2020	X = 31/01/2020 Y = 01/01/2020 Days = 30

[Time: 40 Minutes] [20]

---*---