

## WEEK2 - RECORD

### Program 1

**AIM:** Write a C program to compute the distance between two points whose coordinates are entered by the user

**Scenario:**

A coordinate-based application asks the user to enter the coordinates of two points  $(x_1, y_1)$  and  $(x_2, y_2)$ . The program calculates the distance between the two points using the distance formula.

**Formula**

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

**Note:** `sqrt()` is a function from the `math.h` library used to calculate the square root.

**Algorithm: Distance Between Two Points**

Step 1: Start.

Step 2: Read the coordinates  $(x_1, y_1)$  of the first point.

Step 3: Read the coordinates  $(x_2, y_2)$  of the second point.

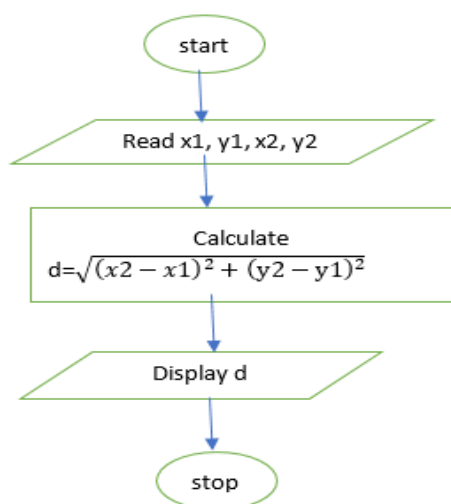
Step 4: Calculate the distance using:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step 5: Display the calculated distance( $d$ ).

Step 6: Stop.

**Flowchart**



### **C Program**

```
#include <stdio.h>
#include <math.h>

int main()
{
    double x1, y1, x2, y2, distance;

    printf("Enter coordinates of first point (x1 y1): ");
    scanf("%lf %lf", &x1, &y1);

    printf("Enter coordinates of second point (x2 y2): ");
    scanf("%lf %lf", &x2, &y2);

    distance = sqrt((x2 - x1) * (x2 - x1) +
                    (y2 - y1) * (y2 - y1));

    printf("Distance between the two points = %.2lf\n", distance);

    return 0;
}
```

### **Sample Input**

Enter coordinates of first point (x1 y1): 2 3

Enter coordinates of second point (x2 y2): 6 6

### **Sample Output**

Distance between the two points = 5.00

## Program 2

**AIM:** Write a C program that reads four integers p, q, r, s and, using relational and logical operators, prints 'Correct values' if  $q > r$ ,  $s > p$  and  $(r + s) > (p + q)$ ; otherwise 'Wrong values'.

### Scenario:

A coordinate-based application asks the user to enter the coordinates of two points  $(x_1, y_1)$  and  $(x_2, y_2)$ . The program calculates the distance between the two points using the distance formula.

### Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

**Note:** `sqrt()` is a function from the `math.h` library used to calculate the square root.

### Algorithm: Check Correct or Wrong Values

Step 1: Start.

Step 2: Read the values of p, q, r, and s.

Step 3: Check the following conditions using relational and logical operators:

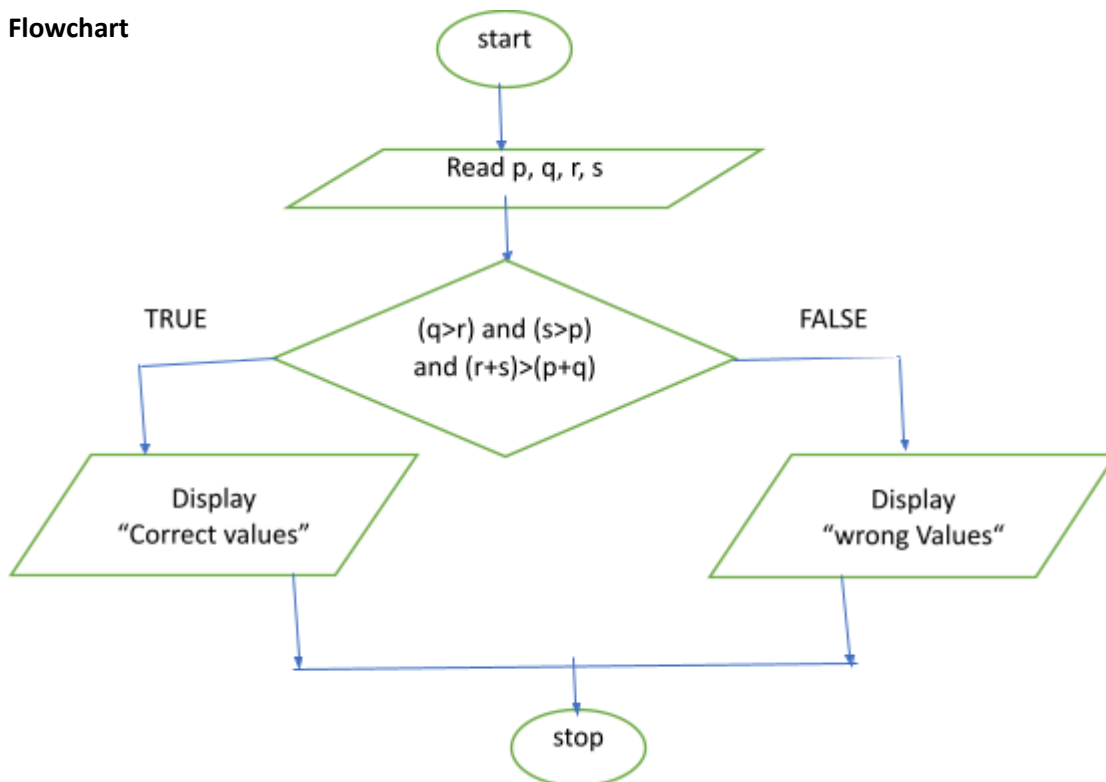
- $q > r$
- $s > p$
- $(r + s) > (p + q)$

Step 4:

- If all three conditions are true, display "Correct values".
- Otherwise, display "Wrong values".

Step 5: Stop.

### Flowchart



### C Program

```
#include <stdio.h>
int main()
{
    int p, q, r, s;

    printf("Enter four integers: ");
    scanf("%d %d %d %d", &p, &q, &r, &s);

    (q > r && s > p && (r + s) > (p + q))
        ? printf("Correct values")
        : printf("Wrong values");

    return 0;
}
```

### Sample Input & output

```
Enter four integers: 1 2 3 4
Wrong values
```

```
Enter four integers: 5 10 6 12
Correct values
```

### Program 3

**AIM:** Write a C program to compute simple and compound interest for values entered by the user.

#### Formula

Simple Interest:

$$SI = \frac{P \times R \times T}{100}$$

Compound Interest:

$$CI = P \left( 1 + \frac{R}{100} \right)^T - P$$

where:

- P = Principal amount
- R = Rate of interest
- T = Time in years

#### Algorithm: Simple and Compound Interest

Step 1: Start

Step 2: Read the **Principal Amount P**, **Rate of interest R**, **Time in years T**.

Step 3: Calculate Simple Interest:

$$SI = (P \times R \times T) / 100$$

compound amount:

$$\text{Amount} = P \times (1 + R/100)^T$$

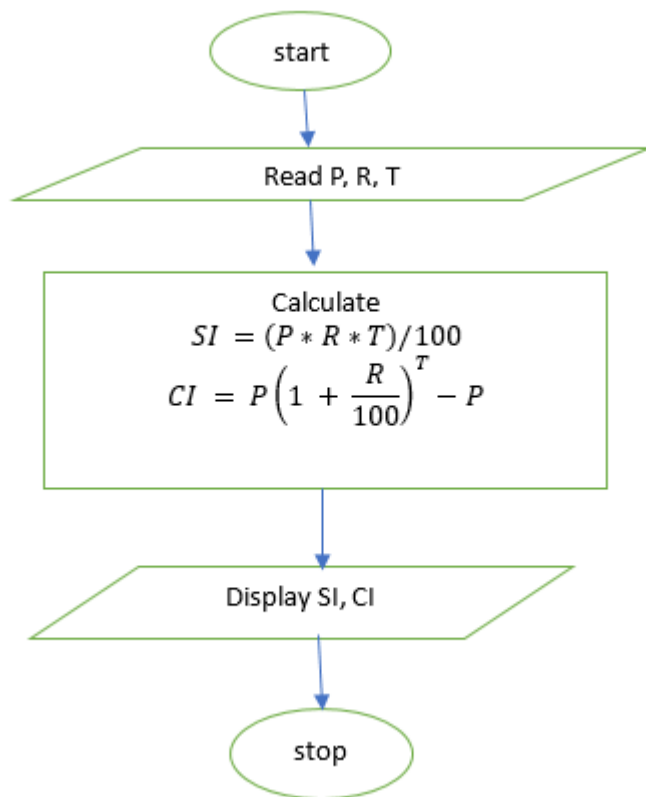
Compound Interest:

$$CI = \text{Amount} - P$$

Step 4: Display the **Simple Interest SI**, **Compound Interest CI**.

Step 5: Stop.

## Flowchart



## C Program

```
#include <stdio.h>
#include <math.h>
int main()
{
    double P, R, T;
    double SI, CI, Amount;

    printf("Enter Principal amount: ");
    scanf("%lf", &P);
    printf("Enter Rate of interest: ");
    scanf("%lf", &R);
    printf("Enter Time in years: ");
    scanf("%lf", &T);
    SI = (P * R * T) / 100;
    Amount = P * pow((1 + R / 100), T);
    CI = Amount - P;
    printf("\nSimple Interest = %.2lf", SI);
    printf("\nCompound Interest = %.2lf", CI);
    return 0;
}
```

}

#### Sample Input & output

```
Enter Principal amount: 10000
Enter Rate of interest: 5
Enter Time in years: 2

Simple Interest = 1000.00
Compound Interest = 1025.00
```