

# Bangladesh University of Business & Technology (BUBT)



## Lab Report

**Course Title** : Structured Programming Language Lab  
**Course Code** : CSE 102

### Submitted by

**Name** : Makhzum-Bin-Harun  
**ID** : 20254103279  
**Intake** : 56  
**Section** : 07  
**Program**: B.Sc. Engg. in CSE

### Submitted to

**Name** : Sourav Kundu  
**Designation**: Lecturer  
**Department of Computer  
Science & Engineering  
Bangladesh University of  
Business & Technology.**

**Date of Submission:**

**Signature of Teacher**

**Problem 01.** Write a C program to input number of days from user and convert it to years, weeks and days.  
Or, how to convert days to years, weeks in C programming.  
Or, Logic to convert days to years, weeks and days in C program.

**01.1. Algorithm:** Algorithm to Convert Days into Years, Weeks, and Days

1. Start
2. Input: Read the total number of days from the user and store it in total\_days.
3. Calculate Years:
  - Set years = total\_days divided by 365 (integer division).
4. Calculate Remaining Days After Years:
  - Set remaining\_days = total\_days modulo 365.
5. Calculate Weeks:
  - Set weeks = remaining\_days divided by 7 (integer division).
6. Calculate Remaining Days After Weeks:
  - Set days = remaining\_days modulo 7.
7. Output: Display the values of years, weeks, and days.
8. Stop

**01.2. C Program Code:**

```
#include <stdio.h>

int main () {
    int total_days, years, weeks, days, remaining_days;

    // Input number of days from user
    printf("Enter the number of days: ");
    scanf("%d", &total_days);

    // Calculate years
    years = total_days/365;

    // Calculate remaining days after extracting years
    remaining_days = total_days % 365;

    // Calculate weeks from remaining days
    weeks = remaining_days / 7;

    // Calculate remaining days after extracting weeks
    days = remaining_days % 7;

    // Display the result
    printf("%d days = %d years, %d weeks, and %d days\n", total_days, years,
    weeks, days);

    return 0;
}
```

}

**01.3. Output:**

**Output**

Clear

```
Enter the number of days: 7264
7264 days = 19 years, 47 weeks, and 0 days

=== Code Execution Successful ===
```

**01.4. Discussion:**

The problem involves converting a total number of days into a more meaningful representation involving years, weeks, and leftover days. This is a common task when dealing with time durations, where expressing time in larger units (years and weeks) is easier to interpret than just a large count of days.

**Key Points:**

- **Year Length Assumption:**  
The program assumes that one year has exactly 365 days. This is a simplification because actual years can be 365 or 366 days (leap years). However, for many general purposes, this assumption is sufficient.
- **Division and Modulus Operations:**  
The main mathematical operations used here are integer division and modulus. Integer division (/) helps find how many complete units (years or weeks) fit into the total days, while modulus (%) gives the remainder or leftover days after extracting those complete units.
- **Stepwise Breakdown:**  
First, the total days are divided by 365 to find the number of complete years. The remainder from this division represents leftover days that do not complete a full year. Next, this remainder is divided by 7 to calculate how many full weeks fit into these leftover days. Finally, the modulus operation finds the days that do not complete a full week.
- **Practical Use:**  
This conversion can be useful in various real-world applications such as age calculations, project timelines, or converting durations into human-friendly formats.
- **Possible Enhancements:**  
The program can be improved by considering leap years or by accepting different lengths for years (e.g., accounting for months). Additionally, input validation can be added to handle negative or invalid inputs.

**Problem 02.** Write a C program to input radius of a circle from user and find diameter, circumference and area of the circle.

Or, how to calculate diameter, circumference and area of a circle whose radius is given by user in C programming.

Or, Logic to find diameter, circumference and area of a circle in C.

**02.1. Algorithm:** Find Diameter, Circumference, and Area of a Circle

1. Start
2. Input the radius of the circle from the user.
3. Calculate the diameter using the formula:

$$\text{diameter} = 2 \times \text{radius}$$

4. Calculate the circumference using the formula:

$$\text{circumference} = 2 \times \pi \times \text{radius}$$

5. Calculate the area using the formula:

$$\text{area} = \pi \times \text{radius}^2$$

6. Display the diameter, circumference, and area.
7. Stop

**02.2. C Program Code:**

```
#include <stdio.h>

#define PI 3.1416 // Constant value of π

int main() {
    float radius, diameter, circumference, area;

    // Input radius from user
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    // Calculations
    diameter = 2 * radius;
    circumference = 2 * PI * radius;
    area = PI * radius * radius;

    // Output results
    printf("Diameter: %.2f\n", diameter);
    printf("Circumference: %.2f\n", circumference);
    printf("Area: %.2f\n", area);
}
```

```
    return 0;  
}
```

### 02. 3. Sample Input/Output

Input: 5

Output:

Output

Clear

```
Enter the radius of the circle: 5  
Diameter: 10.00  
Circumference: 31.42  
Area: 78.54  
  
=== Code Execution Successful ===
```

### 02.4. Discussion:

This program uses the radius entered by the user to compute the diameter, circumference, and area of a circle. The constant value of  $\pi$  (pi) is defined using `#define` to ensure accuracy and ease of modification. Floating-point variables are used to allow decimal precision.

The program flow follows a clear sequence: input  $\rightarrow$  calculations  $\rightarrow$  output. All formulas are derived from basic geometry. The use of `printf` with formatting specifiers ensures that results are displayed with two decimal places for better readability.

This experiment demonstrates:

- Taking user input with `scanf`
- Performing arithmetic calculations in C
- Defining and using constants with `#define`
- Output formatting with `printf`

Problem 03. Write a C program to input temperature in Centigrade and convert to Fahrenheit. How to convert temperature from degree centigrade to degree Fahrenheit in C programming. Logic to convert temperature from Celsius to Fahrenheit in C.

Problem 04. Write a C program to input length in centimeter and convert it to meter and kilometer. How to convert length from centimeter to meter and kilometer in C programming. Length conversion program in C from centimeter to meter and centimeter to kilometer.

Problem 05. Write a C program to input principle, time and rate (P, T, R) from user and find Simple Interest. How to calculate simple interest in C programming. Logic to find simple interest in C program.