Easy (10 programs)

1. Positive or Negative
   - Read an integer; print whether it is positive, negative or zero.

2. Largest of Two Numbers
   - Read two integers; print the largest one.

3. Sum of First N Natural Numbers
   - Read $$N$$; print sum $$1+2+…+N$$ using loop.

4. Even or Odd using Modulus
   - Read integer; use `if` with `% 2` to decide even/odd.

5. Simple Calculator (Switch)
   - Read two numbers and an operator (+, -, , /); print result.

6. Table of a Number
   - Read integer; print its multiplication table up to 10.

7. Factorial of a Number (Iterative)
   - Read non-negative integer; compute factorial using `for` loop.

8. Sum of Digits
   - Read integer; find sum of its digits using `%` and `/`.

9. Reverse of a Number
   - Read integer; print reverse (e.g., 123 → 321).

10. Count Digits in a Number
    - Read integer; print how many digits it contains.

Medium (15 programs)

11. Largest of Three Numbers
12. Check Leap Year
13. Check Prime or Not
14. Print All Primes in a Range
15. Fibonacci Series up to N Terms
16. Palindrome Number Check

17. Armstrong Number Check (3-digit)
18. Menu Driven: Odd/Even, Positive/Negative, Prime/Not
19. Array – Sum and Average of N Numbers
20. Array – Largest and Smallest Element
21. Array – Linear Search
22. Array – Sorting (Bubble Sort)
23. String – Length without `strlen`
24. String – Count Vowels and Consonants
25. String – Check Palindrome

Hard (10 programs)

26. Matrix Addition (2D Array)
27. Matrix Multiplication (2D Array)
28. Transpose of Matrix
29. Recursive Factorial
30. Recursive Fibonacci
31. Pointer – Swap Two Numbers using Function and Pointers
32. Structure – Student Details (name, roll, 3 marks; compute total & average)
33. Array of Structures – Find Topper
34. File Handling – Copy one text file to another
35. File Handling – Count number of characters, words, lines in a file

---

1) Positive / Negative / Zero

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);

    if (n > 0)
        printf("Positive\n");
    else if (n < 0)
        printf("Negative\n");
    else
```

```
    printf("Zero\n");

  return 0;
}
```

Sample Output

```
Enter a number: 5
Positive
```

***

2) Largest of Two Numbers

```c
#include <stdio.h>

int main() {
  int a, b;
  printf("Enter two numbers: ");
  scanf("%d %d", &a, &b);

  if (a > b)
    printf("%d is larger\n", a);
  else if (b > a)
    printf("%d is larger\n", b);
  else
    printf("Both are equal\n");

  return 0;
}
```

Sample Output

```
Enter two numbers: 10 20
20 is larger
```

***

3) Sum of First N Natural Numbers

```c
#include <stdio.h>

int main() {
    int n, i, sum = 0;
    printf("Enter N: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        sum = sum + i;
    }

    printf("Sum = %d\n", sum);
    return 0;
}
```

Sample Output
```
Enter N: 5
Sum = 15
```

***

4) Even or Odd

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
```

```
    if (n % 2 == 0)
        printf("Even\n");
    else
        printf("Odd\n");

    return 0;
}
```

Sample Output

```
Enter a number: 7
Odd
```

***

5) Simple Calculator (switch)

```c
#include <stdio.h>

int main() {
    float a, b;
    char op;

    printf("Enter expression (a op b): ");
    scanf("%f %c %f", &a, &op, &b);

    switch (op) {
        case '+': printf("Result = %.2f\n", a + b); break;
        case '-': printf("Result = %.2f\n", a - b); break;
        case '*': printf("Result = %.2f\n", a * b); break;
        case '/':
            if (b != 0)
                printf("Result = %.2f\n", a / b);
            else
                printf("Division by zero not allowed\n");
            break;
        default: printf("Invalid operator\n");
```

```
    }
    return 0;
}
```

Sample Output

```
Enter expression (a op b): 5 + 3
Result = 8.00
```

***

6) Multiplication Table of a Number

```c
#include <stdio.h>

int main() {
    int n, i;
    printf("Enter a number: ");
    scanf("%d", &n);

    for (i = 1; i <= 10; i++) {
        printf("%d x %d = %d\n", n, i, n * i);
    }
    return 0;
}
```

Sample Output

```
Enter a number: 4
4 x 1 = 4
...
4 x 10 = 40
```

***

7) Factorial (Iterative)

```c
#include <stdio.h>

int main() {
    int n, i;
    long long fact = 1;

    printf("Enter a non-negative integer: ");
    scanf("%d", &n);

    if (n < 0) {
        printf("Factorial not defined for negative numbers\n");
        return 0;
    }

    for (i = 1; i <= n; i++) {
        fact = fact * i;
    }

    printf("Factorial = %lld\n", fact);
    return 0;
}
```

Sample Output

```
Enter a non-negative integer: 5
Factorial = 120
```

***

8) Sum of Digits of a Number

```c
#include <stdio.h>

int main() {
```

```
    int n, rem, sum = 0;
    printf("Enter an integer: ");
    scanf("%d", &n);

    while (n != 0) {
        rem = n % 10;
        sum = sum + rem;
        n = n / 10;
    }

    printf("Sum of digits = %d\n", sum);
    return 0;
}
```

Sample Output

```
Enter an integer: 1234
Sum of digits = 10
```

***

9) Reverse of a Number

```c
#include <stdio.h>

int main() {
    int n, rem, rev = 0;
    printf("Enter an integer: ");
    scanf("%d", &n);

    while (n != 0) {
        rem = n % 10;
        rev = rev * 10 + rem;
        n = n / 10;
    }

    printf("Reversed number = %d\n", rev);
```

```
    return 0;
}
```

Sample Output

```
Enter an integer: 123
Reversed number = 321
```

***

10) Count Digits in a Number

```c
#include <stdio.h>

int main() {
    int n, count = 0;

    printf("Enter an integer: ");
    scanf("%d", &n);

    if (n == 0) {
        count = 1;
    } else {
        if (n < 0)
            n = -n;   /* ignore sign */

        while (n != 0) {
            n = n / 10;
            count++;
        }
    }

    printf("Number of digits = %d\n", count);
    return 0;
}
```

Sample Output
```
Enter an integer: 2025
Number of digits = 4
```

Medium Level Programs (11–25)

11) Largest of Three Numbers

```c
#include <stdio.h>

int main() {
    int a, b, c;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a >= b && a >= c)
        printf("%d is largest\n", a);
    else if (b >= a && b >= c)
        printf("%d is largest\n", b);
    else
        printf("%d is largest\n", c);

    return 0;
}
```

Sample:
```
Enter three numbers: 4 9 2
9 is largest
```

***

12) Leap Year Check

```c
#include <stdio.h>

int main() {
    int year;
    printf("Enter year: ");
    scanf("%d", &year);

    if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))
        printf("Leap year\n");
    else
        printf("Not a leap year\n");

    return 0;
}
```

Sample:

```
Enter year: 2024
Leap year
```

***

13) Prime or Not

```c
#include <stdio.h>

int main() {
    int n, i, flag = 0;
    printf("Enter a number: ");
    scanf("%d", &n);

    if (n <= 1) {
        printf("Not prime\n");
        return 0;
    }
```

```
    for (i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if (flag == 0)
        printf("Prime\n");
    else
        printf("Not prime\n");

    return 0;
}
```

Sample:

```
Enter a number: 13
Prime
```

***

14) All Primes in a Range

```c
#include <stdio.h>

int main() {
    int low, high, i, j, flag;
    printf("Enter low and high: ");
    scanf("%d %d", &low, &high);

    for (i = low; i <= high; i++) {
        if (i <= 1) continue;
        flag = 0;
        for (j = 2; j * j <= i; j++) {
            if (i % j == 0) {
                flag = 1;
```

```
            break;
        }
    }
    if (flag == 0)
        printf("%d ", i);
  }
  printf("\n");
  return 0;
}
```

Sample:

```
Enter low and high: 10 20
11 13 17 19
```

***

15) Fibonacci Series up to N Terms

```c
#include <stdio.h>

int main() {
  int n, i;
  int a = 0, b = 1, c;
  printf("Enter number of terms: ");
  scanf("%d", &n);

  if (n >= 1) printf("%d ", a);
  if (n >= 2) printf("%d ", b);

  for (i = 3; i <= n; i++) {
    c = a + b;
    printf("%d ", c);
    a = b;
    b = c;
  }
  printf("\n");
```

```
    return 0;
}
```

Sample:
```
Enter number of terms: 6
0 1 1 2 3 5
```

***

16) Palindrome Number

```c
#include <stdio.h>

int main() {
    int n, temp, rem, rev = 0;
    printf("Enter an integer: ");
    scanf("%d", &n);

    temp = n;
    while (temp != 0) {
        rem = temp % 10;
        rev = rev * 10 + rem;
        temp = temp / 10;
    }

    if (rev == n)
        printf("Palindrome\n");
    else
        printf("Not palindrome\n");

    return 0;
}
```

Sample:
```
```

```
Enter an integer: 121
Palindrome
```

***

17) Armstrong Number (3-digit)

```c
#include <stdio.h>

int main() {
    int n, temp, rem, sum = 0;
    printf("Enter a 3-digit number: ");
    scanf("%d", &n);

    temp = n;
    while (temp != 0) {
        rem = temp % 10;
        sum = sum + rem * rem * rem;
        temp = temp / 10;
    }

    if (sum == n)
        printf("Armstrong number\n");
    else
        printf("Not Armstrong number\n");

    return 0;
}
```

Sample:
```
Enter a 3-digit number: 153
Armstrong number
```

***

18) Menu Driven (Odd/Even, +/−, Prime)

```c
#include <stdio.h>

int isPrime(int n) {
    int i;
    if (n <= 1) return 0;
    for (i = 2; i * i <= n; i++)
        if (n % i == 0) return 0;
    return 1;
}

int main() {
    int choice, n;
    printf("1. Odd or Even\n2. Positive or Negative\n3. Prime or Not\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    printf("Enter a number: ");
    scanf("%d", &n);

    switch (choice) {
        case 1:
            if (n % 2 == 0) printf("Even\n");
            else printf("Odd\n");
            break;
        case 2:
            if (n > 0) printf("Positive\n");
            else if (n < 0) printf("Negative\n");
            else printf("Zero\n");
            break;
        case 3:
            if (isPrime(n)) printf("Prime\n");
            else printf("Not prime\n");
            break;
        default:
            printf("Invalid choice\n");
    }
    return 0;
```

```
}
```

Sample:

```
1. Odd or Even
2. Positive or Negative
3. Prime or Not
Enter your choice: 3
Enter a number: 11
Prime
```

***

19) Array – Sum and Average

```c
#include <stdio.h>

int main() {
    int n, i;
    int a[100];
    int sum = 0;
    float avg;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
        sum += a[i];
    }

    avg = (float)sum / n;
    printf("Sum = %d\nAverage = %.2f\n", sum, avg);
    return 0;
}
```

Sample:
```
Enter number of elements: 4
Enter 4 integers:
1 2 3 4
Sum = 10
Average = 2.50
```

***

20) Array – Largest and Smallest

```c
#include <stdio.h>

int main() {
    int n, i;
    int a[100], max, min;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    max = min = a[0];

    for (i = 1; i < n; i++) {
        if (a[i] > max) max = a[i];
        if (a[i] < min) min = a[i];
    }

    printf("Max = %d\nMin = %d\n", max, min);
    return 0;
}
```

Sample:

```
Enter number of elements: 5
Enter 5 integers:
3 9 1 7 4
Max = 9
Min = 1
```

***

21) Linear Search

```c
#include <stdio.h>

int main() {
    int n, i, key, pos = -1;
    int a[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    printf("Enter element to search: ");
    scanf("%d", &key);

    for (i = 0; i < n; i++) {
        if (a[i] == key) {
            pos = i;
            break;
        }
    }

    if (pos == -1)
        printf("Element not found\n");
    else
```

```
        printf("Element found at index %d\n", pos);

    return 0;
}
```

Sample:

```
Enter number of elements: 5
Enter 5 integers:
2 5 7 1 9
Enter element to search: 7
Element found at index 2
```

***

22) Bubble Sort

```c
#include <stdio.h>

int main() {
    int n, i, j, temp;
    int a[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - 1 - i; j++) {
            if (a[j] > a[j + 1]) {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
```

```
        }
    }

    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

Sample:

```
Enter number of elements: 5
Enter 5 integers:
5 1 4 2 3
Sorted array:
1 2 3 4 5
```

***

23) String Length without `strlen`

```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0, len = 0;

    printf("Enter a string: ");
    gets(str);   /* for lab use only */

    while (str[i] != '\0') {
        len++;
        i++;
    }

    printf("Length = %d\n", len);
```

```
    return 0;
}
```

Sample:

```
Enter a string: hello
Length = 5
```

***

24) Count Vowels and Consonants

```c
#include <stdio.h>

int main() {
    char str[100];
    int i, vowels = 0, consonants = 0;

    printf("Enter a string: ");
    gets(str);

    for (i = 0; str[i] != '\0'; i++) {
        char ch = str[i];
        if (ch >= 'A' && ch <= 'Z')
            ch = ch + 32;

        if (ch >= 'a' && ch <= 'z') {
            if (ch == 'a' || ch == 'e' || ch == 'i' ||
                ch == 'o' || ch == 'u')
                vowels++;
            else
                consonants++;
        }
    }

    printf("Vowels = %d\nConsonants = %d\n", vowels, consonants);
    return 0;
```

}
```

Sample:
```
Enter a string: hello world
Vowels = 3
Consonants = 7
```

***

### 25) String Palindrome Check

```c
#include <stdio.h>

int main() {
    char str[100];
    int i, j, flag = 0;

    printf("Enter a string: ");
    gets(str);

    for (i = 0; str[i] != '\0'; i++);
    j = i - 1;

    for (i = 0; i < j; i++, j--) {
        if (str[i] != str[j]) {
            flag = 1;
            break;
        }
    }

    if (flag == 0)
        printf("Palindrome string\n");
    else
        printf("Not palindrome string\n");

    return 0;

}
```

Sample:
```
Enter a string: level
Palindrome string
```

***

Hard Level Programs (26–35)

26) Matrix Addition

```c
#include <stdio.h>

int main() {
    int a[10][10], b[10][10], c[10][10];
    int r, col, i, j;

    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &col);

    printf("Enter elements of first matrix:\n");
    for (i = 0; i < r; i++)
        for (j = 0; j < col; j++)
            scanf("%d", &a[i][j]);

    printf("Enter elements of second matrix:\n");
    for (i = 0; i < r; i++)
        for (j = 0; j < col; j++)
            scanf("%d", &b[i][j]);

    for (i = 0; i < r; i++)
        for (j = 0; j < col; j++)
            c[i][j] = a[i][j] + b[i][j];

    printf("Resultant matrix:\n");
```
```

```
    for (i = 0; i < r; i++) {
        for (j = 0; j < col; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }
    return 0;
}
```

***

27) Matrix Multiplication

```c
#include <stdio.h>

int main() {
    int a[10][10], b[10][10], c[10][10];
    int r1, c1, r2, c2, i, j, k;

    printf("Enter rows and columns of first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and columns of second matrix: ");
    scanf("%d %d", &r2, &c2);

    if (c1 != r2) {
        printf("Multiplication not possible\n");
        return 0;
    }

    printf("Enter elements of first matrix:\n");
    for (i = 0; i < r1; i++)
        for (j = 0; j < c1; j++)
            scanf("%d", &a[i][j]);

    printf("Enter elements of second matrix:\n");
    for (i = 0; i < r2; i++)
        for (j = 0; j < c2; j++)
            scanf("%d", &b[i][j]);
```

```
    for (i = 0; i < r1; i++)
        for (j = 0; j < c2; j++) {
            c[i][j] = 0;
            for (k = 0; k < c1; k++)
                c[i][j] += a[i][k] * b[k][j];
        }

    printf("Product matrix:\n");
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }
    return 0;
}
```

***

28) Transpose of Matrix

```c
#include <stdio.h>

int main() {
    int a[10][10], t[10][10];
    int r, c, i, j;

    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);

    printf("Enter elements:\n");
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
            scanf("%d", &a[i][j]);

    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
            t[j][i] = a[i][j];
```

```
    printf("Transpose:\n");
    for (i = 0; i < c; i++) {
        for (j = 0; j < r; j++)
            printf("%d ", t[i][j]);
        printf("\n");
    }
    return 0;
}
```

***

29) Recursive Factorial

```c
#include <stdio.h>

long long fact(int n) {
    if (n == 0 || n == 1) return 1;
    return n * fact(n - 1);
}

int main() {
    int n;
    printf("Enter a non-negative integer: ");
    scanf("%d", &n);

    if (n < 0)
        printf("Not defined\n");
    else
        printf("Factorial = %lld\n", fact(n));

    return 0;
}
```

***

30) Recursive Fibonacci

```c
#include <stdio.h>

int fib(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;
    return fib(n - 1) + fib(n - 2);
}

int main() {
    int n, i;
    printf("Enter number of terms: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++)
        printf("%d ", fib(i));
    printf("\n");
    return 0;
}
```

***

31) Swap Using Pointers

```c
#include <stdio.h>

void swap(int *x, int *y) {
    int temp = *x;
    *x = *y;
    *y = temp;
}

int main() {
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    swap(&a, &b);
```

```
    printf("After swap: a = %d, b = %d\n", a, b);
    return 0;
}
```

***

32) Structure – Single Student

```c
#include <stdio.h>

struct Student {
    char name[30];
    int roll;
    int m1, m2, m3;
    int total;
    float avg;
};

int main() {
    struct Student s;

    printf("Enter name: ");
    gets(s.name);
    printf("Enter roll: ");
    scanf("%d", &s.roll);
    printf("Enter 3 subject marks: ");
    scanf("%d %d %d", &s.m1, &s.m2, &s.m3);

    s.total = s.m1 + s.m2 + s.m3;
    s.avg = s.total / 3.0;

    printf("Total = %d\nAverage = %.2f\n", s.total, s.avg);
    return 0;
}
```

***

33) Array of Structures – Topper

```c
#include <stdio.h>

struct Student {
    char name[30];
    int roll;
    int total;
};

int main() {
    struct Student s[50];
    int n, i, topIndex = 0;

    printf("Enter number of students: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Enter name: ");
        scanf("%s", s[i].name);
        printf("Enter roll: ");
        scanf("%d", &s[i].roll);
        printf("Enter total marks: ");
        scanf("%d", &s[i].total);
    }

    for (i = 1; i < n; i++) {
        if (s[i].total > s[topIndex].total)
            topIndex = i;
    }

    printf("Topper: %s (Roll %d) with %d marks\n",
           s[topIndex].name, s[topIndex].roll, s[topIndex].total);
    return 0;
}
```

***

34) File Copy

```c
#include <stdio.h>

int main() {
    FILE *fs, *ft;
    char ch;

    fs = fopen("source.txt", "r");
    ft = fopen("target.txt", "w");

    if (fs == NULL || ft == NULL) {
        printf("File error\n");
        return 0;
    }

    while ((ch = fgetc(fs)) != EOF) {
        fputc(ch, ft);
    }

    printf("File copied successfully\n");

    fclose(fs);
    fclose(ft);
    return 0;
}
```

***

35) Count Characters, Words, Lines in File

```c
#include <stdio.h>

int main() {
    FILE *fp;
    char ch;
```

```
    int chars = 0, words = 0, lines = 0;
    int inWord = 0;

    fp = fopen("input.txt", "r");
    if (fp == NULL) {
        printf("Cannot open file\n");
        return 0;
    }

    while ((ch = fgetc(fp)) != EOF) {
        chars++;

        if (ch == '\n')
            lines++;

        if (ch == ' ' || ch == '\n' || ch == '\t') {
            if (inWord) {
                words++;
                inWord = 0;
            }
        } else {
            inWord = 1;
        }
    }

    if (inWord)
        words++;

    printf("Characters = %d\nWords = %d\nLines = %d\n",
           chars, words, lines);

    fclose(fp);
    return 0;
}
```

1) Decimal to Binary

```c
#include <stdio.h>

int main() {
    int decimal, rem, i = 0;
    int bin[32];

    printf("Enter a decimal number: ");
    scanf("%d", &decimal);

    if (decimal == 0) {
        printf("Binary = 0\n");
        return 0;
    }

    while (decimal > 0) {
        rem = decimal % 2;
        bin[i] = rem;
        i++;
        decimal = decimal / 2;
    }

    printf("Binary = ");
    while (i > 0) {
        i--;
        printf("%d", bin[i]);
    }
    printf("\n");

    return 0;
}
```

Sample:
`Input: 10 → Output: Binary = 1010`

***

2) Binary to Decimal

```c
#include <stdio.h>

int main() {
    long long bin;
    int dec = 0, base = 1, rem;

    printf("Enter a binary number: ");
    scanf("%lld", &bin);

    while (bin > 0) {
        rem = bin % 10;
        dec = dec + rem * base;
        base = base * 2;
        bin = bin / 10;
    }

    printf("Decimal = %d\n", dec);
    return 0;
}
```

Sample:
`Input: 1011 → Output: Decimal = 11` [1]

***

3) Star Pattern – Right Half Pyramid

```c
#include <stdio.h>

int main() {
    int rows, i, j;
    printf("Enter number of rows: ");
    scanf("%d", &rows);
```

```
    for (i = 1; i <= rows; i++) {
        for (j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Sample (rows=4):

```
*
* *
* * *
* * * *
```

***

4) Number Pattern – Floyd's Triangle

```c
#include <stdio.h>

int main() {
    int rows, i, j, num = 1;
    printf("Enter number of rows: ");
    scanf("%d", &rows);

    for (i = 1; i <= rows; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d ", num);
            num++;
        }
        printf("\n");
    }
    return 0;
}
```

```

Sample (rows=4):
```
1
2 3
4 5 6
7 8 9 10
```

***

## 5) Simple Interest

```c
#include <stdio.h>

int main() {
    float P, R, T, SI;

    printf("Enter principal: ");
    scanf("%f", &P);
    printf("Enter rate: ");
    scanf("%f", &R);
    printf("Enter time: ");
    scanf("%f", &T);

    SI = (P * R * T) / 100.0;

    printf("Simple Interest = %.2f\n", SI);
    return 0;
}
```

Sample:
`P=1000, R=5, T=2 → Simple Interest = 100.00`

****
```