

## Assignment 3

### 1. Equality Check:

Write a program to check if two integers provided by the user are equal or not.

Sol: #include <stdio.h>

```
    int main() {  
    int num1, num2;  
    printf("Enter the first integer: ");  
    scanf("%d", &num1);  
    printf("Enter the second integer: ");  
    scanf("%d", &num2);  
    if (num1 == num2) {  
        printf("The numbers are equal.\n");  
    } else {  
        printf("The numbers are not equal.\n");  
    }  
    return 0;  
}
```

O/p:

Enter the first integer: 67

Enter the second integer: 85

The numbers are not equal.

## 2. Greater Number Identification:

Write a program to determine which of two numbers is greater using relational operators.

Sol: #include <stdio.h>

```
    int main() {  
        int num1, num2  
        printf("Enter the first number: ");  
        scanf("%d", &num1);  
        printf("Enter the second number: ");  
        scanf("%d", &num2);  
        if (num1 > num2) {  
            printf("%d is greater than %d.\n", num1, num2);  
        } else if (num1 < num2) {  
            printf("%d is greater than %d.\n", num2, num1);  
        } else {  
            printf("Both numbers are equal.\n");  
        }  
        return 0;  
    }
```

O/p: Enter the first number: 76

Enter the second number: 90

90 is greater than 76.

## 3. Check if a Number is Positive:

Use relational operators to check if a given number is positive (greater than 0).

Sol: #include <stdio.h>

```
    int main() {  
    int number;  
    printf("Enter a number: ");  
    scanf("%d", &number);  
    if (number > 0) {  
        printf("The number %d is positive.\n", number);  
    } else if (number == 0) {  
        printf("The number is zero.\n");  
    } else {  
        printf("The number %d is not positive.\n", number);  
    }  
    return 0;  
}
```

O/p : Enter a number: 9

The number 9 is positive.

Enter a number: 0

The number is zero.

Enter a number: -5

The number -5 is not positive.

4. Rectangle Validity Check:

Write a program to verify if the given length and breadth of a rectangle satisfy the condition of a valid rectangle (length > 0 and breadth > 0).

Sol: #include <stdio.h>

```
int main() {  
    float length, breadth;  
    printf("Enter the length of the rectangle: ");  
    scanf("%f", &length);  
    printf("Enter the breadth of the rectangle: ");  
    scanf("%f", &breadth);  
    if (length > 0 && breadth > 0) {  
        printf("The dimensions (length = %.2f, breadth = %.2f) form a  
valid rectangle.\n", length, breadth);  
    } else {  
        printf("Invalid dimensions! Length and breadth must be greater  
than 0.\n");  
    }  
    return 0;  
}
```

O/p: Enter the length of the rectangle: 35

Enter the breadth of the rectangle: 68

The dimensions (length = 35.00, breadth = 68.00) form a valid rectangle.

## 5. Grade Eligibility Check:

Given a student's marks in a subject, determine if the student has passed (marks >= 40).

Sol: #include <stdio.h>

```
int main() {  
float marks;  
printf("Enter the marks obtained by the student: ");  
scanf("%f", &marks);  
if(marks>40)  
    printf("The student has passed with %.2f marks.\n", marks);  
else {  
    printf("The student has failed with %.2f marks.\n", marks);  
}  
return 0;  
}
```

O/p:

Enter the marks obtained by the student: 89

The student has passed with 89.00 marks.

Enter the marks obtained by the student: 25

The student has failed with 25.00 marks.

6. Check if Number is Within Range:

Use relational operators to check if a given number lies between 10 and 50 (inclusive).

Sol: #include <stdio.h>

```
int main() {  
    int number;
```

```

printf("Enter a number: ");
scanf("%d", &number);
if (number >= 10 && number <= 50) {
    printf("The number %d is within the range of 10 to 50
(inclusive).\n", number);
} else {
    printf("The number %d is outside the range of 10 to 50.\n",
number);
}

return 0;
}

```

o/p : Enter a number: 30

The number 30 is within the range of 10 to 50 (inclusive).

Enter a number: 79

The number 79 is outside the range of 10 to 50.

## 7. Verify Alphabetic Range:

Write a program to check if a given character is a lowercase English letter (between 'a' and 'z').

Sol: #include <stdio.h>

```

int main() {
    char character;
    printf("Enter a character: ");
    scanf("%c", &character);
}

```

```
if (character >= 'a' && character <= 'z') {  
    printf("The character '%c' is a lowercase English letter.\n",  
character);  
} else {  
    printf("The character '%c' is not a lowercase English letter.\n",  
character);  
}  
return 0;  
}
```

o/p: Enter a character: L

The character 'L' is not a lowercase English letter.

Enter a character: s

The character 's' is a lowercase English letter.

## 8. Age Comparison:

Compare the ages of two people and determine who is older or if both are of the same age.

Sol: #include <stdio.h>

```
int main() {  
    int age1, age2;  
    printf("Enter the age of the first person: ");  
    scanf("%d", &age1);  
    printf("Enter the age of the second person: ");  
    scanf("%d", &age2);  
    if (age1 > age2) {
```

```

printf("The first person is older than the second person.\n");
    } else if (age1 < age2) {
        printf("The second person is older than the first person.\n");
    } else {
        printf("Both persons are of the same age.\n");
    }
    return 0;
}

```

o/p : Enter the age of the first person: 18

Enter the age of the second person: 56

The second person is older than the first person.

Enter the age of the first person: 23

Enter the age of the second person: 23

Both persons are of the same age.

## 9. Weight Limit Check:

Write a program to determine if the weight of an object exceeds the specified maximum limit (e.g., 50 kg).

Sol: #include <stdio.h>

```

int main() {
    float weight;
    printf("Enter the weight of the object (in kg): ");
    scanf("%f", &weight);
    if (weight > 50) {

```



```
    printf("The weight exceeds the maximum limit.\n");
} else {
    printf("The weight is within the limit.\n");
}
return 0;
```

O/p: Enter the weight of the object (in kg): 890

The weight exceeds the maximum limit.

Enter the weight of the object (in kg): 35

The weight is within the limit.

#### 10. Rectangle Larger Area Check:

Compare the areas of two rectangles given their lengths and breadths and determine which rectangle has a larger area.

Sol: #include <stdio.h>

```
int main() {
    float length1, breadth1, length2, breadth2, area1, area2;
    printf("Enter the length and breadth of first rectangle: ");
    scanf("%f %f", &length1, &breadth1);
    printf("Enter the length and breadth of second rectangle: ");
    scanf("%f %f", &length2, &breadth2);
    area1 = length1 * breadth1;
    area2 = length2 * breadth2;
    if (area1 > area2) {
        printf("The first rectangle has a larger area.\n");
    }
}
```

```

    } else if (area2 > area1) {
        printf("The second rectangle has a larger area.\n");
    } else {
        printf("Both rectangles have the same area.\n");
    }
    return 0;
}

```

O/p: Enter the length and breadth of first rectangle: 45 30

Enter the length and breadth of second rectangle: 23 78

The second rectangle has a larger area.

Using bitwise operators

11. Write a program to compute the result of the bitwise AND operation between two integers provided by the user.

Sol: #include <stdio.h>

```

int main() {
    int num1, num2, result;
    printf("Enter the first integer: ");
    scanf("%d", &num1);
    printf("Enter the second integer: ");
    scanf("%d", &num2);
    result = num1 & num2;
    printf("The result of %d & %d is: %d\n", num1, num2, result);
    return 0;
}

```

```
}
```

O/p: Enter the first integer: 4

Enter the second integer: 6

The result of 4 & 6 is: 4

12. Write a program to compute the result of the bitwise OR operation between two integers provided by the user.

Sol: #include <stdio.h>

```
int main() {  
    int num1, num2, result;  
    printf("Enter the first integer: ");  
    scanf("%d", &num1);  
    printf("Enter the second integer: ");  
    scanf("%d", &num2);  
    result = num1 | num2;  
    printf("The result of %d | %d is: %d\n", num1, num2, result);  
    return 0;  
}
```

O/p:

Enter the first integer: 12

Enter the second integer: 5

The result of 12 | 5 is: 13

13. Write a program to compute the result of the bitwise XOR operation between two integers provided by the user.

Sol: #include <stdio.h>

```

int main() {
    int num1, num2, result;
    printf("Enter the first integer: ");
    scanf("%d", &num1);
    printf("Enter the second integer: ");
    scanf("%d", &num2);
    result = num1 ^ num2;
    printf("The result of %d ^ %d is: %d\n", num1, num2, result);
    return 0;
}

```

O/p : Enter the first integer: 5

Enter the second integer: 10

The result of 5 ^ 10 is: 15

14. Write a program to find the bitwise complement of a given integer and print the result.

Sol: #include <stdio.h>

```

int main() {
    int num, result;
    printf("Enter an integer: ");
    scanf("%d", &num);
    result = ~num;
    printf("The bitwise complement of %d is: %d\n", num, result);
}

```

```
    return 0;
}
```

O/p: Enter an integer: 12

The bitwise complement of 12 is: -13

15. Given an integer n and a position p, write a program to toggle the bit at position p using the XOR operator.

Sol: #include <stdio.h>

```
int toggleBit(int n, int p) {
    int mask = 1 << p;
    n = n ^ mask;
    return n;
}

int main() {
    int n, p;
    printf("Enter an integer: ");
    scanf("%d", &n);
    printf("Enter the position to toggle: ");
    scanf("%d", &p);
    int result = toggleBit(n, p);
    printf("After toggling bit at position %d, the new value is: %d\n", p,
result);
    return 0;
}
```

O/p: Enter an integer: 21

Enter the position to toggle: 2

After toggling bit at position 2, the new value is: 17

16. Write a program to set the bit at a given position p in an integer n to 1 using the OR operator.

Sol: #include <stdio.h>

```
int setBit(int n, int p) {  
    int mask = 1 << p;  
    n = n | mask;  
    return n;  
}
```

```
int main() {  
    int n, p;  
    printf("Enter an integer: ");  
    scanf("%d", &n);  
    printf("Enter the position to set: ");  
    scanf("%d", &p);  
    int result = setBit(n, p);  
    printf("After setting bit at position %d, the new value is: %d\n", p,  
result);  
  
    return 0;  
}
```

O/p: Enter an integer: 4

Enter the position to set: 1

After setting bit at position 1, the new value is: 6

17. Write a program to clear (set to 0) the bit at a given position p in an integer n using the AND and NOT operators.

Sol: #include <stdio.h>

```
int clearBit(int n, int p) {  
    int mask = ~(1 << p);  
    n = n & mask;  
    return n;  
}  
  
int main() {  
    int n, p;  
    printf("Enter an integer: ");  
    scanf("%d", &n);  
    printf("Enter the position to clear: ");  
    scanf("%d", &p);  
    int result = clearBit(n, p);  
    printf("After clearing bit at position %d, the new value is: %d\n", p,  
result);  
    return 0;  
}
```

O/p: Enter an integer: 12

Enter the position to clear: 2

After clearing bit at position 2, the new value is: 8

## 18. Number Properties Validation:

Write a program to check if a given integer is both a multiple of 5 (arithmetic operator) and greater than 50 (relational operator). Additionally, verify if its binary representation has its least significant bit set (bitwise AND operation).

Sol: #include <stdio.h>

```
int main() {
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);
    int isMultipleOf5 = (number % 5 == 0);
    int isGreaterThan50 = (number > 50);
    int isLSBSet = (number & 1);
    if (isMultipleOf5) {
        printf("The number is a multiple of 5.\n");
    } else {
        printf("The number is not a multiple of 5.\n");
    }

    if (isGreaterThan50) {
        printf("The number is greater than 50.\n");
    } else {
        printf("The number is not greater than 50.\n");
    }
}
```



```

if (isLSBSet) {
    printf("The least significant bit (LSB) is set.\n");
} else {
    printf("The least significant bit (LSB) is not set.\n");
}

if (isMultipleOf5 && isGreaterThan50 && isLSBSet) {
    printf("The number satisfies all conditions.\n");
} else {
    printf("The number does not satisfy all conditions.\n");
}

return 0;
}

```

O/p: Enter an integer: 55

The number is a multiple of 5.

The number is greater than 50.

The least significant bit (LSB) is set.

The number satisfies all conditions.

## 19. Toggle and Evaluate Bit Status:

Given an integer n and a bit position p:

Use bit masking and bitwise XOR to toggle the bit at position p. After toggling, check if the updated number is positive (arithmetic and relational operators) and divisible by 2 (logical operators).

Sol: #include <stdio.h>

```
int main() {  
    int n, p;  
    printf("Enter an integer (n): ");  
    scanf("%d", &n);  
    printf("Enter the bit position to toggle (p): ");  
    scanf("%d", &p);  
    int mask = 1 << p;  
    int toggledNumber = n ^ mask;  
    int isPositive = (toggledNumber > 0);  
    int isDivisibleBy2 = ((toggledNumber % 2) == 0);  
    printf("Original number: %d\n", n);  
    printf("Bit position to toggle: %d\n", p);  
    printf("Updated number after toggling: %d\n", toggledNumber);  
    if (isPositive) {  
        printf("The updated number is positive.\n");  
    } else {  
        printf("The updated number is not positive.\n");  
    }  
    if (isDivisibleBy2) {  
        printf("The updated number is divisible by 2.\n");  
    } else {  
        printf("The updated number is not divisible by 2.\n");  
    }  
}
```

```
}  
  
    return 0;  
}
```

O/p: Enter an integer (n): 24

Enter the bit position to toggle (p): 1

Original number: 24

Bit position to toggle: 1

Updated number after toggling: 26

The updated number is positive.

The updated number is divisible by 2.

20. Determine Voting Eligibility with Criteria:

A person can vote if:

Their age is greater than or equal to 18 (relational operator).

They are a registered citizen, represented by a specific bit set in their ID number (bit masking and bitwise AND).

Write a program to verify these conditions using logical operators.

Sol: #include <stdio.h>

```
int main() {  
    int age, id;  
    printf("Enter age: ");  
    scanf("%d", &age);  
    printf("Enter ID number: ");
```

```

scanf("%d", &id);
if (age >= 18) {
    if (id & (1 << 3)) {
        printf("You are eligible to vote.\n");
    } else {
        printf("You are not a registered citizen.\n");
    }
} else {
    printf("You are not eligible to vote due to age.\n");
}

return 0;
}

```

O/p:

Enter age: 23

Enter ID number: 1

You are not a registered citizen.

21. Set, Clear, and Check Specific Bit:

Write a program to:

Use bit masking and bitwise OR to set a specific bit in a number.

Use bitwise AND and NOT to clear another specific bit.

Check if the resulting number is odd (arithmetic and relational operators) and lies within a range (logical operators).

Sol: #include <stdio.h>

```
int main() {
    int number, setBitPosition, clearBitPosition;
    int lowerRange, upperRange;
    printf("Enter the initial number: ");
    scanf("%d", &number);
    printf("Enter the bit position to set (0-based): ");
    scanf("%d", &setBitPosition);
    printf("Enter the bit position to clear (0-based): ");
    scanf("%d", &clearBitPosition);
    printf("Enter the lower bound of the range: ");
    scanf("%d", &lowerRange);
    printf("Enter the upper bound of the range: ");
    scanf("%d", &upperRange);
    int setMask = 1 << setBitPosition;
    number = number | setMask;
    int clearMask = ~(1 << clearBitPosition);
    number = number & clearMask;
    int isOdd = (number % 2 != 0);
    int isWithinRange = (number >= lowerRange && number <=
upperRange);
    printf("Updated number after setting and clearing bits: %d\n",
number);
```

```
if (isOdd) {  
    printf("The resulting number is odd.\n");  
} else {  
    printf("The resulting number is even.\n");  
}
```

```
if (isWithinRange) {  
    printf("The resulting number lies within the range [%d, %d].\n",  
lowerRange, upperRange);  
} else {  
    printf("The resulting number does not lie within the range [%d,  
%d].\n", lowerRange, upperRange);  
}
```

```
return 0;  
}
```

O/p: Enter the initial number: 12

Enter the bit position to set (0-based): 3

Enter the bit position to clear (0-based): 1

Enter the lower bound of the range: 12                      5

Enter the upper bound of the range: 7

Updated number after setting and clearing bits: 12

The resulting number is even.

The resulting number does not lie within the range [5, 7].

## 22. Custom Mathematical Condition with Bits:

Given two integers a and b, perform the following:

Compute their sum and product (arithmetic operators).

Verify if the sum is greater than 100 and the product is divisible by 4 (relational and logical operators).

Check if the binary representation of a has its second bit set (bitwise AND with a mask).

Sol: #include <stdio.h>

```
int main() {
    int a, b;
    printf("Enter the first integer (a): ");
    scanf("%d", &a);
    printf("Enter the second integer (b): ");
    scanf("%d", &b);
    int sum = a + b;
    int product = a * b;
    int isSumGreaterThan100 = (sum > 100);
    int isProductDivisibleBy4 = (product % 4 == 0);
    int secondBitMask = 1 << 1;
    int isSecondBitSet = (a & secondBitMask) != 0;
    printf("Sum of a and b: %d\n", sum);
    printf("Product of a and b: %d\n", product);
    if (isSumGreaterThan100) {
        printf("The sum is greater than 100.\n");
    }
}
```

```
    } else {  
        printf("The sum is not greater than 100.\n");  
    }  
  
    if (isProductDivisibleBy4) {  
        printf("The product is divisible by 4.\n");  
    } else {  
        printf("The product is not divisible by 4.\n");  
    }  
  
    if (isSecondBitSet) {  
        printf("The second bit of a is set.\n");  
    } else {  
        printf("The second bit of a is not set.\n");  
    }  
  
    return 0;  
}
```

O/p: Enter the first integer (a): 24

Enter the second integer (b): 12

Sum of a and b: 36

Product of a and b: 288

The sum is not greater than 100.



The product is divisible by 4.

The second bit of a is not set.

## 23. If Statements

Check for Positivity:

Write a program to check if a number entered by the user is positive using an if statement.

Sol: #include <stdio.h>

```
int main() {  
    int number;  
    printf("Enter a number: ");  
    scanf("%d", &number);  
    if (number > 0) {  
        printf("The number is positive.\n");  
    }  
    return 0;  
}
```

O/p: Enter a number: 2

The number is positive.

## 24. Divisibility Check:

Write a program to check if a number is divisible by 3 using an if statement.

Sol: #include <stdio.h>

```
int main() {  
    int number;
```

```
printf("Enter a number: ");  
scanf("%d", &number);  
if (number % 3 == 0) {  
    printf("The number is divisible by 3.\n");  
}  
return 0;  
}
```

O/p: Enter a number: 21

The number is divisible by 3.

## 25. If-Else Statements

Odd or Even:

Write a program to determine if a number is odd or even using an if-else statement.

Sol: #include <stdio.h>

```
int main() {  
    int number;  
    printf("Enter a number: ");  
    scanf("%d", &number);  
    if (number % 2 == 0) {  
        printf("The number is even.\n");  
    } else {  
        printf("The number is odd.\n");  
    }  
}
```

```
return 0;
```

```
}
```

O/p: Enter a number: 5

The number is odd.

## 26. Passing Criteria:

Write a program to check if a student has passed an exam based on their marks (pass marks are 40). If the marks are below 40, display

Sol: #include <stdio.h>

```
int main() {
```

```
int marks;
```

```
printf("Enter the marks obtained: ");
```

```
scanf("%d", &marks);
```

```
if (marks >= 40) {
```

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

```
} else {
```

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

```
}
```

```
return 0;
```

```
}
```

O/p: Enter the marks obtained: 80

Pass

## 27. Nested If-Else Statements

Triangle Type Checker:

Given the lengths of three sides, write a program to determine if the triangle is valid using nested if-else. If valid, check if it is an equilateral triangle.

Sol: #include <stdio.h>

```
int main() {  
    int a, b, c;  
    printf("Enter the three sides of the triangle: ");  
    scanf("%d %d %d", &a, &b, &c);  
    if (a + b > c && a + c > b && b + c > a) {  
        printf("The triangle is valid.\n");  
        if (a == b && b == c) {  
            printf("It is an equilateral triangle.\n");  
        } else {  
            printf("It is not an equilateral triangle.\n");  
        }  
    } else {  
        printf("The triangle is not valid.\n");  
    }  
    return 0;  
}
```

O/p: Enter the three sides of the triangle: 23 12 50

The triangle is not valid.

28. Eligibility for Admission:

Write a program to check if a student is eligible for admission based on the following criteria:

Marks in mathematics  $\geq 50$

Marks in physics  $\geq 50$

Total marks (math + physics)  $\geq 120$

Use nested if-else statements.

Sol: #include <stdio.h>

```
int main() {  
    int mathMarks, physicsMarks, totalMarks;  
    printf("Enter the marks obtained in Mathematics: ");  
    scanf("%d", &mathMarks);  
    printf("Enter the marks obtained in Physics: ");  
    scanf("%d", &physicsMarks);  
    totalMarks = mathMarks + physicsMarks;  
    if (mathMarks  $\geq$  50) {  
        if (physicsMarks  $\geq$  50) {  
            if (totalMarks  $\geq$  120) {  
                printf("The student is eligible for admission.\n");  
            } else {  
                printf("The student is not eligible for admission due to  
insufficient total marks.\n");  
            }  
        } else {
```

```
printf("The student is not eligible for admission due to insufficient
marks in Physics.\n");

}

} else {

    printf("The student is not eligible for admission due to insufficient
marks in Mathematics.\n");

    } return 0;

}
```

O/p: Enter the marks obtained in Mathematics: 87

Enter the marks obtained in Physics: 54

The student is eligible for admission.

## 29. If-Else-If Ladder

Grade Calculator:

Write a program to calculate and print the grade of a student based on their percentage using an if-else-if ladder:

= 90: Grade A

= 75: Grade B

= 50: Grade C

< 50: Fail

Sol: #include <stdio.h>

```
int main() {

    float percentage;

    printf("Enter the percentage: ");

    scanf("%f", &percentage);
```

```
if (percentage >= 90) {  
    printf("Grade: A\n");  
} else if (percentage >= 75) {  
    printf("Grade: B\n");  
} else if (percentage >= 50) {  
    printf("Grade: C\n");  
} else {  
    printf("Grade: Fail\n");  
}  
return 0;  
}
```

O/p: Enter the percentage: 87

Grade: B

### 30. Number Classification:

Write a program to classify an integer as positive, negative, or zero using an if-else-if ladder.

Sol: #include <stdio.h>

```
int main() {  
    int number;  
    printf("Enter an integer: ");  
    scanf("%d", &number);  
    if (number > 0) {
```

```
    printf("The number is positive.\n");
} else if (number < 0) {
    printf("The number is negative.\n");
} else {
    printf("The number is zero.\n");
}
return 0;
}
```

O/p: Enter an integer: 34

The number is positive.

Enter an integer: -9

The number is negative.

### 31. Electricity Bill Calculation:

Write a program to calculate the electricity bill based on the number of units consumed using the following criteria:

Units  $\leq$  100: ₹5 per unit

Units  $>$  100 and  $\leq$  200: ₹7 per unit

Units  $>$  200: ₹10 per unit

Use an if-else-if ladder to implement this.

Sol: #include <stdio.h>

```
int main() {
    int units;
    float bill;
```



```
printf("Enter the number of units consumed: ");
scanf("%d", &units);
if (units <= 100) {
    bill = units * 5.0;
} else if (units <= 200) {
    bill = units * 7.0;
} else {
    bill = units * 10.0;
}
printf("The electricity bill is: ₹%.2f\n", bill);
return 0;
}
```

O/p: Enter the number of units consumed: 800

The electricity bill is: ₹8000.00

### 32. Day of the Week:

Write a program to print the name of the day of the week based on a number entered by the user (1 for Monday, 2 for Tuesday, ..., 7 for Sunday) using an if-else-if ladder.

Sol: #include <stdio.h>

```
int main() {
    int day;
    printf("Enter a number (1-7) for the day of the week: ");
    scanf("%d", &day);
    if (day == 1) {
```

```
    printf("Monday\n");
} else if (day == 2) {
    printf("Tuesday\n");
} else if (day == 3) {
    printf("Wednesday\n");
} else if (day == 4) {
    printf("Thursday\n");
} else if (day == 5) {
    printf("Friday\n");
} else if (day == 6) {
    printf("Saturday\n");
} else if (day == 7) {
    printf("Sunday\n");
} else {
    printf("Invalid input! Please enter a number between 1 and 7.\n");
}

return 0;
}
```

O/p: Enter a number (1-7) for the day of the week: 3

Wednesday

### 33. Switch Case

Write a program that takes an integer (1-7) as input and uses a switch-case to print the corresponding day of the week (e.g., 1 for Monday, 2 for Tuesday, etc.).

Sol: #include <stdio.h>

```
int main() {  
    int day;  
    printf("Enter a number (1-7): ");  
    scanf("%d", &day);  
    switch(day) {  
        case 1:  
            printf("Monday\n");  
            break;  
        case 2:  
            printf("Tuesday\n");  
            break;  
        case 3:  
            printf("Wednesday\n");  
            break;  
        case 4:  
            printf("Thursday\n");  
            break;  
        case 5:  
            printf("Friday\n");  
            break;  
        case 6:  
            printf("Saturday\n");
```

```

        break;
    case 7:
        printf("Sunday\n");
        break;
    default:
        printf("Invalid input! Please enter a number between 1 and
7.\n");
    }
    return 0;
}

```

O/p: Enter a number (1-7): 5

Friday

34. Write a program to perform basic arithmetic operations (addition, subtraction, multiplication, division) based on the operator input (+, -, \*, /) using a switch-case statement.

Sol: #include <stdio.h>

```

int main() {
    float num1, num2;
    char operator;
    printf("Enter first number: ");
    scanf("%f", &num1);
    printf("Enter an operator (+, -, *, /): ");
    scanf(" %c", &operator);
}

```

```
printf("Enter second number: ");
scanf("%f", &num2);
switch(operator) {
    case '+':
        printf("Result: %.2f\n", num1 + num2);
        break;
    case '-':
        printf("Result: %.2f\n", num1 - num2);
        break;
    case '*':
        printf("Result: %.2f\n", num1 * num2);
        break;
    case '/':
        if (num2 != 0) {
            printf("Result: %.2f\n", num1 / num2);
        } else {
            printf("Error! Division by zero.\n");
        }
        break;
    default:
        printf("Invalid operator! Please enter one of +, -, *, /\n");
}
return 0;
```

```
}
```

O/p: Enter first number: 12

Enter an operator (+, -, \*, /): +

Enter second number: 24

Result: 36.00

Enter first number: 12

Enter an operator (+, -, \*, /): -

Enter second number: 24

Result: -12.00

Enter first number: 12

Enter an operator (+, -, \*, /): \*

Enter second number: 24

Result: 288.00

Enter first number: 12

Enter an operator (+, -, \*, /): /

Enter second number: 24

Result: 0.50

35. Write a program that takes a single character as input and uses a switch-case to determine if it is a vowel or a consonant.

Sol: #include <stdio.h>

```
int main() {
```

```
    char ch;
```

```
printf("Enter a character: ");
scanf(" %c", &ch);
switch(ch) {
    case 'a':
    case 'A':
    case 'e':
    case 'E':
    case 'i':
    case 'I':
    case 'o':
    case 'O':
    case 'u':
    case 'U':
        printf("The character '%c' is a vowel.\n", ch);
        break;
    default:
        printf("The character '%c' is a consonant.\n", ch);
}
return 0;
}
```

O/p: Enter a character: a

The character 'a' is a vowel.

36. Write a program to convert a single-digit number (0-9) into its word representation (e.g., 1 to "One", 2 to "Two") using a switch-case statement.

Sol: #include <stdio.h>

```
int main() {
```

```
    int num;
```

```
    // Prompt the user to enter a single-digit number
```

```
    printf("Enter a single-digit number (0-9): ");
```

```
    scanf("%d", &num);
```

```
    // Use a switch-case statement to print the word representation
```

```
    switch (num) {
```

```
        case 0:
```

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

```
            break;
```

```
        case 1:
```

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

```
            break;
```

```
        case 2:
```

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

```
            break;
```

```
        case 3:
```



```
        printf("Three\n");
        break;
case 4:
        printf("Four\n");
        break;
case 5:
        printf("Five\n");
        break;
case 6:
        printf("Six\n");
        break;
case 7:
        printf("Seven\n");
        break;
case 8:
        printf("Eight\n");
        break;
case 9:
        printf("Nine\n");
        break;
default:
        printf("Invalid input! Please enter a number between 0 and
9.\n");
```

```
        break;
    }

    return 0;
}
```

O/p: Enter a single-digit number (0-9): 4

Four

37. Write a program that takes an integer (1-12) as input and uses a switch-case to print the name of the corresponding month (e.g., 1 for January, 2 for February, etc.).

Sol: #include <stdio.h>

```
int main() {
    int month;

    printf("Enter a number (1-12): ");
    scanf("%d", &month);

    switch(month) {
        case 1:
            printf("January\n");
            break;
        case 2:
            printf("February\n");
            break;
        case 3:
```

```
    printf("March\n");  
    break;  
case 4:  
    printf("April\n");  
    break;  
case 5:  
    printf("May\n");  
    break;  
case 6:  
    printf("June\n");  
    break;  
case 7:  
    printf("July\n");  
    break;  
case 8:  
    printf("August\n");  
    break;  
case 9:  
    printf("September\n");  
    break;  
case 10:  
    printf("October\n");  
    break;
```

```

    case 11:
        printf("November\n");
        break;
    case 12:
        printf("December\n");
        break;
    default:
        printf("Invalid input! Please enter a number between 1 and 12.\n");
}

return 0;
}

```

O/p: Enter a number (1-12): 8

August

38: Write a program that takes a grade (A, B, C, D, F) as input and uses a switch-case to print the description of the grade (e.g., A: "Excellent", B: "Good", etc.).

Sol: #include <stdio.h>

```

int main() {
    char grade;
    printf("Enter a grade (A, B, C, D, F): ");
    scanf(" %c", &grade);
    switch(grade) {
        case 'A':

```

```
case 'a':
    printf("Excellent\n");
    break;
case 'B':
case 'b':
    printf("Good\n");
    break;
case 'C':
case 'c':
    printf("Average\n");
    break;
case 'D':
case 'd':
    printf("Below Average\n");
    break;
case 'F':
case 'f':
    printf("Fail\n");
    break;
default:
    printf("Invalid input! Please enter a grade between A and F.\n");
}
```

```
    return 0;
}
```

O/p: Enter a grade (A, B, C, D, F): B

Good

39: Write a menu-driven program that offers the user options for basic mathematical operations (addition, subtraction, etc.). Based on the user's choice, perform the corresponding operation using a switch-case.

Sol: #include <stdio.h>

```
int main() {
    float num1, num2;
    int choice;

    while (1) {
        printf("Menu:\n");
        printf("1. Addition\n");
        printf("2. Subtraction\n");
        printf("3. Multiplication\n");
        printf("4. Division\n");
        printf("5. Exit\n");
        printf("Enter your choice (1-5): ");
        scanf("%d", &choice);

        if (choice == 5) {
```

```
    printf("Exiting program.\n");  
    break;  
}
```

```
printf("Enter first number: ");  
scanf("%f", &num1);
```

```
printf("Enter second number: ");  
scanf("%f", &num2);
```

```
switch(choice) {  
    case 1:  
        printf("Result: %.2f\n", num1 + num2);  
        break;  
    case 2:  
        printf("Result: %.2f\n", num1 - num2);  
        break;  
    case 3:  
        printf("Result: %.2f\n", num1 * num2);  
        break;  
    case 4:  
        if (num2 != 0) {  
            printf("Result: %.2f\n", num1 / num2);  
        }
```

```
        } else {
            printf("Error! Division by zero.\n");
        }
        break;
    default:
        printf("Invalid choice! Please select a valid option (1-5).\n");
    }
}

return 0;
}
```

O/p: Menu:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5): 3

Enter first number: 12

Enter second number: 2

Result: 24.00

40. Write a program to simulate a traffic light system. Take input as R, Y, or G (Red, Yellow, Green) and use a switch-case to display the corresponding action (e.g., R for Stop, Y for Get Ready, G for Go).



Sol: #include <stdio.h>

```
int main() {
```

```
    char trafficLight;
```

```
    printf("Enter the traffic light color (R for Red, Y for Yellow, G for  
Green): ");
```

```
    scanf(" %c", &trafficLight);
```

```
    switch(trafficLight) {
```

```
        case 'R':
```

```
        case 'r':
```

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

```
            break;
```

```
        case 'Y':
```

```
        case 'y':
```

```
            printf("Get Ready\n");
```

```
            break;
```

```
        case 'G':
```

```
        case 'g':
```

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

```
            break;
```

```
        default:
```

```
            printf("Invalid input! Please enter R, Y, or G.\n");
```

```
    }
```

```
    return 0;
```

```
}
```

O/p: Enter the traffic light color (R for Red, Y for Yellow, G for Green):  
green

Go

41. Write a program that takes the year as input and uses a switch-case to check and print whether it is a leap year or not (use logical division by 4 and additional logic in cases).

Sol: #include <stdio.h>

```
int main() {
```

```
    int year;
```

```
    printf("Enter a year: ");
```

```
    scanf("%d", &year);
```

```
    switch (year % 4) {
```

```
        case 0:
```

```
            if (year % 100 == 0) {
```

```
                if (year % 400 == 0) {
```

```
                    printf("%d is a leap year.\n", year);
```

```
                } else {
```

```
                    printf("%d is not a leap year.\n", year);
```

```
                }
```

```
            } else {
```

```
        printf("%d is a leap year.\n", year);
    }
    break;
default:
    printf("%d is not a leap year.\n", year);
}

return 0;
}
```

O/p: Enter a year: 2024

2024 is a leap year.

42. Write a program to calculate the area of different shapes based on user input:

1 for Circle

2 for Rectangle

3 for Triangle

Use a switch-case to perform the respective area calculations.

Sol: #include <stdio.h>

#define PI 3.14159

int main() {

int choice;

float area, radius, length, width, base, height;

printf("Select a shape to calculate the area:\n");

```
printf("1. Circle\n");
printf("2. Rectangle\n");
printf("3. Triangle\n");
printf("Enter your choice (1-3): ");
scanf("%d", &choice);
switch(choice) {
    case 1:
        printf("Enter the radius of the circle: ");
        scanf("%f", &radius);
        area = PI * radius * radius;
        printf("Area of the circle: %.2f\n", area);
        break;
    case 2:
        printf("Enter the length and width of the rectangle: ");
        scanf("%f %f", &length, &width);
        area = length * width;
        printf("Area of the rectangle: %.2f\n", area);
        break;
    case 3:
        printf("Enter the base and height of the triangle: ");
        scanf("%f %f", &base, &height);
        area = 0.5 * base * height;
        printf("Area of the triangle: %.2f\n", area);
```

```
        break;
    default:
        printf("Invalid choice! Please select a valid option (1-3).\n");
    }

    return 0;
}
```

O/p: Select a shape to calculate the area:

1. Circle
2. Rectangle
3. Triangle

Enter your choice (1-3): 1

Enter the radius of the circle: 12.4

Area of the circle: 483.05

Select a shape to calculate the area:

1. Circle
2. Rectangle
3. Triangle

Enter your choice (1-3): 2

Enter the length and width of the rectangle: 12 45

Area of the rectangle: 540.00

Select a shape to calculate the area:

1. Circle

2. Rectangle

3. Triangle

Enter your choice (1-3): 3

Enter the base and height of the triangle: 12 5

Area of the triangle: 30.00