Data types: *

Bio-data program

Operators: *

All the programs in your assignment 2

Basic if, else, else if, nested if else: *

1. Check a number is positive or negative
2. Check a number is even or odd
3. Write a C program that classifies a number as positive, negative, or zero using an else if ladder.
4. C program that checks if a number is positive or negative, and if it's positive, it further checks whether it's an even or odd number.
5. Program to Check Eligibility to Vote (if-else)
6. C program that assigns a grade to a student based on their marks using the else if ladder.
7. C program that finds the largest of three numbers using the else if ladder.
8. Nested if-else to Check Eligibility for a Loan

Ternary operator: *

1. Program to Check if a Number is Positive, Negative, or Zero Using Ternary Operator
2. Program to Find the Largest of Two Numbers Using Ternary Operator
3. Program to Check Voting Eligibility Using Ternary Operator

Switch statement:

1. Write C program takes a character as input and checks whether it's a vowel or consonant using the switch statement.
2. Write C program takes a student's marks as input and determines their grade using the switch statement.*
3. program takes a month number (1-12) as input and displays the corresponding season using the switch statement.*

Pattern Programs

Write a C program for the following Patterns

1. **Basic patterns***

i)
```
*
**
***
****
*****
```

ii)
```
*****
****
***
**
*
```

iii)
```
1
12
123
1234
12345
```

iv)
```
1
22
333
```

```
4444
55555
```

v)
```
a a a a
b b b b
c c c c
d d d d
```

2. Intermediate patterns*

i)
```
    *
   ***
  *****
 *******
*********
```

ii)
```
*********
 *******
  *****
   ***
    *
```

iii)
```
   1
  2 3
 4 5 6
7 8 9 10
```

iv)
```
1
2 3
4 5 6
7 8 9 10
```

3. Advanced patterns

i)
```
    1 *
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

ii)
```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

iii)
```
    1
   121
  12321
 1234321
123454321
```

1234321
12321
121
1

Break and continue:

1. Print the odd numbers only *
2. Print the positive numbers only *
3. Write a c program to generates a multiplication table from 1 to 10, but skips the multiples of 5 using continue. *
4. C program to keeps accepting user input until the user enters a negative number, at which point the loop is terminated using break. *
5. C program searches for the first odd number in a range and stops the search using break once it's found. *

Number Theory Problems:

i) Write a C program to check whether a number is **Armstrong number** or not *
ii) Write a C program to check a whether number is **Prime number** or not *
iii) Write a C program to check a whether number is **Perfect number** or not *
iv) Write a C program to check a whether number is **Strong number** or not *
v) Write a C program to check a whether number is **Automorphic number** or not
vi) Write a C program that finds the greatest common divisor **(GCD)** of two numbers.
vii) program finds the Least common divisor **(LCM)** of two numbers.

Mathematical Operations:

1. Write a C program to find Power of a Number
2. Write a C program to Factorial Calculation *
3. Write a C program to find Sum of First N Natural Numbers *
4. Write a C program to Simple interest calculation *
5. Write a C program to Convert Celsius to Fahrenheit *

Digit Manipulation:

1. Reverse a Number *
2. Sum of Digits *
3. Palindrome Number Check *
4. Count number of digits in a number *
5. Counting how many times a digit appears in a number
6. Removing unnecessary leading zeros from a number
7. Converting a decimal number to binary
8. Separating even and odd digits of a number

Sequence Generation:

1. Fibonacci Sequence *
2. Arithmetic Progression
3. Geometric Progression

Arrays:

1. Input and Display Elements of an Array *

2. Find the Sum and Average of Elements in an Array *
3. Find the Maximum and Minimum Elements in an Array *
4. Count Even and Odd Elements in an Array *
5. Print the even and odd numbers in an array *
6. Find the Second Largest Element in an Array *
7. Find the positive and negative numbers in an array *
8. Print the frequency count of an element in an array *
9. Insert an Element at a Specific Position in an Array
10. Delete an Element from a Specific Position in an Array
11. Remove Duplicate Elements from an Array *
12. Search an Element in an Array **(Linear Search)** *
13. Sort an Array in Ascending Order **(Selection sort)** *
14. Sort an Array in Descending Order **(Selection sort)** *

2D Array:

1. Addition of Two Matrices *
2. Transpose of a Matrix *
3. Sum of elements in a 2D array *
4. Multiplication of Two Matrices

String:*

1. Compare two strings
2. Concatenate two strings
3. Copy a string
4. Find the length of a string
5. Convert a string to uppercase
6. Convert a string to lowercase
7. Count number of vowels and constants in a string
8. Check if a string is palindrome or not