**[1] Write a program to read n numbers from the keyboard and find their sum.**

```
t#include <stdio.h>

int main()

 {

   int n, num, sum = 0, i = 0;

   printf("Enter n number: ");

   scanf("%d", &n);

   printf("Enter %d numbers:\n", n);

   while (i < n)

   {

      scanf("%d", &num);

      sum += num;

      i++;

   }

   printf("Sum: %d\n", sum);

   return 0;

}
```

**Output :**

```
Enter n number: 3

Enter 3 numbers:

10

20

10

Sum: 40
```

**Explaination :** This program asks the user how many numbers they want to enter (n). Then, it takes n numbers one by one, adds them together, and finally prints the total sum. The while loop runs n times to collect and sum the number.

**[2] Write a program to read a number n, and print the sum of all the numbers from 1 to n.**

```
#include <stdio.h>

Int main() {

   Int n, sum = 0, I = 1;
```

```
   Printf(“Enter a number: “);
   Scanf(“%d”, &n);
   While (I <= n) {
      Sum += I;
      I++;
   }
   Printf(“Sum of numbers from 1 to %d is: %d\n”, n, sum);
   Return 0;
}
```

**Output :**

Enter a number: 5

Sum of numbers from 1 to 5 is: 15

**Explanation :** This program takes a number n from the user and adds all the numbers from 1 to n. It keeps adding numbers one by one using a while loop until it reaches n, then prints the final sum.

**[3] Write a program to read a number n and print the factorial of n.**

```
#include <stdio.h>

int main() {
   int n, fact = 1, i = 1;
   printf("Enter n number: ");
   scanf("%d", &n);
   while (i <= n)
 {
      fact *= i;
      i++;
   }
   printf("Factorial of %d is %d\n", n, fact);
   return 0;
}
```

**Output :**

Enter n number : 4

Factorial of 4 is 24

**Explanation :**

The program calculates the factorial of a number **n**. The factorial of a number (**n!**) is the product of all numbers from **1 to n**.

**4! = 4 × 3 × 2 × 1 = 24**

The program takes **n** as input, multiplies numbers from **1 to n**, and prints the result.

**[4] Write a program to print the value of below series:**

```
#include <stdio.h>

int main()

{

   int n;

   double sum = 0.0;

   printf("Enter n number: ");

   scanf("%d", &n);

   for (int i = 1; i <= n; i++)

   {

      if (i % 2 == 0)

         sum -= 1.0 / i;  // Subtract if i is even

      else

         sum += 1.0 / i;  // Add if i is odd

   }

   printf("Sum of the series: %.6f\n", sum);

   return 0;

}
```

**Output :**

Enter n number: 3

Sum of the series: 0.833333

**Explanation :** This program calculates the sum of the alternating series **1 - 1/2 + 1/3 - 1/4 + ... + 1/n** based on user input. It loops from **1 to n**, adding or subtracting terms depending on whether the index is odd or even, and then prints the final sum.

**[5] Write a program to multiply two numbers without using * operator.**

```
#include <stdio.h>

Int main()

{

   Int num1, num2, result = 0;


   Printf(“Enter two numbers:\n”);

   Scanf(“%d %d”, &num1, &num2);

   While (num2 > 0) {

      Result += num1;

      Num2--;

   }

   Printf(“Result is %d\n”, result);

   Return 0;

}
```

**Output :**

Enter two numbers:

5

3

Result is 15

**Explanation :** This program multiplies two numbers without using the * operator. Instead, it uses a loop to repeatedly add one number to itself based on the other number's value.

**[6] Write a program to find the power of a number to the given number.**

```
#include <stdio.h>

Int main()

 {

   Int base, exponent, result = 1;
```

```
   Printf(“Enter base and exponent: “);
   Scanf(“%d %d”, &base, &exponent);
   For (int I = 0; I < exponent; i++)
.  {
      Result *= base;
   }
   Printf(“%d to the power of %d -> %d\n”, base, exponent, result);
   Return 0;
}
```

**Output :**

Enter base and exponent: 3 5

3 to the power of 5 -> 243

**Explanation :** This program calculates the power of a number by multiplying the base with itself repeatedly for the given exponent using a loop. It takes two inputs (base and exponent) and prints the value.

Write the following programs, using while loop, with counting based logic :

[1] Read n number of numbers from user (n value taken as input) and print if each number is even or odd.

```
#include <stdio.h>

Int main() {
   Int n, x, count = 0;  // Counter initialized to 0

   Printf(“Enter n number: “);
   Scanf(“%d”, &n);

   While (count < n) {  // Loop runs n times
      Printf(“Enter x value: “);
      Scanf(“%d”, &x);

      If (x % 2 == 0)
         Printf(“The given number %d is EVEN.\n”, x);
```

```
    Else
        Printf("The given number %d is ODD.\n", x);

    Count++;  // Increment the counter
  }

  Return 0;
}
```

**Output :**

Enter n number: 2

Enter x value: 5

The given number 5 is ODD.

Enter x value: 4

The given number 4 is EVEN.

**Explanation :** The program reads an integer n from the user, which represents how many numbers will be entered. Then, using a **while loop**, it reads n numbers one by one and checks if each number is **even or odd**. It prints the result for each number accordingly. This approach is called **counting-based logic** because the loop runs a fixed number of times (n), ensuring each input is processed systematically.

[2] Read n characters from user, and print if each character is vowel or not.

```
#include <stdio.h>

Int main() {
  Int n, I = 0;
  Char ch;

  Printf("Enter n characters: ");
  Scanf("%d", &n);

  While (I < n) {
    Printf("Enter a character: ");
    Scanf(" %c", &ch); // Space before %c to ignore newline from previous input
```

```
    If (ch == ‘a’ || ch == ‘e’ || ch == ‘I’ || ch == ‘o’ || ch == ‘u’ ||
        Ch == ‘A’ || ch == ‘E’ || ch == ‘I’ || ch == ‘O’ || ch == ‘U’) {
        Printf(“The given character ‘%c’ is a vowel.\n”, ch);
    } else {
        Printf(“The given character ‘%c’ is not a vowel.\n”, ch);
    }

    I++;
  }

  Return 0;
}
```

**Output :**

```
Enter n characters: 2
Enter a character: b
The given character ‘b’ is not a vowel.
Enter a character: i
The given character ‘I’ is a vowel.
```

**Explanation :** This program reads **n** characters from the user and checks whether each character is a **vowel or not** using a **while loop**. It takes an integer **n** as input, then iterates **n** times, asking for a character in each iteration. The program compares the input with vowels **(a, e, I, o, u)** (both uppercase and lowercase) and prints the result accordingly. The loop ensures that each character is checked before moving to the next input.

[3] Read n characters from user and print if each character is a capital alphabet, small alphabet, numeric character or special character.

```
#include <stdio.h>

Int main() {
  Int n, I = 0;
  Char ch;

  // Taking input for number of characters
```

```
    Printf(“Enter n characters: “);
    Scanf(“%d”, &n);

    // Using while loop to process each character
    While (I < n) {
        Printf(“Enter a character: “);
        Scanf(“ %c”, &ch); // Space before %c to avoid newline issue

        // Checking the type of character
        If (ch >= ‘A’ && ch <= ‘Z’) {
            Printf(“The given character ‘%c’ is a Capital alphabet.\n”, ch);
        } else if (ch >= ‘a’ && ch <= ‘z’) {
            Printf(“The given character ‘%c’ is a Small alphabet.\n”, ch);
        } else if (ch >= ‘0’ && ch <= ‘9’) {
            Printf(“The given character ‘%c’ is a Numeric character.\n”, ch);
        } else {
            Printf(“The given character ‘%c’ is a Special character.\n”, ch);
        }

        I++; // Incrementing counter
    }

    Return 0;
}
```

**Output :**

Enter n characters: 3

Enter a character: b

The given character ‘b’ is a Small alphabet.

Enter a character: ?

The given character ‘?’ is a Special character.

Enter a character: B

The given character ‘B’ is a Capital alphabet.

**Explanation :** This program reads n characters from the user and classifies each one as a capital alphabet, small alphabet, numeric character, or special character using a while loop.

How it works:

1. The user enters the number of characters (n).
2. The program loops n times, asking for a character each time.
3. It checks the character's type:
    - 'A' to 'Z' → Capital Alphabet
    - 'a' to 'z' → Small Alphabet
    - '0' to '9' → Numeric Character
    - Others → Special Character
4. It prints the classification of each character.

This ensures all types of characters are identified correctly.

[4] Read n numbers from the user, and while reading every number, print if the number is bigger or smaller than the previous number. For the first number, there won’t be any output as it is the first one.

```
#include <stdio.h>


Int main() {
   Int n, num, prev, I = 1;


   Printf(“Enter n number: “);
   Scanf(“%d”, &n);


   If (n > 0) {
      Printf(“Enter a number: “);
      Scanf(“%d”, &prev); // First number, no comparison


      While (I < n) {
         Printf(“Enter a number: “);
         Scanf(“%d”, &num);


         If (num > prev)
```

```
            Printf(“The given number is bigger than the previous number.\n”);
        Else if (num < prev)
            Printf(“The given number is smaller than the previous number.\n”);

        Prev = num; // Update previous number for next iteration
        I++;
    }
  }

  Return 0;
}
```

**Output :**

Enter n number: 3

Enter a number: 5

Enter a number: 6

The given number is bigger than the previous number.

Enter a number: 2

The given number is smaller than the previous number.

**Explanation :** This program reads **n numbers** from the user and compares each number with the **previous number**. The first number is taken as the reference, so no comparison is made for it. For the rest of the numbers:

- If the number is greater than the previous one, it prints ”bigger than the previous number.”

- If the number is smaller than the previous one, it prints ”smaller than the previous number.”

This continues until all n numbers are processed.

[5] Read n numbers in ascending order. If a number entered is bigger than the previous number, then count it. If it is smaller, then don’t count it, instead let the loop repeat itself and read another number.

```
#include <stdio.h>
Int main() {
  Int n, num, prev, count = 0;

  // Taking input for number of values
  Printf(“Enter n number: “);
```

```
    Scanf(“%d”, &n);

    Printf(“Enter %d numbers:\n”, n);

    // Read the first number
    Scanf(“%d”, &prev);
    Count++; // First number is always counted

    For (int I = 1; I < n; ) { // I is only incremented when a valid number is entered
        Scanf(“%d”, &num);

        If (num > prev) { // Check if the number is greater than the previous number
            Count++;
            Prev = num;
            I++; // Only increment count if valid input is given
        }
    }

    Printf(“The count is %d\n”, count);
    Return 0;
}
```

**Output :**

Enter n number: 5

Enter 5 numbers:

3

4

2  (ignored)

6

8

The count is 3

**Explanation :** This program reads `n` numbers from the user and counts only the numbers that are greater than the previous number. If a number is smaller or equal to the previous one, it is ignored, and the user is asked to enter another number. The final count of valid numbers is displayed.

[6] Read  n numbers from the user ,and print the smallest number of all.

```
#include <stdio.h>


int main() {
    int n, num, smallest, i = 0;


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


    printf("Enter %d numbers:\n", n);


    while (i < n) {
        scanf("%d", &num);


        if (i == 0) {
            smallest = num;  // Initialize the first number as the smallest
        } else if (num < smallest) {
            smallest = num;  // Update if a smaller number is found
        }


        i++;
    }


    printf("The smallest number is %d.\n", smallest);


    return 0;
}
```

**Output :**

Enter n number: 5

Enter 5 numbers:

3

4

2

6

8

The smallest number is 2.

**Explanation :** This program reads n numbers from the user and determines the smallest one. It starts by assuming the first number is the smallest, then compares each subsequent number to update the smallest value if necessary. Finally, it prints the smallest number.

[7] Read a number from the user, and print its multiplication table upto 10 multiples.

```
#include <stdio.h>


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


   // Asking for the number of tables to generate
   printf("Enter n number: ");
   scanf("%d", &n);


   while (count < n) {
      printf("Enter x number: ");
      scanf("%d", &x);


      // Printing the multiplication table for x
      for (i = 1; i <= 10; i++) {
         printf("%d * %d = %d\n", x, i, x * i);
      }
      count++;  // Increment the count to track how many numbers are processed
   }
```

```
    return 0;
}
```

**Output :**

```
Enter n number: 2
Enter x number: 2
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
Enter x number: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

**Explanation :** This program reads a number n from the user, which indicates how many multiplication tables to generate. For each number x entered, it prints its multiplication table up to 10 multiples. The program uses a loop to display each multiplication step in the format x * I = result.

[8] For n students from a class, read 6 subject marks for every student, and calculate their percentage of marks, and print who is the topper of all (print the student number).

```
#include <stdio.h>


int main() {
    int n, i, j, topper_id = 1;
    float marks, sum, percentage, highest_percentage = 0;

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

    for (i = 1; i <= n; i++) {
        sum = 0;  // Reset sum for each student

        printf("Enter %d student 6 subjects marks:\n", i);
        for (j = 0; j < 6; j++) {
            scanf("%f", &marks);
            sum += marks;
        }

        // Calculate percentage
        percentage = (sum / 600) * 100;
        printf("Student %d percentage is: %.2f%%\n", i, percentage);

        // Check if this student is the topper
        if (percentage > highest_percentage) {
            highest_percentage = percentage;
            topper_id = i;
        }
    }
```

```
    // Print the topper student ID
    printf("Topper student ID is %d\n", topper_id);


    return 0;
}
```

**Output :**

```
Enter n students: 2
Enter 1st student 6 subjects marks:
40
50
45
75
60
55
Student 1 percentage is: 54.16%
Enter 2nd student 6 subjects marks:
45
65
55
80
40
55
Student 2 percentage is: 56.66%
Topper student ID is 2
```

**Explanation :** This program takes the number of students as input and then reads the marks of six subjects for each student. It calculates the percentage of marks for each student by summing up their scores and dividing by the total possible marks. As it processes each student's marks, it keeps track of the highest percentage to determine the topper. After all students' percentages are calculated and displayed, the program identifies and prints the student ID of the topper, ensuring that the highest-scoring student is recognized.

[9] Read two numbers from the user, and print all serial numbers between those numbers.

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

```
int main() {
    int start, end;

    // Taking input from the user
    printf("Enter two numbers: ");
    scanf("%d %d", &start, &end);

    // Printing serial numbers between start and end
    for (int i = start; i <= end; i++) {
        printf("%d ", i);
    }

    printf("\n");
    return 0;
}
```

**Output :**

For example, if the input is:

10 20

The output will be:

10 11 12 13 14 15 16 17 18 19 20

**Explanation :** This program takes two numbers as input from the user and prints all the numbers between them, including both. It uses a for loop to iterate from the starting number to the ending number, printing each number in sequence. If the user inputs 10 and 20, the output will be 10 11 12 13 14 15 16 17 18 19 20.

[10] write the calculator program to read two numbers and one character (+, -, *, / , %) from the user, and based on character, do appropriate operations on numbers and print the output. Modify the program to repeat this task n number of times.

```
#include <stdio.h>

Int main() {

    Int n, num1, num2, result, I = 0;
```

```
Char op;

// Read the number of calculations

Printf(“Enter n number: “);

Scanf(“%d”, &n);


While (I < n) {

   // Read two numbers

   Printf(“Enter two numbers:\n”);

   Scanf(“%d %d”, &num1, &num2);

   // Read the operator

   Printf(“Enter a character (+,-,*,/,%%):\n”);

   Scanf(“ %c”, &op);  // Space before %c to avoid reading newline

   // Perform calculation based on operator

   Switch (op) {

      Case ‘+’:

         Result = num1 + num2;

         Printf(“Result of %d + %d is %d\n”, num1, num2, result);

         Break;

      Case ‘-‘:

         Result = num1 – num2;

         Printf(“Result of %d - %d is %d\n”, num1, num2, result);

         Break;

      Case ‘*’:

         Result = num1 * num2;

         Printf(“Result of %d * %d is %d\n”, num1, num2, result);
```

```
            Break;
        Case '/':
            If (num2 != 0) {
                Result = num1 / num2;
                Printf("Result of %d / %d is %d\n", num1, num2, result);
            } else {
                Printf("Error: Division by zero is not allowed.\n");
            }
            Break;
        Case '%':
            If (num2 != 0) {
                Result = num1 % num2;
                Printf("Result of %d %% %d is %d\n", num1, num2, result);
            } else {
                Printf("Error: Modulo by zero is not allowed.\n");
            }
            Break;
        Default:
            Printf("Invalid operator! Please enter a valid one (+, -, *, /, %%).\n");
    }
    I++;
  }
  Return 0;
}
```

**Output :**

Enter n number: 2

Enter two numbers:

5

4

Enter a character (+,-,*,/,%):

+

Result of 5 + 4 is 9

Enter two numbers:

4

3

Enter a character (+,-,*,/,%):

-

Result of 4 – 3 is 1

**Explanation :** This program acts as a simple calculator that performs arithmetic operations based on user input. The user first specifies how many calculations (`n`) they want to perform. For each calculation, the program asks for two numbers and an arithmetic operator (`+`, `-`, `*`, `/`, `%`). Using a `switch` statement, it executes the corresponding operation and displays the result. The program also handles division and modulo operations carefully, ensuring the second number is not zero to prevent errors. This process repeats `n` times, allowing the user to perform multiple calculations in one execution.

[11] Read two numbers from the user, and print all odd numbers between those numbers and then all even numbers.

```
#include <stdio.h>

Int main() {

   Int start, end, num;

   // Taking input from the user

   Printf(“Enter two numbers: “);

   Scanf(“%d %d”, &start, &end);

   // Printing odd numbers first

   Printf(“Odd numbers: “);
```

```
    Num = start;
    While (num <= end) {
        If (num % 2 != 0) {
            Printf("%d ", num);
        }
        Num++;
    }
    Printf("\nEven numbers: ");
    Num = start;
    While (num <= end) {
        If (num % 2 == 0) {
            Printf("%d ", num);
        }
        Num++;
    }
    Printf("\n");
    Return 0;
}
```

**Output :**

Enter two numbers: 10 20

Odd numbers: 11 13 15 17 19

Even numbers: 10 12 14 16 18 20

**Explanation :** This program takes two numbers as input from the user and prints all the odd and even numbers between them separately. First, it identifies and prints all even numbers, followed by all odd numbers. The program iterates through the range using a loop and checks whether each number is even or odd using the modulus (%) operator. Even numbers (num % 2 == 0) are printed first, and odd numbers (num % 2 != 0) are printed afterward.

[12] Read two numbers from the user, and print all numbers in reverse order.

```
#include <stdio.h>


Int main() {
    Int start, end;
    // Taking input from the user
    Printf(“Enter two numbers: “);
    Scanf(“%d %d”, &start, &end);
    // Printing numbers in reverse order
    Int num = end;
    While (num >= start) {
        Printf(“%d “, num);
        Num--;
    }
    Printf(“\n”);
    Return 0;
}
```

**Output :**

Enter two numbers: 10 20

20 19 18 17 16 15 14 13 12 11 10

**Explanation :** This program takes two numbers as input from the user and prints all the numbers between them in reverse order using a while loop. It starts from the second number and decrements down to the first number, printing each value along the way. This ensures that the numbers are displayed in descending order.