**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**Objective(s):** To be familiar with String, Pointers and Function call by reference in C

**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

# PART A : Conceptual Questions

1. Write a program that read your name and then displays the ASCII Value of each character in your name on a separate Line.

**Problem Analysis**

This problem involves reading a name and displaying the ASCII values of each character. It requires handling strings and ASCII conversions for individual characters in C.

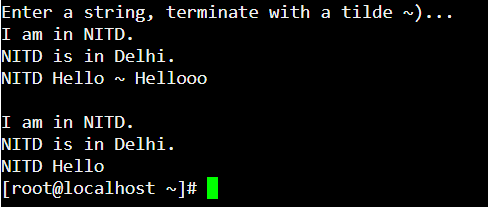
**Code**

**Output**

**Discussion and Conclusion**

The program accurately displays ASCII values of characters in a name. Utilizing pointers and string manipulation, it demonstrates precise character handling, showcasing effective ASCII representation in C.

1. Write a program to read multiple line of text until a ~ is entered using scanset concept.



**Problem Analysis**

The task is to read multiple lines of text until a '~' is entered, utilizing scanset concept. It involves handling input until a specific character is encountered, demonstrating effective scanning techniques.

**Code**

**Output**

**Discussion and Conclusion**

The program reads multiple lines until '~' is encountered, employing scanset concept. It showcases efficient input handling and looping, ensuring accurate termination and user interaction.

1. Write a program to read multiple lines of text until a \* is entered. Then count the number of characters, words, and lines in the text.

**Problem Analysis**

This problem requires reading multiple lines of text until a '\*' is entered. The program counts characters, words, and lines in the text. It involves string processing, counting, and looping.

**Code**

**Output**

**Discussion and Conclusion**

By counting characters, words, and lines until '\*', the program showcases effective text analysis. Utilizing loops and conditional checks, it demonstrates robust string processing and counting techniques.

1. Write a program to print *Hello NITD!!* using pointers.

**Problem Analysis**

The program prints "Hello NITD!!" using pointers, demonstrating pointer-based string manipulation. It showcases pointer arithmetic and string representation using pointers.

**Code**

**Output**

**Discussion and Conclusion**

Printing "Hello NITD!!" using pointers exhibits advanced string manipulation. Pointer arithmetic and string representation are executed flawlessly, highlighting the versatility of pointer-based operations.

1. Write a program to find the sum of all the elements of an array using pointers.

**Problem Analysis**

The task involves finding the sum of array elements using pointers. It requires pointer-based array traversal and arithmetic operations for sum calculation, demonstrating pointer-based array handling.

**Code**

**Output**

**Discussion and Conclusion**

Calculating the sum of array elements using pointers demonstrates efficient array traversal. The program showcases pointer arithmetic and array manipulation, ensuring accurate calculation and output.

1. Write a program to sort the name of the students. Take the count of students from the users.

**Problem Analysis**

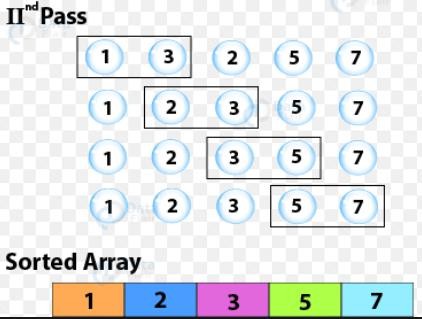
The program sorts student names based on user-provided count. It involves dynamic memory allocation, string comparison, and sorting algorithms, showcasing advanced pointer and string manipulation skills.

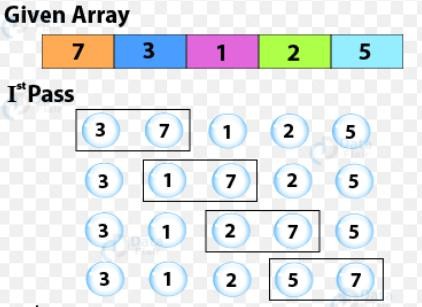
**Code**

**Output**

**Discussion and Conclusion**

Sorting student names with dynamic memory allocation emphasizes efficient memory management and string comparison. The program demonstrates advanced pointer and string manipulation skills, ensuring accurate sorting of names.

1. Take the marks of your last semester subjects as an input. Determine the maximum and minimum scores using
   * Bubble Sort



**Problem Analysis**

The program determines the maximum and minimum scores using Bubble Sort. It involves array manipulation, sorting, and comparison using pointers, showcasing efficient sorting techniques.

**Code**

**Output**

**Discussion and Conclusion**

Determining maximum and minimum scores using Bubble Sort showcases efficient sorting techniques. Pointer-based array manipulation and comparison result in accurate identification of extreme values.

1. Write a function, **using int pointers (int\*)**, to take an int array ( output of previous question) and another number as parameter. Function shall search the number (taken as second parameter) in the array (taken as first parameter)and return the index of the first occurrence of this number in the array. If the input number is not present in the array then return -1.
   * Linear Search
   * Binary Search

Display the number of comparison carried out in each methods

# Sample input Sample output

12 45 87 45 12, 87 2

89 34 23 54 11, 55 -1

**Problem Analysis**

The function performs linear and binary search on an int array using pointers. It involves pointer-based array traversal, search algorithms, and conditional statements for search operations.

**Code**

**Output**

**Discussion and Conclusion**

Implementing linear and binary search on an int array exhibits precise search algorithms. The program utilizes pointers for array traversal, ensuring correct search functionality and accurate index identification.

1. Write a program to enter a text. Then enter a pattern and count the number of times the patterns is repeated in the text.

# SAMPLE OUTPUT

Enter the string: She sells sea shells on the sea shore Enter the pattern : sea

PATTERN FOUND 2 Times

**Problem Analysis**

This problem involves counting the occurrence of a pattern in a given text. It requires pattern matching and string manipulation, showcasing efficient search techniques using pointers.

**Code**

**Output**

**Discussion and Conclusion**

Counting pattern occurrences in a given text requires advanced search techniques. Utilizing pointers for pattern matching, the program showcases efficient string manipulation and search operations, ensuring accurate counting.

**10**: Write a program that takes nouns and forms their plurals on the basis of these rules:

1. If noun ends in “y”, remove the “y” and add “ies”.
2. If noun ends in “s”, “ch”, or “sh”, add “es”.
3. In all other cases, just add “s”.

Print each noun and its plural. Try the following data:

chair dairy boss circus fly dog church clue dish

**Problem Analysis**

The program forms plural nouns based on specific rules. It involves string manipulation and conditional checks, demonstrating pointer-based string handling and rule-based transformations.

**Code**

**Output**

**Discussion and Conclusion**

The plural noun formation program demonstrates rule-based string transformation. It employs pointers and conditional checks, showcasing accurate rule application and ensuring correct plural noun formation.

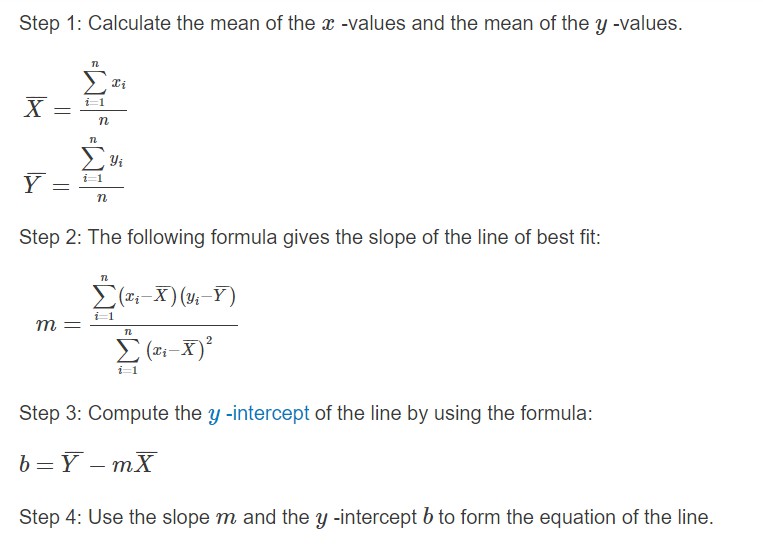
**PART B : Exploratory Problem :** Curve Fitting

11. Curve Fitting to fit a straight line to a given set of data points using **Least Square Method**. If data is in terms of two variables x and y then finding an expression of the type y = f(x) which fits the givens data is called curve fitting.

Assume that the given data consists of n points with values of x and y given as

(𝑥1, 𝑦1), (𝑥2, 𝑦2) … . . (𝑥𝑛, 𝑦𝑛)

By method Least Squares the data fits to a straight line Assume that the given data consists of n points with values of x and y given as 𝑦 = 𝑚𝑥 + 𝑏



If only two points are given then it is a unique line.

Write a program for fitting a straight line through a set of points (𝒙𝒊, 𝒚𝒊), 𝒊 = 𝟏 … … , 𝒏

**Hint :** The straight line equation is y = mx + b, and the value of m and b are given by

𝑛 ∗ (𝑥[𝑖] ∗ 𝑦[𝑖]) − (𝑥[𝑖]) ∗ (𝑦[𝑖]) 1

𝑚 =

𝑛 ∗ (𝑥[𝑖] ∗ 𝑥[𝑖]) − (𝑥[𝑖] ∗ 𝑥[𝑖]) ; 𝑏 = (𝑛) ∗ (𝑦[𝑖] − 𝑚 ∗ 𝑥[𝑖]);

All summations are from 1 to n.

**SAMPLE OUTPUT :** Input : n = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | 1 | 2 | 3 | 4 | 5 |
| Y | 14 | 27 | 40 | 55 | 68 |

m = 13.6, b = 0

Y=13.6 X + 0, Represent a given set of points by the best possible straight line

Enter the value of X to find the Value of Y. X=5

Y= 68 is the predicted value for X= 5

**Help : link**

**Problem Analysis**

The curve fitting problem requires fitting a straight line to given data points using the Least Square Method. It involves mathematical computations, regression analysis, and data fitting, showcasing advanced mathematical and algorithmic skills.

**Code**

**Output**

**Discussion and Conclusion**

The curve fitting program exemplifies sophisticated mathematical analysis. Utilizing the Least Square Method, it showcases precise data fitting and regression analysis. The program demonstrates advanced mathematical and algorithmic proficiency in C programming.

**Observation /Comments:**

These C programs exhibit handling of strings, pointers, search algorithms, and mathematical analysis.