

Objectives:

- Iterative, recursive algorithm, contiguous and non-contiguous allocation

Question 1. A: C++ program to find the factorial of a given number (Recursion)

Example Output:

A Factorial of 5 using Recursion is: 120

Question 1. B: C++ program to find factorial of given number (Iteration)

Example Output

A Factorial of 5 using Iteration is: 120

Question 2. A: Display multiplication table up to 10.

Example Output

Enter an integer: 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

Question 2. B: Display multiplication table up to a given range.

Example Output

Enter an integer: 8

Enter range: 7

8 * 1 = 8

8 * 2 = 16

8 * 3 = 24

8 * 4 = 32

$$8 * 5 = 40$$

$$8 * 6 = 48$$

$$8 * 7 = 56$$

Question 3.: Write a program that will include numbers from zero to n. The program will be carried out in the following steps:

- Use arraylist for store the data (contiguous allocation)
- Use linked list for store the data (non-contiguous allocation)
- And show each representations (arraylist and linked list) data and addresses.
- Repeat all implementations for different data types (double, int, etc.)

Hint for non-contiguous allocation !: <https://www.geeksforgeeks.org/list-cpp-stl/>

Example Output

(contiguous allocation for char)

X[0] = A, Address = 0x7ffdeaa78040

X[1] = B, Address = 0x7ffdeaa78041

X[2] = C, Address = 0x7ffdeaa78042

X[3] = D, Address = 0x7ffdeaa78043

X[4] = E, Address = 0x7ffdeaa78044

Question 4.: Write a program to show contiguous allocation on 2D arrays. The program will be carried out in the following steps:

- Define a 2-dimensional integer array.
- Get row and column information from the user.
- Print the address information of the elements in the array to the screen.
- Print the total amount of memory used on the screen.

Example Output (if user enters 3 for row and column)

&X[0][0] = 100, Address = 0x7ffdeaa78040

&X[0][1] = 104, Address = 0x7ffdeaa78044

&X[0][2] = 108, Address = 0x7ffdeaa78048

&X[1][0] = 112, Address = 0x7ffdeaa7804c

&X[1][1] = 116, Address = 0x7ffdeaa78050

&X[1][2] = 120, Address = 0x7ffdeaa78054

&X[2][0] = 124, Address = 0x7ffdeaa78058

&X[2][1] = 128, Address = 0x7ffdeaa7805c

&X[2][2] = 132, Address = 0x7ffdeaa78060