

1. **Homework-1:** Write a program that displays the square of integers 1 to 5.

Use \t to line up the columns.

The output should read

N	Square
1	1
2	4
3	9
4	16

2. **Homework-2:** Write a program that computes the Nth sum of the geometric series.

The output should read

*****This program that computes the Nth sum of the geometric series*****

Please enter the N : 3

The sum is : 1.75

3. **Homework-3:** Write a program that takes in two numbers x and y and an unsigned int n and computes $(x + y)^n$ using the binomial theorem. Please verify the theorem by displaying the value $(x + y)^n$ obtained by a direct evaluation.

4. **Homework-4:** Write a program calculating the roots of the quadratic equation

$$ax^2 + bx + c = 0.$$

Get a user input of the coefficients a, b, c and output the **real** roots if any else display a message which informs the user that there are no real roots for this quadratic equation.

5. **Homework-5:** Write a program that inputs a matrix of order 3×3 and outputs its determinant. The user should be given a warning in case the matrix is singular. (That is, in case the matrix has zero determinant).

6. **Homework-6:** Write a program that fills out a 3×5 matrix A with the following display of entries:

		1	2	3	4	5
A	=	2	3	4	5	6
		3	4	5	6	7

Note: This task should be completed without hard coding the entries rather using the for loops.

7. **Homework-7:** Write a program that arranges the randomly generated array of length 3 and sorts the entries in decreasing order.

8. **Homework-8:** Write a program that does the above task for a randomly generated array of length n .

9. **Homework 9** : Consider the following output which compares
 -swap of variable value via **reassignment of variable value** versus,
 -swap of **variable address**.

```
int main()
{
  int x =4;
  int y =5;
  int *ptr_x, *ptr_y;
  ptr_x = &x;
  ptr_y = &y;
  cout <<" The initial values of ptr_x="
    <<*ptr_x<< " and ptr_y=" << *ptr_y << endl;
  cout << "*****Pass through value*****"<< endl;
  cout <<" The values of x = " <<x<< " and y = " << y << endl;
  *ptr_x =*ptr_y; // reassigned a value of x.
  cout <<" Assignment of x's value via x=y, x = "
    <<*ptr_x<< " while y = " << *ptr_y << endl;

  //assignment of ptr_x to ptr_y.
  x =4, y=5;
  cout <<" The reset values of ptr_x="
    <<*ptr_x<< " and ptr_y=" << *ptr_y << endl;
  ptr_x =ptr_y ;// reassignment of the address.

  cout <<" Reassignment of pointer_x via ptr_x=ptr_y, the values of x=" <<*ptr_x<<
    " and y=" << *ptr_y << endl;
  return 0;
}
```

- (a) Explain why in this case, it doesn't matter whether one is assigning value of x (to value of y) or if one is assigning the address of x (to value of y). You may use a diagram to explain this.
- (b) **(The subtlety of address assignment)** In the program above, please explain the output if we additionally have the following declaration:
 *ptr_x = -4 ;
 I. What are the values of *ptr_x, *ptr_y, x and y in this case?
 II. Can you explain the assignment especially the final value of y/ptr_y?

10. **Homework 10 (a):**

Read [guide_hwk_10.cpp](#) carefully. It calculates the factorial of an integer via a function (with argument as pointer rather than value).

With this example in mind, please write a program that calls a function **sum** with argument as pointer rather than value (as we have previously done so far) for the calculating the **N**th sum of the geometric series (you can use your old code from homework-2.)

Homework 10 (b): Write a program that generates an array call it **array** of length **n**. (**n** here should be a user defined input and even numbered if it is not even numbered, please prompt the user to enter another array length until the user inputs an even length array).

Fill out the array with entries { 0, 1, 2, 3, ..., n-1 }.

Access the even and odd indexed entries through pointers.

Print the even and odd indexed entries **strictly** with the use of pointers.

The output should look something similar to:

Please enter the array size (Please make sure it is even) : 4

The array is { 0, 1, 2, 3 }

Even indexed array entries = 0, odd indexed array entries = 1

Even indexed array entries = 2, odd indexed array entries = 3

I run this program again and this time, I enter an odd number:

Please enter the array size (Please make sure it is even) : 67

Please make sure the size is an even number ! Try again.

Please enter the array size (Please make sure it is even) : 6

(suppose 6 is entered by the user)

Given array = { 0 1 2 3 4 5 }

Even indexed array entries = 0, odd indexed array entries = 1

Even indexed array entries = 2, odd indexed array entries = 3

Even indexed array entries = 4, odd indexed array entries = 5

Homework 11: Write a program that extends the existing matrix structure we discussed in class to include the member functions trace and the transpose of the matrix.

You may refer to this [link](#) to understand structures better.

Homework 12: Write a program that introduces the polynomial structure and performs the tasks of point evaluation, first second and third derivative computation.

Homework 13: We discussed the code on Math Modeling. In the cpp file added ode_solver_rhs.cpp, we have included the rhs function **F**.

The homework is to implement the following algorithm.

Given an initialization of the first row: $u[0][]$,

for $t=0,1,2, \dots,10$.

```
{
    compute for  $j=0,1,2,\dots,7$ .
    {
         $u[t+1][j] = u[t][j] + \text{delta}_t * F[t][j]$ 
    } //calculation of concentration levels at time t+1.
```

}//calculation for the time step t.

Homework 14: The above was computed over a span of 10 days and data recorded each day. Please change the number of days to be 100 days and the data recorded twice a day. How will the time-step Δt from the previous program to this one ?