Question 1:

Exercise 1:

- Create a class Date having following private data members:

Int Day

Int Month

Int Year

- Create an object of Date “date1” and run your program

Exercise 2 [Default Constructor]:

- Write a default Constructor of Date that initializes the object to 1st January 1926 i.e., (1, 1, 1926) and prints.

“Default Constructor Called” in start

- Now run your program and test what does date1 prints?

Exercise 3 [Print Function]:

- Implement a function Print in Date class which prints a date in following format

dd/mm/yyyy (e.g. 1/1/1926 for date1)

- Print object date1 in your main function and run the program.

- What does it print and how can we initialize the data of date1 at the time of creation?

Exercise 4 [Overloaded Constructor with Default Argument]:

- Write an overloaded Constructor of Date class that initializes the date object to date, month and year provided as parameter and prints “Overloaded Function Called”

- Now create another object independanceDay in main that is 14/08/1947

- Print independanceDay by calling Print function of Date class and run your program

Exercise 5 [Input Function]:

- Write a function Input in your Date class that takes input from user to populate a Date object

- User will enter his/her date of Birth in date1 object.

- Call “date1.Input()” and “date1.Print()” in your driver program and test it

Exercise 6 [Destructor]:

- Write Destructor of Date class that prints “Destructor called”
- Run your program and test it

Exercise 7 [Setters]:

- Create an object xmasDay using default constructor
- Print xmasDay and see what it prints
- Write Setters i.e. SetDay, SetMonth and SetYear in your class
- Now set xmasDay to 25/12/2018 using Setters in main

Exercise 8 [Getters]:

- Write Getters i.e. GetDay, GetMonth and GetYear in your date class
- Now print xmasDay using Getters in your Driver program

Exercise 9 [Built-in Assignment Operator]:

- Create an object temp of Date class
- Assign value of xmasDay to temp
- Print temp and test your program

Exercise 10 [Passing object by value]:

- Write a function int Compare(Date) that compares two dates, returns 1 if left hand side object is greater than right hand side object, -1 if lhs is smaller and 0 otherwise
- Test your function

Exercise 11 [Return object by value]:

- Write a function Date IncrementMonth() that returns a newly created Date object with one month next to the current date object. For example, if date1 = 2/01/2016 date1.IncrementMonth() will return 2/02/2016 without changing date1
- Print both the date1 and newly created date in your driver program to test the result

Exercise 12 [Create a destructor]: print the message “object deleted” with the information of day/month and year of object. Syntax of destructor ~className(){}

Q#2: Create a static 2D array of integers “data” of (N rows and M columns). You can hard code the data in the array. Some cells in the array contain the value 0, representing "holes" where data has been removed, causing the array to be scattered. Your task is to clean up the data by removing the cells with value 0, thereby compacting the array and making it more memory-efficient. The goal is to eliminate the gaps (zero-valued cells) while preserving the integrity of the non-zero data. There must not be any memory leak in your program. The “**data**” pointer received in the parameters list should point to the newly created 2D dynamic array and the row count should also be updated. Print the elements of 2D array after compaction.

Current Status

31	0	19	0	0	25
0	0	20	0	30	0
6	0	0	8	0	5
0	0	0	0	0	0
42	0	0	35	0	0

Expected Outcome

31	19	25	-1
20	30	-1	
6	8	5	-1
42	35	-1	

void compactArray(int &data, int & N, int M){//just complete this function. Don’t write any header**

Q#3: Write a C++ program that performs the following operations on a character array:
You are given a static 1D character array. Your task is to extract the information of email services used by the people and store the information in a dynamic 2D character array “**services**”.
You can hard code the given string. Check the sample output to understand what is required in this task.

Hard code this input.
char data[] = “I tried to reach Sarah at her **@gmail.com** address, but she prefers using her **@icloud.com** for more privacy, while John sent his update from his **@outlook.com** account and Jane replied from her **@yahoo.com** email.”

services➔

0	➔	gmail\0
1	➔	icloud\0
2	➔	outlook\0
3	➔	yahoo\0

Hint! You need to count the occurrences of @ sign in the input string. Now create an array of (pointer to character) of size equal to the count of @ sign which is 4 in this scenario. Do not create the array of pointers of size 4 directly. Your program should be generic. Any number of @ sign may exists in the input string. Once the array of (pointers to characters) is created now search @ sign again. Once @ sign is encountered you need to count the characters after between @ sign and dot (.) sign. Allocate the memory one greater than the count to store null character. Store the email service name after memory allocation and store null character at the end. Repeat the process for remaining @ signs.

Q#4: Create a 2D static array of N rows and M columns, hard code the data in the matrix. Now store the transpose of matrix in a separate 2D dynamic array. Print the matrix and transpose on console. Delete the memory and make the double pointer (null). (for practice: Now you have 2 matrices of dimensions N*M and M*N. Calculate the product of matrices (matrix multiplication) and store the result in 2D dynamic array “res”.)