

WEEK – 1

PROBLEM

1# Open a new document and type the following letter.

July 15, 2020,
Chennai

From
VENKATESH.P
Sri Ranga Apartments,
No: 120, II Avenue,
T. Nagar. Chennai-17

To
<<Name>>
<<Address>>

Respected <<Name>>
With the current slowdown in hiring within the high-tech field, you must be flooded with resumes from out-placed software engineers such as me. Please take a moment to consider my qualifications. I believe in particular is highly marketable in this tight market.
I worked on the team that pioneered the technology that put the Palm Pilot on the map.
In today's increasingly mobile society, this technology has places to go, and I have ideas that could take us to the next step in office independence.
Please call me with prospective job opportunities. I am interested in a project management position in the Rs. 9K range.

Thank you!
Venkatesh.

Enclosure: Resume

Format of Resume

Name	:
Father's Name	:
Date of Birth	:
Age	:
Address	:
Educational Qualification	:

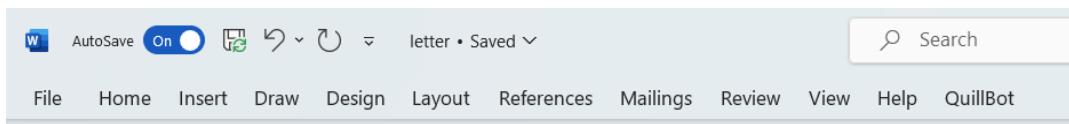
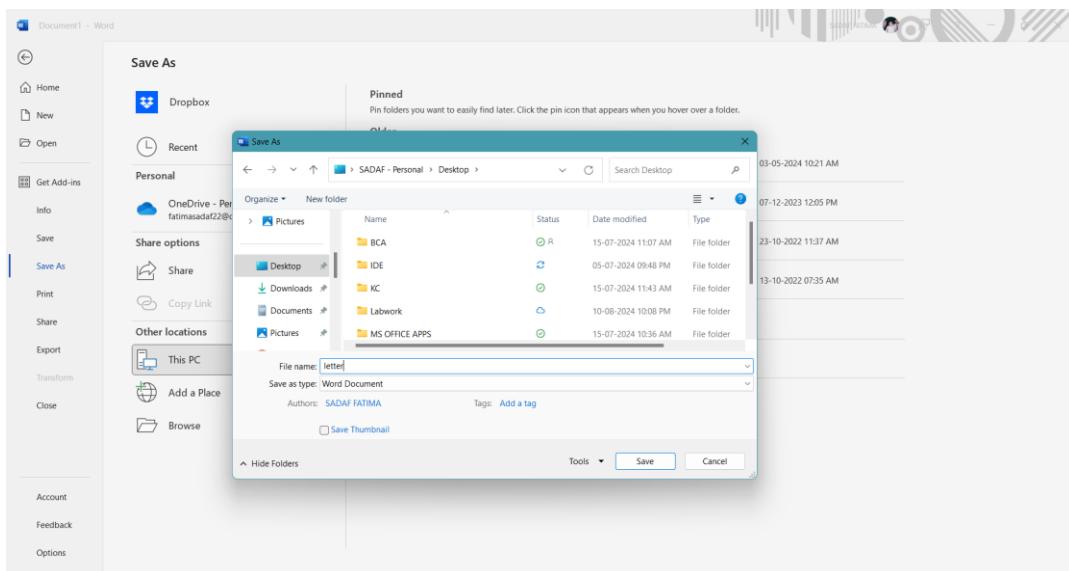
Sr.	Qualification	Board/ University	Percentage

Work Experience	:
Technical Skills	:
Personal Skills	:
Hobbies	:

Dated: _____

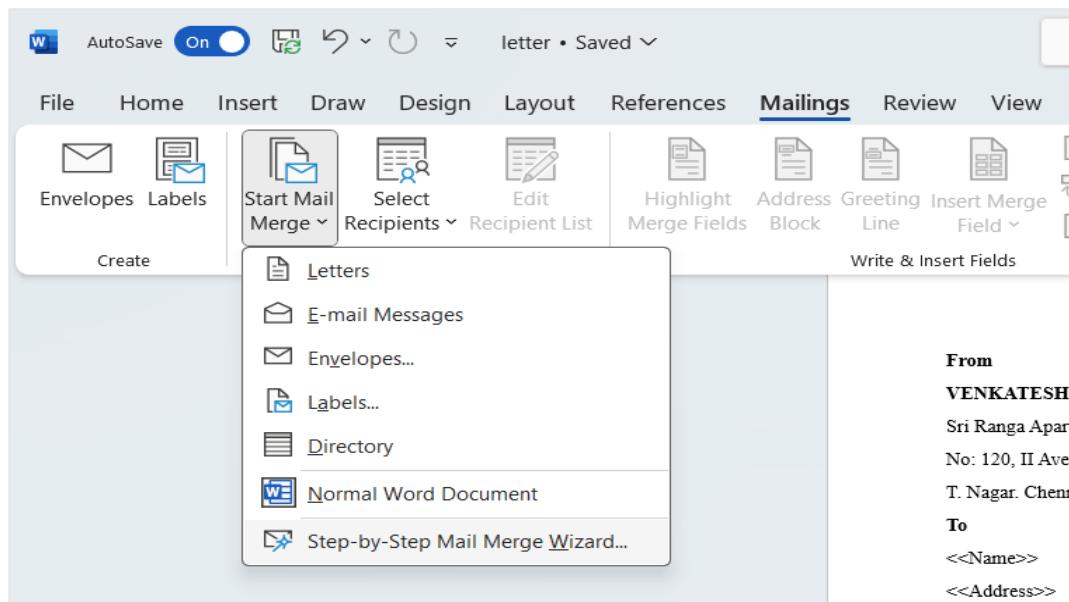
Signature _____

i. Save the document as “Letter.doc”

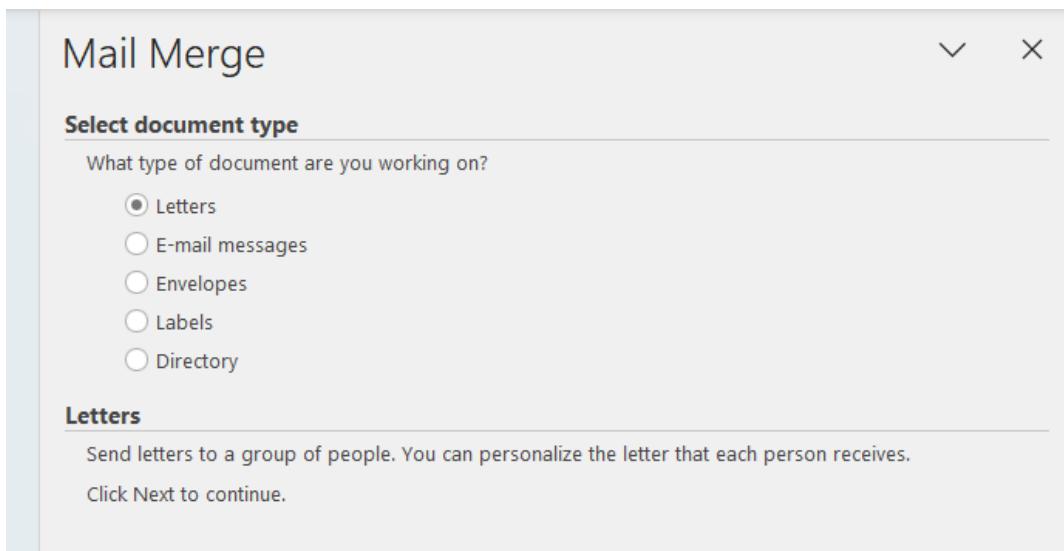


ii. Send the document to 3 recipients using Mail merge. (Use three different addresses)

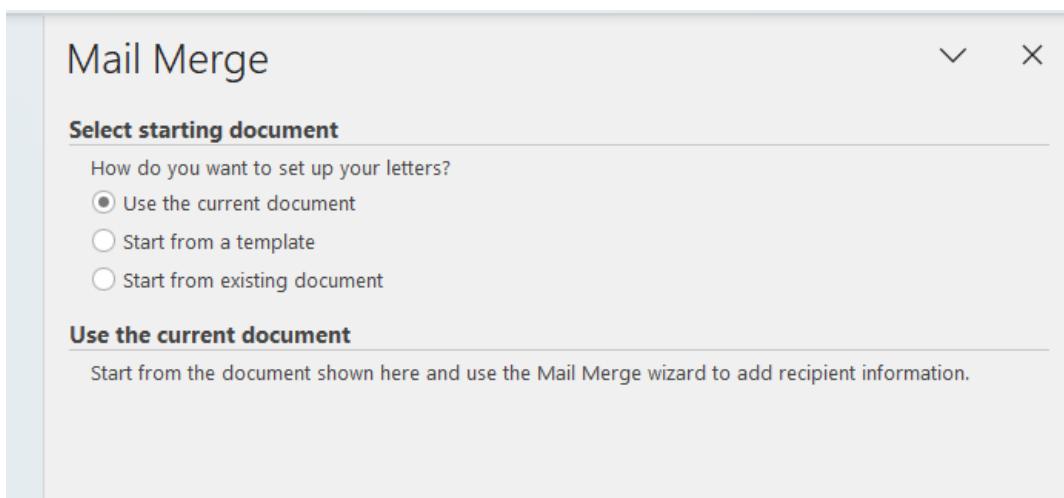
Step 1: Click on the **Mailings** tab in the ribbon and Click on **Start Mail Merge > Step-by-Step Mail Merge Wizard**.



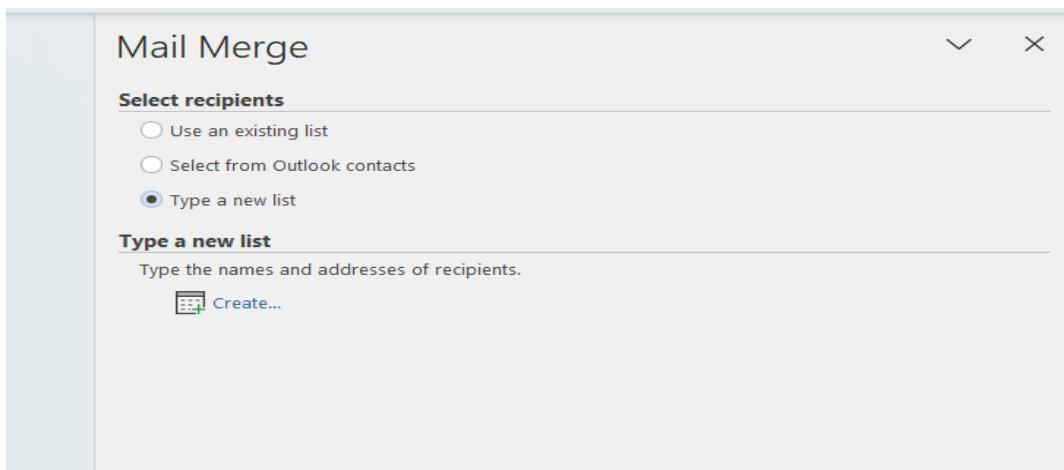
Step 2: In Step 1 of the wizard, select **Letters**. Click **Next: Starting Documents**.



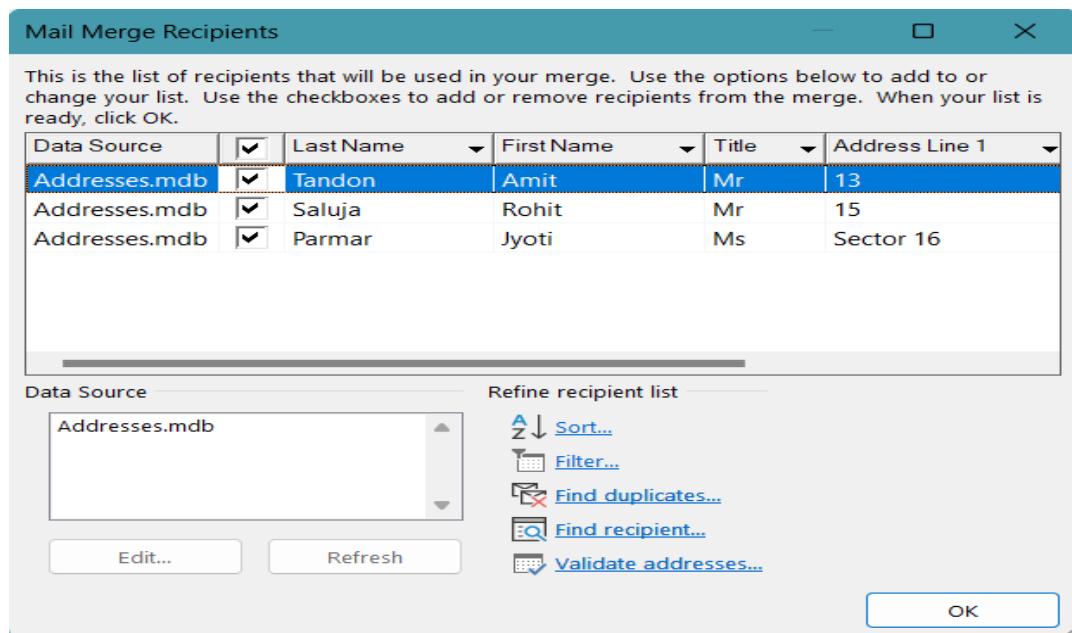
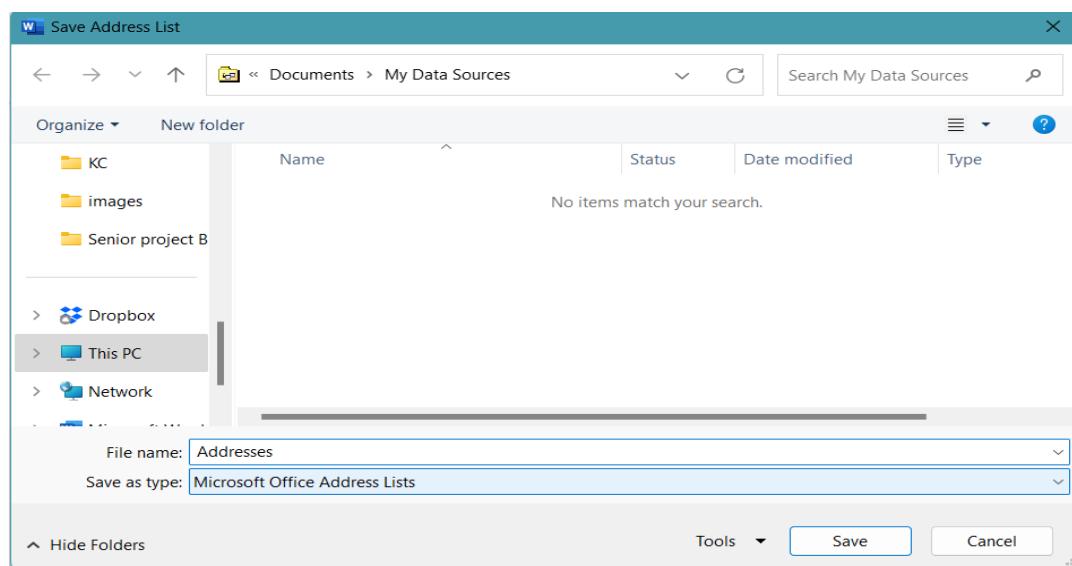
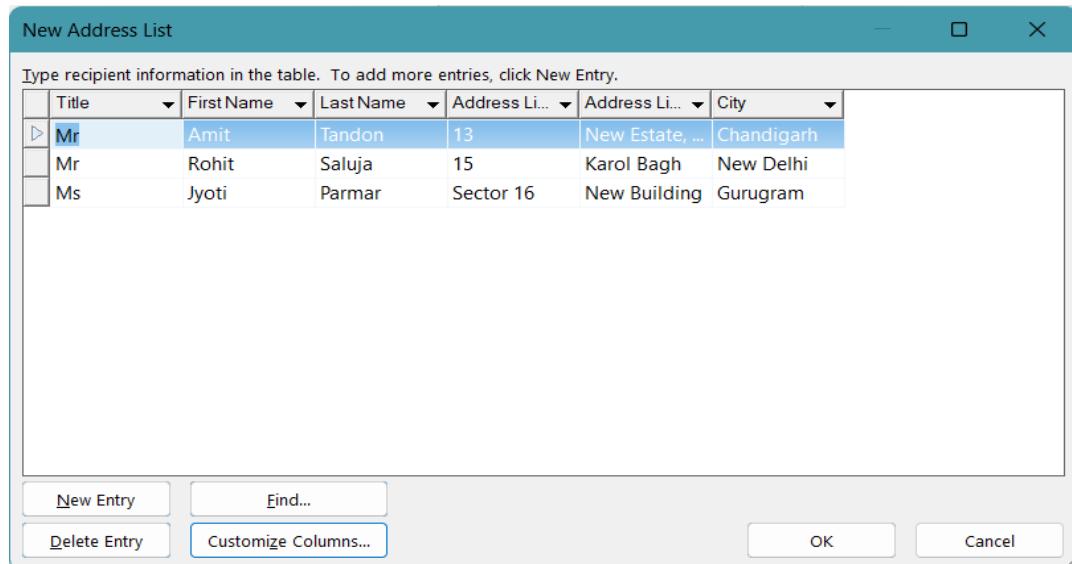
Step 3: In Step 2, choose **Use the current document** if you have your letter ready. Click **Next: Select recipients**.



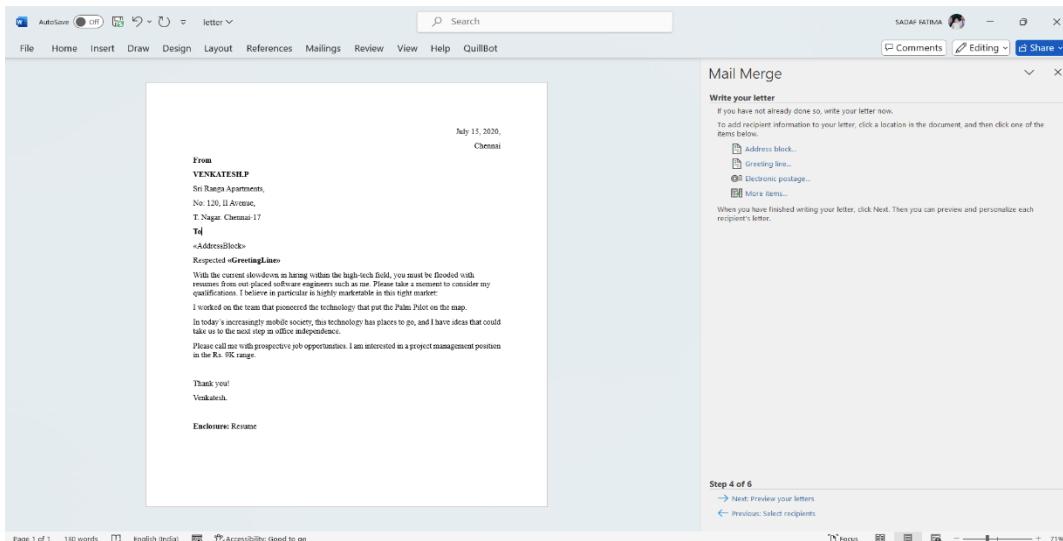
Step 4: In Step 3 of the wizard, choose **Type a new list**. Click **Create**.



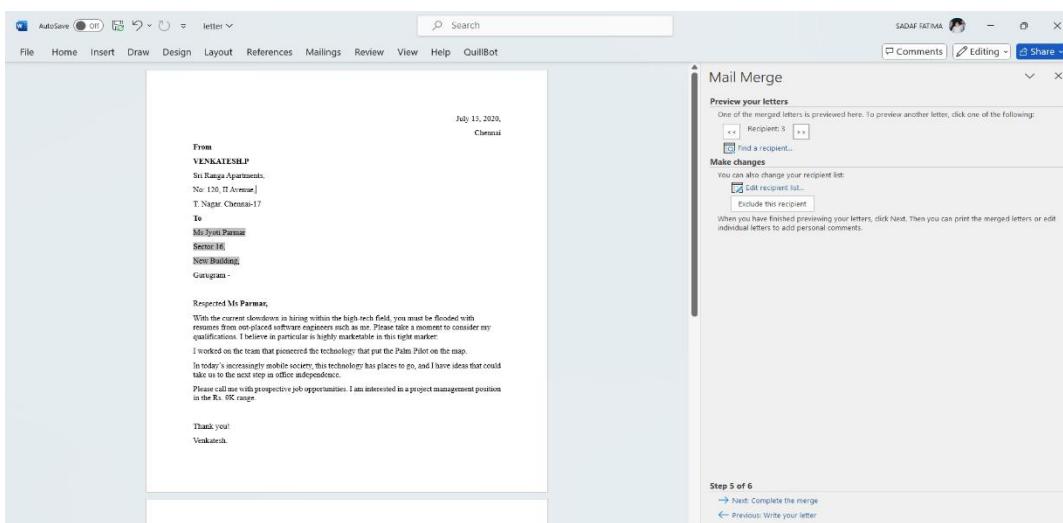
Create new address list and save address list.



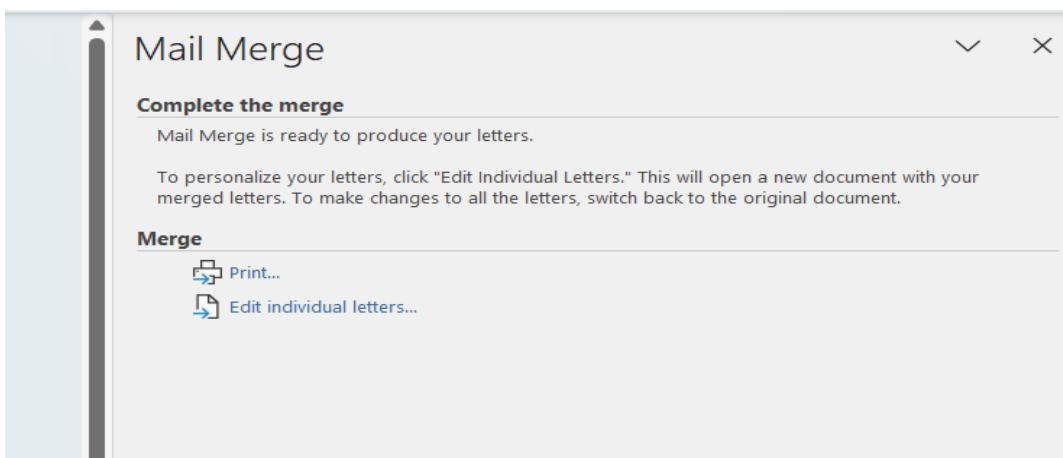
Step 5: Click on **Insert Merge Field** from the wizard and select the field you want to insert. Click **Next: Preview your letters**.



Step 5: The wizard will show how each recipient's letter will look. Click **Next: Complete the merge**.

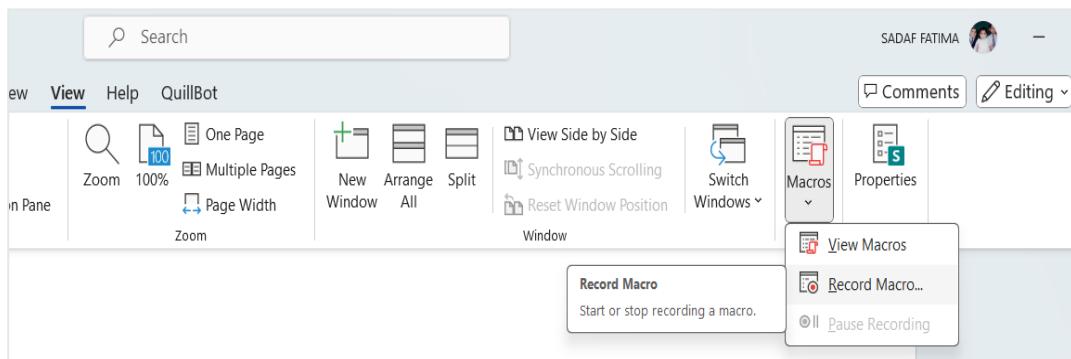


Step 5: In the final step, choose how you want to finish the process.

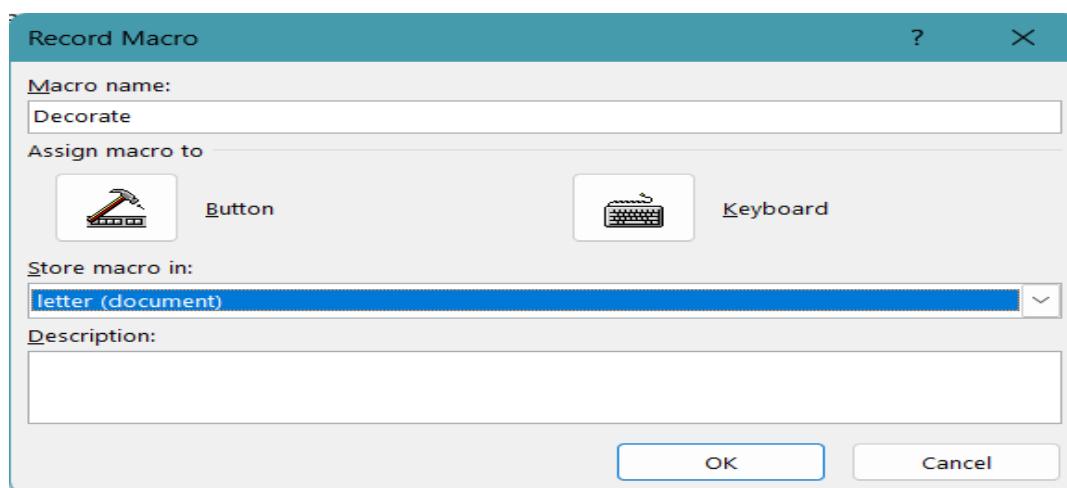


- iii. Define a Macro 'Decorate' which makes the text bold, Red in color and italic, font size. Assign a shortcut key Alt + Z to this macro.

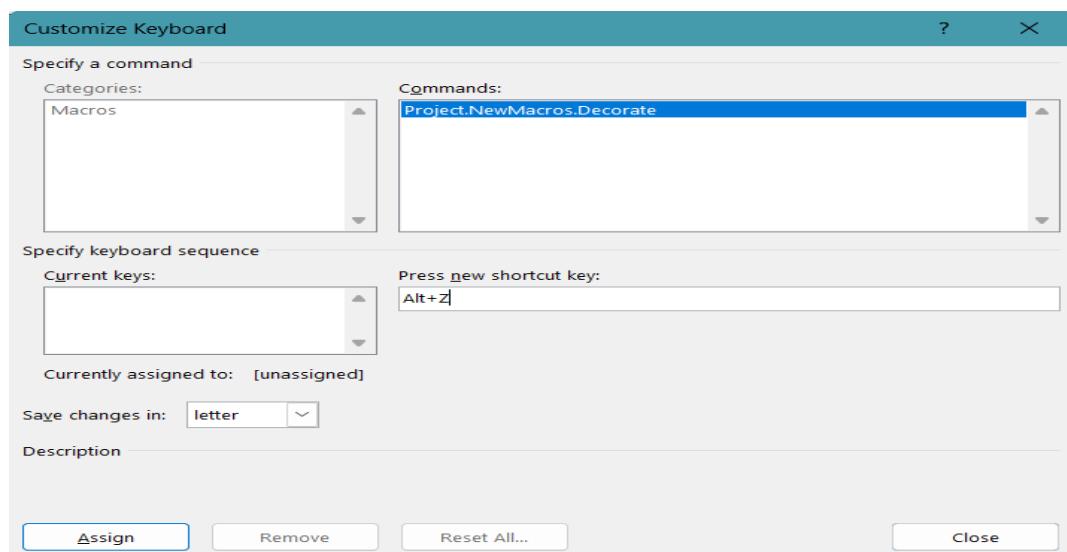
Step 1: Click on the **View** tab in the ribbon. Click on **Macros > Record Macro**.



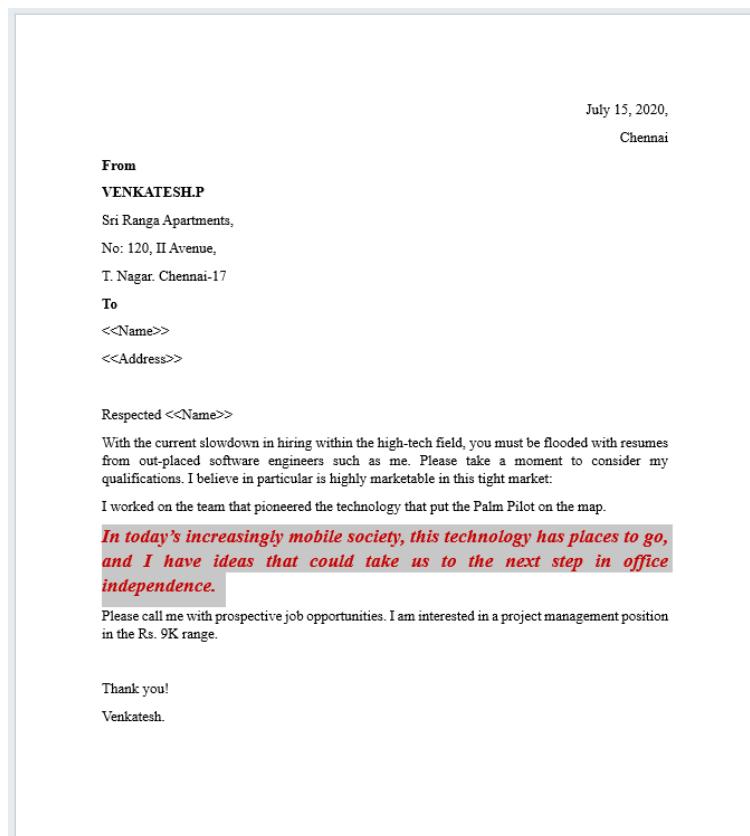
Step 2: In the **Record Macro** dialog box, type Decorate in the **Macro name** field. Click on the **Keyboard** button



Step 3: In the **Customize Keyboard** dialog, click in the **Press new shortcut key** field. Press Alt + Z on your keyboard. Click **Assign** and then click **Close** to return to the Record Macro dialog box.



Step 3: **Format the Text** with bold, red color, italic, and a specific font size. **Stop Recording** and use the macro with the assigned shortcut.



iv. Close the document.



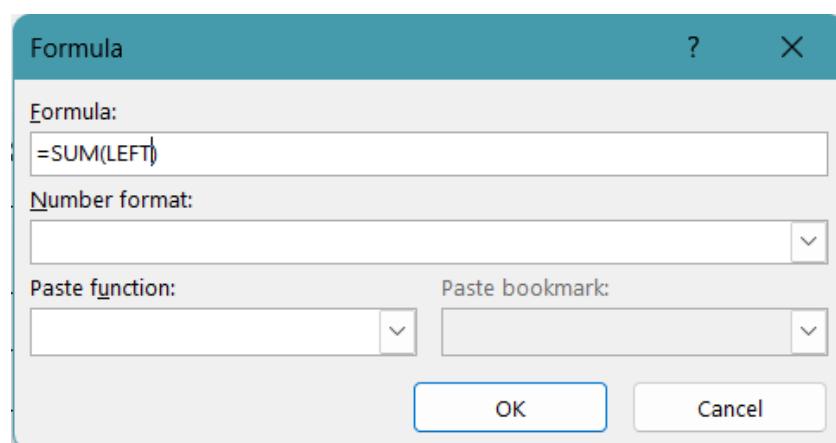
v. The sample addresses are:

- i. Mr. Amit Tandon
13, New Estate,
Ring Road, Chandigarh
- ii. Mr. Rohit Saluja
15, Karol Bagh, New Delhi
- iii. Mrs. Jyoti Parmar
Sector 16, New Building, Gurugram

2# Create a table in word as shown below:

Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
1	Sakshi	80	70	
2	Rohit	70	80	
3	Amit	60	50	
4	Rakesh	40	60	
5	Komal	30	70	
6	Garima	80	80	

- i. In the total marks' column, entries should be calculated using formulas and it is the sum of marks in physics and marks in chemistry.

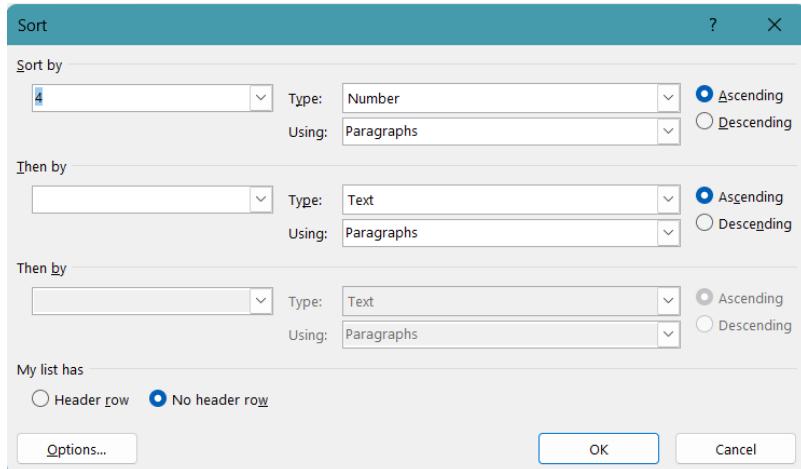


Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
1	Sakshi	80	70	150
2	Rohit	70	80	150
3	Amit	60	50	110
4	Rakesh	40	60	100
5	Komal	30	70	100
6	Garima	80	80	160

- ii. Insert a new row at the end of the table and also find grand total using formula.

Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
1	Sakshi	80	70	150
2	Rohit	70	80	150
3	Amit	60	50	110
4	Rakesh	40	60	100
5	Komal	30	70	100
6	Garima	80	80	160
Grand Total				770

- iii. Sort the table based on total marks.



Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
4	Rakesh	40	60	100
5	Komal	30	70	100
3	Amit	60	50	110
1	Sakshi	80	70	150
2	Rohit	70	80	150
6	Garima	80	80	160
Grand Total				770

- iv. The date and heading should be centre aligned.

Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
4	Rakesh	40	60	100
5	Komal	30	70	100
3	Amit	60	50	110
1	Sakshi	80	70	150
2	Rohit	70	80	150
6	Garima	80	80	160
Grand Total				770

- v. Heading should be in bold and underlined.

<u>Roll No</u>	<u>Name</u>	<u>Marks in Physics</u>	<u>Marks in Chemistry</u>	<u>Total Marks</u>
4	Rakesh	40	60	100
5	Komal	30	70	100
3	Amit	60	50	110
1	Sakshi	80	70	150
2	Rohit	70	80	150
6	Garima	80	80	160
Grand Total				770

3# Using a spreadsheet package you have studied, construct T Morongo's pay slip for December 2016 following the instructions below. Insert a custom footer with your *name, subject, course, exam/Test &question number*. Save it as Salary advice.

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary		Pension @ 8%		
Housing Subsidy		P.A.Y.E		
Vehicle Allowance		U.I.F.		
		Medical Aid		
		Bond Repayment		
Gross Income		Total deductions		
Net Salary				

Instructions:

- i) Housing Subsidy 6000.00 per year.

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary	2600	Pension @ 8%		208
Housing Subsidy	500	P.A.Y.E		
Vehicle Allowance		U.I.F.		
		Medical Aid		
		Bond Repayment		
Gross Income		Total deductions		
Net Salary				

- ii) Car Allowance 100.00 per month

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary	2600	Pension @ 8%		208
Housing Subsidy	500	P.A.Y.E		
Vehicle Allowance	100	U.I.F.		
		Medical Aid		
		Bond Repayment		
Gross Income		Total deductions		
Net Salary				

iii) Pension 8% on Basic Salary.

B8			
A	B	C	D
SALARY ADVICE FOR MARCH 2016			
2 EMPLOYEE	T MARONGO		
3 STAFF NO		4	
4 DATE		31-Mar-16	
5 NEXT PAY DATE		30-Apr-16	
6 BASIC SALARY p.a.		31200	
7 INCOME	AMOUNT	DEDUCTIONS	AMOUNT
8 Basic Salary	2600	Pension @ 8%	
9 Housing Subsidy		P.A.Y.E	
10 Vehicle Allowance		U.I.F.	
11		Medical Aid	
12		Bond Repayment	
13 Gross Income		Total deductions	
14 Net Salary			
15			

D8			
A	B	C	D
SALARY ADVICE FOR MARCH 2016			
2 EMPLOYEE	T MARONGO		
3 STAFF NO		4	
4 DATE		31-Mar-16	
5 NEXT PAY DATE		30-Apr-16	
6 BASIC SALARY p.a.		31200	
7 INCOME	AMOUNT	DEDUCTIONS	AMOUNT
8 Basic Salary	2600	Pension @ 8%	208
9 Housing Subsidy		P.A.Y.E	
10 Vehicle Allowance		U.I.F.	
11		Medical Aid	
12		Bond Repayment	
13 Gross Income		Total deductions	
14 Net Salary			
15			

iv) PAYE 636.83

D9			
A	B	C	D
SALARY ADVICE FOR MARCH 2016			
2 EMPLOYEE	T MARONGO		
3 STAFF NO		4	
4 DATE		31-Mar-16	
5 NEXT PAY DATE		30-Apr-16	
6 BASIC SALARY p.a.		31200	
7 INCOME	AMOUNT	DEDUCTIONS	AMOUNT
8 Basic Salary	2600	Pension @ 8%	208
9 Housing Subsidy	500	P.A.Y.E	636.83
10 Vehicle Allowance	100	U.I.F.	
11		Medical Aid	
12		Bond Repayment	
13 Gross Income		Total deductions	
14 Net Salary			
15			

v) Medical Aid 70.00

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	4		
DATE	31-Mar-16		
NEXT PAY DATE	30-Apr-16		
BASIC SALARY p.a.	31200		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary	2600	Pension @ 8%	208
Housing Subsidy	500	P.A.Y.E	636.83
Vehicle Allowance	100	U.I.F.	
		Medical Aid	70
		Bond Repayment	
Gross Income		Total deductions	
Net Salary			
15			

vi) U.I.F. 1% on Basic Salary + Housing Subsidy

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	4		
DATE	31-Mar-16		
NEXT PAY DATE	30-Apr-16		
BASIC SALARY p.a.	31200		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary	2600	Pension @ 8%	208
Housing Subsidy	500	P.A.Y.E	636.83
Vehicle Allowance	100	U.I.F.	526
		Medical Aid	70
		Bond Repayment	
Gross Income		Total deductions	
Net Salary			
15			

vii) Bond Repayment 630.00

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	4		
DATE	31-Mar-16		
NEXT PAY DATE	30-Apr-16		
BASIC SALARY p.a.	31200		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary	2600	Pension @ 8%	208
Housing Subsidy	500	P.A.Y.E	636.83
Vehicle Allowance	100	U.I.F.	526
		Medical Aid	70
		Bond Repayment	630
Gross Income		Total deductions	
Net Salary			
14			

viii) Calculate Net Salary.

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary	2600	Pension @ 8%	208	
Housing Subsidy	500	P.A.Y.E	636.83	
Vehicle Allowance	100	U.I.F.	526	
		Medical Aid	70	
		Bond Repayment	630	
13 Gross Income	3200	Total deductions		
14 Net Salary				

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary	2600	Pension @ 8%	208	
Housing Subsidy	500	P.A.Y.E	636.83	
Vehicle Allowance	100	U.I.F.	526	
		Medical Aid	70	
		Bond Repayment	630	
13 Gross Income	3200	Total deductions		2070.83
14 Net Salary				

SALARY ADVICE FOR MARCH 2016				
EMPLOYEE	T MARONGO			
STAFF NO	4			
DATE	31-Mar-16			
NEXT PAY DATE	30-Apr-16			
BASIC SALARY p.a.	31200			
INCOME	AMOUNT	DEDUCTIONS	AMOUNT	
Basic Salary	2600	Pension @ 8%	208	
Housing Subsidy	500	P.A.Y.E	636.83	
Vehicle Allowance	100	U.I.F.	526	
		Medical Aid	70	
		Bond Repayment	630	
13 Gross Income	3200	Total deductions		2070.83
14 Net Salary	1129.17			
15				

ix) Format all figures to two decimal places and insert ₹ currency symbol.

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	4		
DATE	31-Mar-16		
NEXT PAY DATE	30-Apr-16		
BASIC SALARY p.a.	₹ 31,200.00		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary	₹ 2,600.00	Pension @ 8%	₹ 208.00
Housing Subsidy	₹ 500.00	P.A.Y.E	₹ 636.83
Vehicle Allowance	₹ 100.00	U.I.F.	₹ 526.00
		Medical Aid	₹ 70.00
		Bond Repayment	₹ 630.00
Gross Income	₹ 3,200.00	Total deductions	₹ 2,070.83
Net Salary	₹ 1,129.17		

x) Insert a custom footer with your name, subject, and question number. Save it as salary advice2.

2	4	6	8	10	12	14	16	18	20	22	24
A	B	C	D	E	F	G	H	I			
SALARY ADVICE FOR MARCH 2016											
1	EMPLOYEE	T MARONGO									
2	STAFF NO	4									
3	DATE	31-Mar-16									
4	NEXT PAY DATE	30-Apr-16									
5	BASIC SALARY p.a.	₹ 31,200.00									
6	INCOME	AMOUNT	DEDUCTIONS	AMOUNT							
7	Basic Salary	₹ 2,600.00	Pension @ 8%	₹ 208.00							
8	Housing Subsidy	₹ 500.00	P.A.Y.E	₹ 636.83							
9	Vehicle Allowance	₹ 100.00	U.I.F.	₹ 526.00							
10			Medical Aid	₹ 70.00							
11			Bond Repayment	₹ 630.00							
12	Gross Income	₹ 3,200.00	Total deductions	₹ 2,070.83							
13	Net Salary	₹ 1,129.17									
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											
38											
39											
40											
41											
42											

SADAF FATIMA, PRACTICAL LAB-I, Q3

4# Use a new workbook & construct a worksheet with the data given & save it as LYONS.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75			234			
5	Quench	3.65			345			
6	Xtra	4.25			456			
7	Sun splash	1.50			123			
8	House brand	1.50			245			
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

Instructions

- **Markup** = Cost price/Litre * 35%
 - **Selling price** = Cost price/Litre + Mark up
 - **Total income** = Litres sold * Selling Price
 - **Profit** = Total income – (Cost price/Litre * Litres sold)
- i. The MARKUP % (35%) must be inserted in a separate cell under the heading. USE IT as an absolute cell reference in the formula to calculate the mark up per item.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75			234			
5	Quench	3.65			345			
6	Xtra	4.25			456			
7	Sun splash	1.50			123			
8	House brand	1.50			245			
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- ii. Calculate the mark up for each item.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31		234			35.00%
5	Quench	3.65	1.28		345			
6	Xtra	4.25	1.49		456			
7	Sun splash	1.50	0.53		123			
8	House brand	1.50	0.53		245			
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- iii. Calculate the selling price for each item.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234			35.00%
5	Quench	3.65	1.28	4.93	345			
6	Xtra	4.25	1.49	5.74	456			
7	Sun splash	1.50	0.53	2.03	123			
8	House brand	1.50	0.53	2.03	245			
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- iv. Calculate the Total Income for each item.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234	1184.63		35.00%
5	Quench	3.65	1.28	4.93	345	1699.99		
6	Xtra	4.25	1.49	5.74	456	2616.30		
7	Sun splash	1.50	0.53	2.03	123	249.08		
8	House brand	1.50	0.53	2.03	245	496.13		
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- v. Calculate the profit for each item.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13	35.00%
5	Quench	3.65	1.28	4.93	345	1699.99	440.74	
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30	
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58	
8	House brand	1.50	0.53	2.03	245	496.13	128.63	
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- vi. Format the column LITRES SOLD to display the number of litres as integers. The rest of the worksheet must be formatted to display two decimals.

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13	35.00%
5	Quench	3.65	1.28	4.93	345	1699.99	440.74	
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30	
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58	
8	House brand	1.50	0.53	2.03	245	496.13	128.63	
9	TOTAL							
10	HIGHEST							
11	LOWEST							
12	AVERAGE							

- vii. Use statistical functions to calculate the:

- AVERAGE

	A	B	C	D	E	F	G	H
1	LYSONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13	35.00%
5	Quench	3.65	1.28	4.93	345	1699.99	440.74	
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30	
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58	
8	House brand	1.50	0.53	2.03	245	496.13	128.63	
9	TOTAL	14.65	5.13	19.78	1403	6246.11	1619.36	
10	HIGHEST	4.25	1.49	5.74	456	2616.30	678.30	
11	LOWEST	1.50	0.53	2.03	123	249.08	64.58	
12	AVERAGE	2.93	1.03	3.96	280.60	1249.22	323.87	

- HIGHEST (MAX.)

B10 : =MAX(B4:B8)

A	B	C	D	E	F	G	H
1	LYSONS INC						
2	Orange JUICE sales						
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13
5	Quench	3.65	1.28	4.93	345	1699.99	440.74
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58
8	House brand	1.50	0.53	2.03	245	496.13	128.63
9	TOTAL	14.65	5.13	19.78	1403	6246.11	1619.36
10	HIGHEST	4.25	1.49	5.74	456	2616.30	678.30
11	LOWEST						
12	AVERAGE						

- LOWEST (MIN) for Selling Price column up to Profit Column.

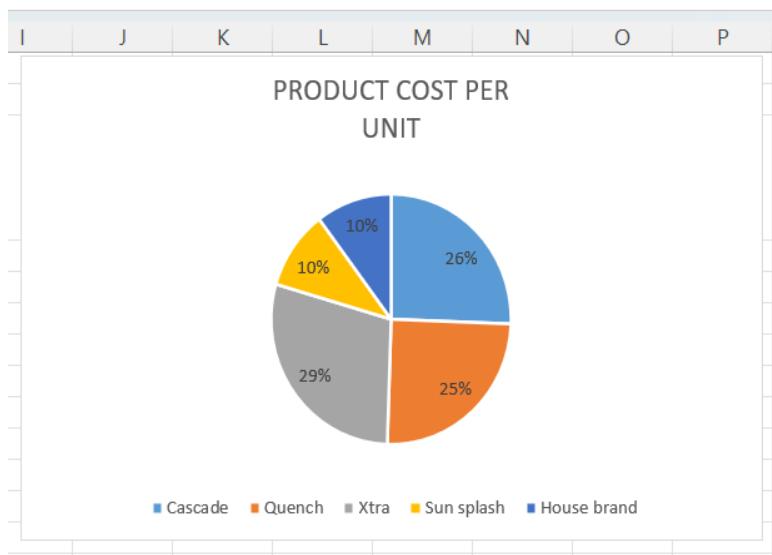
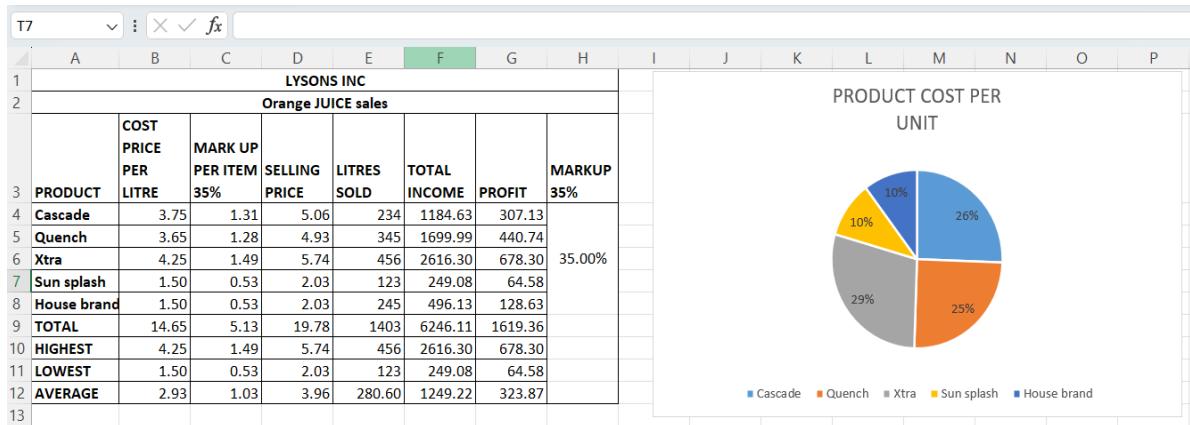
B11 : =MIN(B4:B8)

A	B	C	D	E	F	G	H
1	LYSONS INC						
2	Orange JUICE sales						
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13
5	Quench	3.65	1.28	4.93	345	1699.99	440.74
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58
8	House brand	1.50	0.53	2.03	245	496.13	128.63
9	TOTAL	14.65	5.13	19.78	1403	6246.11	1619.36
10	HIGHEST	4.25	1.49	5.74	456	2616.30	678.30
11	LOWEST	1.50	0.53	2.03	123	249.08	64.58
12	AVERAGE						

- viii. Show all formulas you have used in a new sheet. Adjust the column width so that the formulae are displayed in full and the sheets fits into one side of A4 landscape format and save it as formulas.

A	B
FORMULA	
LITRES SOLD	-
COST PRICE PER LITRE	
MARK UP PER ITEM 35%	COST PRICE PER LITRE * MARKUP 35%
SELLING PRICE	COST PRICE PER LITRE + MARK UP PER ITEM 35%
TOTAL INCOME	LITRES SOLD * SELLING PRICE
PROFIT	TOTAL INCOME - (COST PRICE PER LITRE * LITRES SOLD)
MARKUP 35%	35%
HIGHEST	MAX(FIRST CELL : LAST CELL)
LOWEST	MIN(FIRST CELL : LAST CELL)
AVERAGE	AVERAGE(FIRST CELL : LAST CELL)

- ix. Under the worksheet Create a pie chart titled PRODUCT COST PER UNIT for Product & Cost price per Litre columns. Data labels indicating percentages should be displayed.



- x. Put borders neatly on the on the work sheet & save it as LYONS2.

The screenshot shows an Excel spreadsheet with data for 'LYONS INC Orange JUICE sales'. The data includes columns for Product, Cost Price per Litre, Markup per Item 35%, Selling Price, Litres Sold, Total Income, Profit, and Markup 35%. A formula bar is visible at the top of the screen. The 'MARKUP 35%' column has a formula applied to it, indicated by a small icon in the top right corner of the cell.

	A	B	C	D	E	F	G	H
1	LYONS INC							
2	Orange JUICE sales							
3	PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT	MARKUP 35%
4	Cascade	3.75	1.31	5.06	234	1184.63	307.13	35.00%
5	Quench	3.65	1.28	4.93	345	1699.99	440.74	
6	Xtra	4.25	1.49	5.74	456	2616.30	678.30	
7	Sun splash	1.50	0.53	2.03	123	249.08	64.58	
8	House brand	1.50	0.53	2.03	245	496.13	128.63	
9	TOTAL	14.65	5.13	19.78	1403	6246.11	1619.36	
10	HIGHEST	4.25	1.49	5.74	456	2616.30	678.30	
11	LOWEST	1.50	0.53	2.03	123	249.08	64.58	
12	AVERAGE	2.93	1.03	3.96	280.60	1249.22	323.87	

WEEK – 2

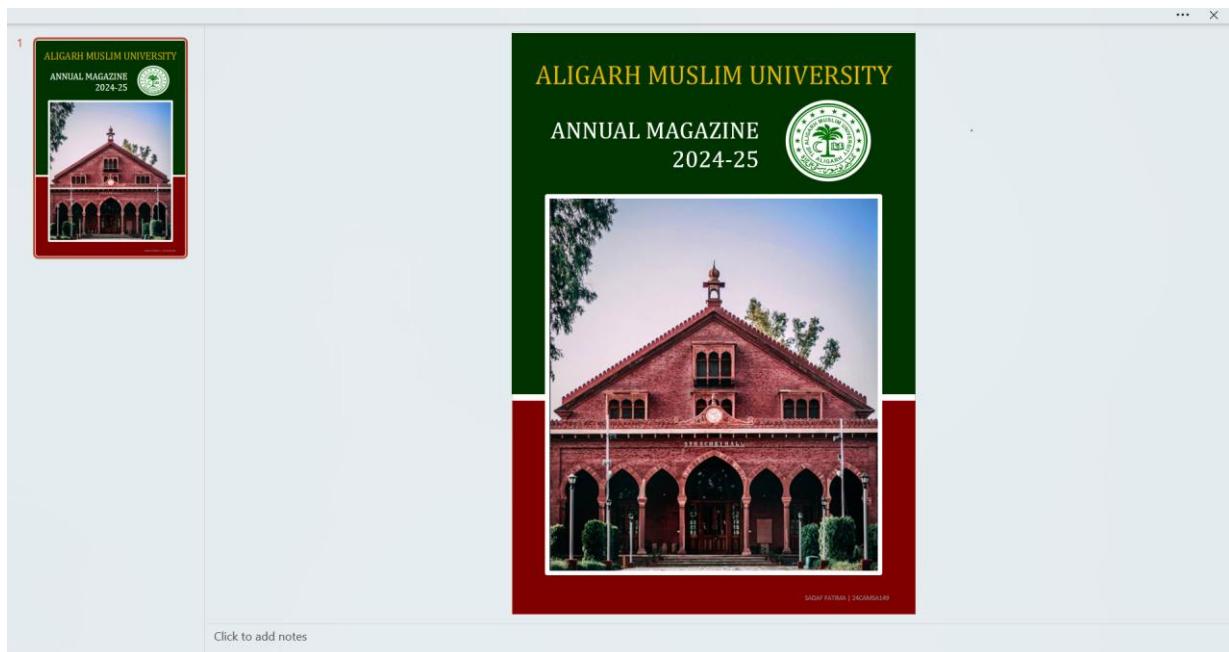
PROBLEM

1# Design Seasonal Greeting cards using MS-Power Point.



2# Design an AMU Magazine cover in MS-Power Point. Use the following:

- i. Select a theme for the page,
- ii. Insert either a picture or clipart, and
- iii. Use WordArt.



3# Design a poster inviting all students of your department to the IT Fest (using MS-Power Point).



4# Create a 5-slide presentation on any topic. Use Images, Graphs, Chart, Tables, Animation, Time, Bullets, Transition, Sound, Hyperlink, Background template, Header and Footer (using MS-Power Point).

WHAT IS SUICIDE?

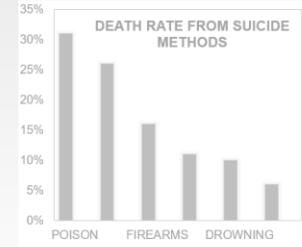
- Suicide is not about wanting to die but rather about not wanting to live.
- Suicide is about escaping unbearable pain.
- Suicide is a means to try and end situation in which a person feels trapped.
- Suicide is a major national public health issue in the India. 1.71 lakh suicides were recorded in 2022, registering a 4.2% increase over 2021 and a jump of 27% compared to 2018.



SADAFAFATIMA | 24CAMSA149

SUICIDE METHODS

COMMON METHODS OF SUICIDE AND ITS PERCENTAGE	
POISON	31%
HANGING	26%
FIREARMS	16%
BURNS	11%
DROWNING	10%
FALLING FROM HEIGHT	6%



Suicide attempts do not always result in death, and a non fatal suicide attempt can leave the person with serious physical injuries, long term health problem and brain damage.

SADAFAFATIMA | 24CAMSA149

CAUSES OF SUICIDE

While the cause of suicide is unknown, some common risk factors include :

- Major psychiatric illness like depression, bipolar, schizophrenia.
- Substance abuse (primarily alcohol abuse).
- Family history of suicide.
- Long term difficulties with relationships with family and friends.
- Losing hope or the will to live.
- Significant losses in personal life like loss of loved one, self-esteem, important relationship.
- Unbearable emotional or physical pain



SADAFAFATIMA | 24CAMSA149

SUICIDE PREVENTION

Some warning signs may help to determine if a person is at risk for suicide :

- Often talking or writing about death
- Making comments about being hopeless, worthless, helpless
- Expressions of having no reason for living
- Withdrawal from family, friends.
- Dramatic mood changes
- Talking about feeling trapped, or being burden to others

WORLD SUICIDE PREVENTION DAY IS CELEBRATED ON 10TH SEPT.

SADAF FATIMA | 24CAMSA149



HOW CAN WE HELP IN PREVENTING SUICIDE

- Ask someone whom you are worried about, if they are thinking about suicide
- Keep them safe. Reduce access to lethal means for those at risk
- Be there with them. Listen to what they need
- Help them connect with on-going support
- Stay connected. Follow up to see how they are doing

Be the one to save a life

SADAF FATIMA | 24CAMSA149



Thank You

SADAF FATIMA | 24CAMSA149

WEEK – 4

PROBLEM

1# Write a C++ program to check whether a number is even or odd using ternary operator.

Program:

```
#include <iostream>
using namespace std;

int main() {
    int x;
    cout << "Enter a number: ";
    cin >> x;

    string result = (x%2==0) ? "Even number" : "Odd Number";
    cout << result;

    return 0;
}
```

Console Output:

```
Enter a number:5
Odd Number
Process finished with exit code 0
```

```
Enter a number:6
Even number
Process finished with exit code 0
```

2# Write a C++ program to perform the addition of two numbers without using + operator.

Program:

```
#include <iostream>
using namespace std;

int add(int a, int b) {
    while (b!=0) {
        int carry = (a & b) << 1;
        a = a ^ b;
        b = carry;
    }
    return a;
}

int main(){
    int num1, num2;
    cout << "Enter first number: ";
    cin >> num1;
```

```

cout << "\nEnter second number: ";
cin >> num2;

cout << "Sum: " << add(num1, num2) << endl;
return 0;
}

```

Console Output:

```

Enter first number:6
Enter second number:9
Sum: 15

Process finished with exit code 0

```

3# Write a C++ program to evaluate the arithmetic expression $((a + b / c * d - e) * (f - g))$. Read the values a, b, c, d, e, f, g from the standard input device.

Program:

```

#include <iostream>
using namespace std;

int main() {

    float a, b, c, d, e, f, g;

    cout << "Enter the value of a: ";
    cin >> a;

    cout << "Enter the value of b: ";
    cin >> b;

    cout << "Enter the value of c: ";
    cin >> c;

    cout << "Enter the value of d: ";
    cin >> d;

    cout << "Enter the value of e: ";
    cin >> e;

    cout << "Enter the value of f: ";
    cin >> f;

    cout << "Enter the value of g: ";
    cin >> g;

    float value = ((a + b / c * d - e) * (f - g));

    cout << "\nResultant value: " << value;

    return 0;
}

```

Console Output:

```

Enter the value of a:
8
Enter the value of b:
6
Enter the value of c:
3
Enter the value of d:
5
Enter the value of e:
2
Enter the value of f:
6
Enter the value of g:
1

Resultant value: 80
Process finished with exit code 0

```

4# A Fibonacci sequence is defined as follows: The first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.

Program:

```

#include <iostream>
using namespace std;

int fibonacci(int n){

    if (n == 0){
        return 0;
    }
    else if(n == 1){
        return 1;
    }
    else{
        return fibonacci(n-1) + fibonacci(n-2);
    }
}

int main(){

    int x;
    cout << "Enter the number to which the series is to be printed: " ;
    cin >> x;

    for( int i = 0; i <= x; i++){
        cout << fibonacci(i) << " ";
    }

    return 0;
}

```

Console Output:

```
Enter the number to which the series is to be printed:9
 0  1  1  2  3  5  8  13  21  34
Process finished with exit code 0
```

5# Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Program:

```
#include <iostream>
using namespace std;

int main(){
    int n;
    cout << "Enter the number to display prime number: ";
    cin >> n;

    int noPrime = 0;
    for(int i = 2; i <= n ; i++){
        bool isPrime = true;
        for( int j = 2 ; j <= i/2; j++ ){
            if (i % j == 0){
                isPrime = false;
                break;
            }
        }

        if (isPrime){
            cout << i << " ";
            noPrime = 1;
        }
    }

    if (!noPrime){
        cout << "No prime number exists.";
    }

    return 0;
}
```

Console Output:

```
Enter the number to display prime number:44
 2 3 5 7 11 13 17 19 23 29 31 37 41 43
Process finished with exit code 0
```

6# A character is entered through keyboard. Write a C++ program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.

Characters ASCII values A – Z: 65 – 90, a – z: 97 – 122, 0 – 9: 48 – 57

Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127

Program:

```
#include <iostream>
using namespace std;

void character_using_ifElse(unsigned char a) {

    int ascii = a;
    if (ascii >= 65 and ascii <= 90){
        cout << "Character entered is a capital letter.";
    }

    else if (ascii >= 97 and ascii <= 122){
        cout << "Character entered is a small case letter.";
    }

    else if (ascii >= 48 and ascii <= 57){
        cout << "Character entered is a number.";
    }

    else{
        cout << "Character entered is a special symbol.";
    }
}

void character_using_switchCase(char ch){

    switch(ch){
        case 'A'...'Z':
            cout << "Character entered is a capital letter.";
            break;

        case 'a'...'z':
            cout << "Character entered is a small letter";
            break;

        case '0'...'9':
            cout << "Character entered is a number";
            break;

        default:
            cout << "Character entered is a special symbol.";
    }
}

int main(){
    char x;
    int sw;
    char choice;

    cout << "Which method to use SWITCH CASE (1) or IF-ELSE (2): ";
    cin >> sw;

    switch (sw) {
        case 1:

```

```

        do {
            cout << "Enter a character: ";
            cin >> x;
            character_using_switchCase(x);

            cout << "\n\nDo you want to check another character? (y/n):";
            cin >> choice;
        }

        while(choice == 'y' || choice == 'Y');
        break;

    case 2:

        do{
            cout << "\nEnter a character: ";
            cin >> x;
            character_using_ifElse(x);

            cout << "\n\nDo you want to check another character? (y/n):";
            cin >> choice;
        }
        while(choice == 'y' || choice == 'Y');
        break;

    default:
        cout << "Wrong input\n";
    }

    return 0;
}

```

Console Output:

Which method to use SWITCH CASE (1) or IF-ELSE (2):2

Enter a character:s
Character entered is a small case letter.

Do you want to check another character? (y/n):y

Enter a character:G
Character entered is a capital letter.

Do you want to check another character? (y/n):y

Enter a character:9
Character entered is a number.

Do you want to check another character? (y/n):y

Enter a character:*

Character entered is a special symbol.

Do you want to check another character? (y/n):n

Process finished with exit code 0

7# Write a C++ program to find the roots of a quadratic equation.

Program:

```
#include <iostream>
#include <cmath>
#include <ccomplex>

using namespace std;

void findRoots(double a, double b, double c){

    double discriminant = (b * b) - (4 * a * c);

    if (discriminant > 0 ){

        double root1 = (-b + sqrt(discriminant)) / (2 * a);
        double root2 = (-b - sqrt(discriminant)) / (2 * a);

        cout << "\nRoots are real and distinct.";
        cout <<"\nRoots = " << root1 << " , " << root2;
    }

    else if (discriminant == 0){
        double root = (-b) / (2 * a);
        cout << "\nRoots are real and repeated.";
        cout <<"\nRoots = " << root << " , " << root;
    }

    else{
        cout << "\nRoots are imaginary.";
    }
}

int main(){

    double x, y, z;
    char ss_Two = 253;
    char choice;

    do{
        cout << "Enter the coefficient of x" << ss_Two << " : ";
        cin >> x;

        if (x == 0){
            cout << "Coefficient of x" << ss_Two << " can't be 0.";
        }
        else{
            cout << "Enter the coefficient of x : ";
            cin >> y;
            cout << "Enter the constant term : ";
            cin >> z;
            cout << "\nQuadratic Equation: " << x << "x" << ss_Two << " + " << y
<< "x + " << z << " = 0";
            cout << "\n";

            findRoots(x, y, z);
        }

        cout << "\n\nDo you want to continue? (y/n): ";
        cin >> choice;
    }
    while(choice == 'Y' || choice == 'y');
}
```

```

        return 0;
}

```

Console Output:

Quadratic Equation: $1x^2 + 2x + 1 = 0$

Roots are real and repeated.

Roots = -1 , -1

Do you want to continue? (y/n):**y**

Enter the coefficient of x^2 :**1**

Enter the coefficient of x :**3**

Enter the constant term :**2**

Quadratic Equation: $1x^2 + 3x + 2 = 0$

Roots are real and distinct.

Roots = -1 , -2

Do you want to continue? (y/n):**y**

Enter the coefficient of x^2 :**1**

Enter the coefficient of x :**0**

Enter the constant term :**4**

Quadratic Equation: $1x^2 + 0x + 4 = 0$

Roots are imaginary.

8# Write a C++ program to check whether a given 3-digit number is Armstrong number or not.

Program:

```

#include <iostream>
#include <cmath>

using namespace std;

int main(){

    char choice;

    do{
        int x,n,count = 0;
        double sum =0;
        cout << "\nEnter a number to check whether it is a armstrong number or
not: ";
        cin >> x;
        n = x;

        while(n!=0){
            count++;
            n=n/10;
        }

```

```

n = x;

while(n != 0){
    int r = n % 10;
    sum = sum + pow(r, count);
    n = n/10;
}

if (x == sum){
    cout << "It is a Armstrong number.";
}
else{
    cout << "It is not a Armstrong number.";
}

cout << "\n\nDo you want to continue? (y/n): ";
cin >> choice;

}

while(choice == 'y' || choice == 'Y');

return 0;
}

```

Console Output:

Enter a number to check whether it is a armstrong number or not:**132**
It is not a Armstrong number.

Do you want to continue? (y/n):**y**

Enter a number to check whether it is a armstrong number or not:**153**
It is a Armstrong number.

Do you want to continue? (y/n):**y**

Enter a number to check whether it is a armstrong number or not:**8208**
It is a Armstrong number.

Do you want to continue? (y/n):**n**

Process finished with exit code 0

WEEK – 5

PROBLEM

1# Write a C++ program to enter elements in the array and display the array elements.

Program:

```
#include <iostream>
using namespace std;

int main() {
    int n;

    cout << "Enter number to elements to be inserted in an Array: ";
    cin >> n;

    int array[n];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for(int i = 0; i<n; i++){
        cin >> array[i];
    }

    cout << "\nDisplaying Inserted Array elements: ";
    for(int i = 0; i<n-1;i++){

        cout << array[i] << " , ";
    }

    cout << array[n-1] << "}";

    return 0;
}
```

Console Output:

```
Enter number to elements to be inserted in an Array:6
Enter elements to be inserted in an array:5
3
2
7
1
9

Displaying Inserted Array elements: {5 , 3 , 2 , 7 , 1 , 9}
Process finished with exit code 0
```

2# Write a C++ program to find the sum of the all-array element.

Program:

```
#include <iostream>
using namespace std;

int main(){
    int n;
    cout << "Enter number of elements to be inserted in an array: ";
    cin >> n;
    int array[n];

    int sum = 0;
    cout << "Insert array elements: ";
    for(int i = 0; i<n ; i++){
        cin >> array[i];
        sum += array[i];
    }
    cout << "\nSum of array elements: " << sum;

    return 0;
}
```

Console Output:

```
Enter number of elements to be inserted in an array:6
Insert array elements:3
2
9
1
7
4

Sum of array elements: 26
Process finished with exit code 0
```

3# Write a C++ program to find the length of the array.

Program:

```
#include <iostream>
using namespace std;

int main(){

    int array[] = {2, 3, 4, 6, 7};

    int length = sizeof(array) / sizeof (array[0]);

    cout << "Length of array: " << length;

    return 0;
}
```

Console Output:

```
Length of array: 5
Process finished with exit code 0
```

4# Write a C++ program to find the second-largest integer in a list of integers.

Program:

```
#include <iostream>
using namespace std;

int main(){
    int n;

    cout << "Enter number of elements to be inserted in an Array: ";
    cin >> n;

    int array[n];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for(int i = 0; i<n; i++){
        cin >> array[i];
    }

    cout << "\nDisplaying Inserted Array elements: ";
    for(int i = 0; i<n;i++){

        cout << array[i] << "   ";
    }

    //Sorting elements using bubble sort

    for (int i =0; i<n-1;i++){
        for(int j = 0 ; j<n-1; j++){
            if (array [j] < array [j+1]){
                int temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
            }
        }
    }

    cout << "\nElements after Sorting: ";
    for(int i = 0; i<n; i++){
        cout << array[i] << "   ";
    }

    int max = array[0];
    bool isMax = true;

    for(int i = 1; i < n;i++){
        if (max !=array[i]){
            cout << "\nSecond largest element: " << array[i];
            isMax = false;
            break;
        }
    }
}
```

```

if(isMax) {
    cout << "\nSecond largest element does not exists in this array.";
}

}

```

Console Output:

```

Enter number to elements to be inserted in an Array:5
Enter elements to be inserted in an array:2
3
5
5
4

Displaying Inserted Array elements: 2  3  5  5  4
Elements after Sorting: 5  5  4  3  2
Second largest element: 4
Process finished with exit code 0

```

5# Write a C++ Program to reverse the position of the array element (Hint: First eminent to the last element.)

Program:

```

#include <iostream>
using namespace std;

int main(){

    int n;
    cout << "Enter number to elements to be inserted in an Array: ";
    cin >> n;

    int array[n];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for(int i = 0; i<n; i++){
        cin >> array[i];
    }

    cout << "\nDisplaying Inserted Array elements: ";
    for(int i = 0; i<n;i++){
        cout << array[i] << "   ";
    }

    for(int i = 0; i < n/2;i++){

        int temp = array[i];
        array[i] = array[n-i-1];
        array [n-i-1] = temp;
    }
}

```

```

cout << "\nArray elements after reversing: ";

for(int i = 0; i < n; i++) {
    cout << array[i] << " ";
}

return 0;
}

```

Console Output:

```

Enter number to elements to be inserted in an Array:6
Enter elements to be inserted in an array:1
2
3
4
5
6

Displaying Inserted Array elements: 1  2  3  4  5  6
Array elements after reversing: 6  5  4  3  2  1
Process finished with exit code 0

```

6# Write a C++ program to perform the following:

- Addition of two matrices
- Multiplication of two matrices

Program:

```

#include <iostream>
#include <vector>

using namespace std;

void matrix_mul(int m, int n, int p, const vector<vector<int>>& matrix1, const
vector<vector<int>>& matrix2, vector<vector<int>>& matrix3) { // m = r1 , n = c1
, n = r2, p = c2 , m = r3/r1 p = c3/c2;

    for(int i = 0; i < m ; i++){
        for(int j = 0; j < p ; j++){
            matrix3[i][j] = 0;
            for(int k = 0; k < n; k++){
                matrix3[i][j] += matrix1[i][k]*matrix2[k][j];
            }
        }
    }
}

void matrix_add(int m, int n, const vector<vector<int>>& matrix1, const
vector<vector<int>>& matrix2, vector<vector<int>>& matrix3){

    for(int i = 0; i < m; i++){

```

```

        for(int j = 0; j<n; j++){
            matrix3[i][j] = matrix1[i][j]+matrix2[i][j];
        }
    }

void matrix_insert_elements(int m, int n, vector<vector<int>>& matrix){
    for (int i = 0; i < m;i++){
        for(int j = 0; j < n ;j++) {
            cin >> matrix[i][j];
        }
        cout << "\n";
    }
}

void display_matrix_elements(int m, int n, vector<vector<int>>& matrix){

    for (int i = 0; i < m ; i++){
        for (int j = 0; j < n ;j++) {
            cout << matrix[i][j] << "   ";
        }
        cout << "\n";
    }
}

int main(){

    int m, n, p;

    int sw;
    cout << "Select '1' for Matrix Multiplication and '2' for Matrix Addition:
";
    cin >> sw;

    switch(sw) {

        case 1: {
            cout << "\nMATRIX MULTIPLICATION";

            cout << "\nEnter Rows for first Matrix: ";
            cin >> m;

            cout << "Enter Columns for first Matrix: ";
            cin >> n;

            cout << "Enter Rows of Second Matrix: " << n;
            cout << "\nEnter Columns for second matrix: ";
            cin >> p;

            vector<vector<int>> matrix1_mul(m, vector<int>(n));
            vector<vector<int>> matrix2_mul(n, vector<int>(p));
            vector<vector<int>> result_mul(m, vector<int>(p));

            cout << "\nEnter elements for the first matrix:\n";
            matrix_insert_elements(m, n, matrix1_mul);

            cout << "\nEnter elements for the second matrix:\n";
            matrix_insert_elements(n, p, matrix2_mul);

            cout << "\nFirst matrix: \n";
            display_matrix_elements(m, n, matrix1_mul);
        }
    }
}

```

```

        cout << "\nSecond matrix: \n";
        display_matrix_elements(n, p, matrix2_mul);

        matrix_mul(m, n, p, matrix1_mul, matrix2_mul, result_mul);

        cout << "\nResulting Matrix after Multiplication:\n";
        display_matrix_elements(m, p, result_mul);

        break;
    }

    case 2: {

        cout << "\nMATRIX ADDITION";
        cout << "\nEnter rows for matrices: ";
        cin >> m;

        cout << "Enter columns for matrices: ";
        cin >> n;

        vector<vector<int>> matrix1_add(m, vector<int>(n));
        vector<vector<int>> matrix2_add(m, vector<int>(n));
        vector<vector<int>> result_add(m, vector<int>(n));

        cout << "\nEnter first matrix elements: \n";
        matrix_insert_elements(m, n, matrix1_add);

        cout << "\nEnter second matrix elements: \n";
        matrix_insert_elements(m, n, matrix2_add);

        cout << "\nFirst matrix: \n";
        display_matrix_elements(m, n, matrix1_add);

        cout << "\nSecond matrix: \n";
        display_matrix_elements(m, n, matrix2_add);

        matrix_add(m, n, matrix1_add, matrix2_add, result_add);

        cout << "\nResulting Matrix after Addition:\n";
        display_matrix_elements(m, n, result_add);

        break;
    }

    default:
        cout << "Select either 1 or 2.";
    }

    return 0;
}

```

Console Output:

```
Select '1' for Matrix Multiplication and '2' for Matrix Addition:1
```

```
MATRIX MULTIPLICATION
```

```
Enter Rows for first Matrix:2
```

```
Enter Columns for first Matrix:2
```

```
Enter Rows of Second Matrix: 2
```

```
Enter Columns for second matrix:2
```

```
Enter elements for the first matrix:
```

```
3
```

```
4
```

```
6
```

```
5
```

```
Enter elements for the second matrix:
```

```
1
```

```
2
```

```
6
```

```
5
```

```
First matrix:
```

```
3 4
```

```
6 5
```

```
Second matrix:
```

```
1 2
```

```
6 5
```

```
Resulting Matrix after Multiplication:
```

```
27 26
```

```
36 37
```

```
Process finished with exit code 0
```

```
Select '1' for Matrix Multiplication and '2' for Matrix Addition:2
```

MATRIX ADDITION

```
Enter rows for matrices:2
```

```
Enter columns for matrices:2
```

```
Enter first matrix elements:
```

```
3
```

```
2
```

```
5
```

```
4
```

```
Enter second matrix elements:
```

```
1
```

```
2
```

```
5
```

```
3
```

```
First matrix:
```

```
3 2
```

```
5 4
```

```
Second matrix:
```

```
1 2
```

```
5 3
```

```
Resulting Matrix after Addition:
```

```
4 4
```

```
10 7
```

```
Process finished with exit code 0
```

7# Write a C++ program to count and display positive, negative, odd and even numbers in an array.

Program:

```
#include <iostream>
using namespace std;

void positive_negative_integers(int n, int array[]){
    int positive_count = 0;
    int negative_count = 0;

    cout << "\n\nPositive integers in an Array: ";
    for (int i = 0 ; i < n; i++) {
```

```

    if (array[i] > 0) {
        cout << array[i] << " ";
        positive_count++;
    }
}

cout << "\nCount of Positive integers in an Array: " << positive_count;

cout << "\n\nNegative integers in an Array: ";
for (int i = 0 ; i < n; i++) {
    if (array[i] < 0) {
        cout << array[i] << " ";
        negative_count++;
    }
}

cout << "\nCount of Negative integers in an Array: " << negative_count;
}

void odd_even_integers(int n, int array[]){
    int odd_count = 0;
    int even_count = 0;

    cout << "\n\nOdd integers in an Array: ";
    for (int i = 0 ; i < n; i++) {
        if (array[i] % 2 != 0) {
            cout << array[i] << " ";
            odd_count++;
        }
    }

    cout << "\nCount of Odd integers in an Array: " << odd_count;

    cout << "\n\nEven integers in an Array: ";
    for (int i = 0 ; i < n; i++) {
        if (array[i] % 2 == 0) {
            cout << array[i] << " ";
            even_count++;
        }
    }

    cout << "\nCount of Even integers in an Array: " << even_count;
}

int main(){
    int n;

    cout << "Enter number of elements to be inserted in an Array: ";
    cin >> n;

    int array[n];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for (int i = 0; i < n; i++) {
        cin >> array[i];
    }

    cout << "\nDisplaying Inserted Array elements: ";
    for (int i = 0; i < n; i++) {

```

```

        cout << array[i] << "   ";
    }

positive_negative_integers(n, array);
odd_even_integers(n, array);

return 0;
}

```

Console Output:

Enter number to elements to be inserted in an Array:7

Enter elements to be inserted in an array:2

3

4

-2

-9

-5

8

Displaying Inserted Array elements: 2 3 4 -2 -9 -5 8

Positive integers in an Array: 2 3 4 8

Count of Positive integers in an Array: 4

Negative integers in an Array: -2 -9 -5

Count of Negative integers in an Array: 3

Odd integers in an Array: 3 -9 -5

Count of Odd integers in an Array: 3

Even integers in an Array: 2 4 -2 8

Count of Even integers in an Array: 4

Process finished with exit code 0

8# Write a C++ program to merge two sorted arrays into another array in sorted order.

Program:

```

#include <iostream>
using namespace std;

void sort(int n, int array[]){

    for (int i = 0; i < n-1; i++){
        for (int j= 0; j < n-1 ;j++){
            if (array[j] > array [j+1]){
                int temp = array[j];
                array [j] = array [j+1];
                array[j+1] = temp;
            }
        }
    }
}

```

```

}

int main(){

    int n, m;

    //First array

    cout << "Enter number of elements to be inserted in first Array: ";
    cin >> n;

    int array_n[n];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for(int i = 0; i<n; i++){
        cin >> array_n[i];
    }

    //Second Array

    cout << "\nEnter number of elements to be inserted in second Array: ";
    cin >> m;

    int array_m[m];

    //Scanning and Inserting array elements

    cout << "Enter elements to be inserted in an array: ";
    for(int i = 0; i<m; i++){
        cin >> array_m[i];
    }

    //Displaying both arrays.

    sort(n, array_n);
    sort(m, array_m);

    cout << "\nDisplaying Inserted First Array elements: ";
    for(int i = 0; i<n; i++){
        cout << array_n[i] << "    ";
    }

    cout << "\nDisplaying Inserted Second Array elements: ";
    for(int i = 0; i<m; i++){
        cout << array_m[i] << "    ";
    }

    //merged array
    int array_mn[m+n];

    int i = 0, j = 0, k = 0;

    while (i < n && j < m) {
        if (array_n[i] < array_m[j]) {
            array_mn[k++] = array_n[i++];
        } else {
            array_mn[k++] = array_m[j++];
        }
    }
}

```

```

// Adding remaining elements of array_n
while (i < n) {
    array_mn[k++] = array_n[i++];
}

// Adding remaining elements of array_m
while (j < m) {
    array_mn[k++] = array_m[j++];
}

cout << "\nMerged Sorted Array: ";
for (int z = 0; z < (m + n); z++) {
    cout << array_mn[z] << " ";
}

return 0;
}

```

Console Output:

Enter number to elements to be inserted in first Array:4

Enter elements to be inserted in an array:1

7
3
3

Enter number to elements to be inserted in second Array:7

Enter elements to be inserted in an array:2

3
1
8
7
6
5

Displaying Inserted First Array elements: 1 3 3 7

Displaying Inserted Second Array elements: 1 2 3 5 6 7 8

Merged Sorted Array: 1 1 2 3 3 3 5 6 7 7 8

Process finished with exit code 0

9# Write a C++ program to find the frequency of a particular number in a list of integers.

Program:

```

#include <iostream>
using namespace std;

int main(){

    int n, element;
    cout << "Enter number to elements to be inserted in an Array: ";
    cin >> n;

```

```

int array[n];

//Scanning and Inserting array elements

cout << "Enter elements to be inserted in an array: ";
for(int i = 0; i<n; i++){
    cin >> array[i];
}

cout << "\nDisplaying Inserted Array elements: ";
for(int i = 0; i<n;i++) {

    cout << array[i] << " ";
}

cout << "\nEnter the element to found its frequency: ";
cin >> element;

int frequency = 0;

for (int i = 0;i<n;i++) {
    if(element == array[i]){
        frequency++;
    }
}

if (frequency==0){
    cout << "\nElement doesn't exists.";
}
else{
    cout << "\nFrequency of " << element <<" is: "<<frequency;
}

return 0;
}

```

Console Output:

Enter number to elements to be inserted in an Array:7

Enter elements to be inserted in an array:2

1

4

4

5

3

6

Displaying Inserted Array elements: 2 1 4 4 5 3 6

Enter the element to found its frequency:4

Frequency of 4 is: 2

Process finished with exit code 0

WEEK – 6

PROBLEM

1# Write a C++ Program for Add Two Numbers Using Pointer.

Program:

```
#include <iostream>

int add(int* a, int* b) {
    return *a + *b;
}

int main() {
    int a, b;

    std::cout << "Enter two numbers: ";
    std::cin >>a >>b;

    std::cout << "Sum: " << add(&a, &b);
    return 0;
}
```

Console Output:

```
Enter two numbers:6 7

Sum: 13
Process finished with exit code 0
```

2# Write a C++ Program for Swap Numbers Using Pointers.

Program:

```
#include <iostream>

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {

    int a, b;
    std::cout << "Enter two variables: ";
    std::cin >>a>>b;

    std::cout << "\nNumber before Swapping: " << a << "    " << b;
    swap(&a, &b);
    std::cout << "\nNumber after Swapping: " << a << "    " << b;
```

```

        return 0;
}

```

Console Output:

```
Enter two variables:4 7
```

```

Number before Swapping: 4 7
Number after Swapping: 7 4
Process finished with exit code 0

```

3# Write a C++ Program to Print the address of the Variable Using a Pointer.**Program:**

```
#include <iostream>

int main(){

    int x, *ptr = &x;
    std::cout << "Address of a declared variable is: " << ptr;
    return 0;
}
```

Console Output:

```
Address of a declared variable is: 0x37c5dff834
```

```
Process finished with exit code 0
```

4# Write a C++ Program for Increment and Decrement Integer Using Pointer.**Program:**

```
#include <iostream>
using namespace std;

int main(){

    int x, *ptr = &x;
    int sw;
    char choice;

    cout << "Enter a variable: ";
    cin >> x;
    do {
        cout << "\nEnter '1' for INCREMENT and '2' for DECREMENT: ";
        cin >> sw;

        switch (sw) {
            case 1: {

```

```

        cout << "\nNumber after Increment: " << ++(*ptr);
        break;
    }
    case 2: {
        cout << "\nNumber after Decrement: " << --(*ptr);
        break;
    }

    default:
        cout << "Wrong Input";
}

cout << "\nDo you want to continue (y/n) : ";
cin >> choice;
}
while(choice == 'y' || choice == 'Y');

return 0;
}

```

Console Output:

```

Enter a variable:3

Select '1' for INCREMENT and '2' for DECREMENT:1

Number after Increment: 4

Do you want to continue (y/n) :y

Select '1' for INCREMENT and '2' for DECREMENT:2

Number after Decrement: 3

Do you want to continue (y/n) :n

```

5# Write a C++ Program to Print String Using Pointer.**Program:**

```

#include <iostream>

int main(){
    char c[50], *ptr = c;
    std::cout << "Enter a string: ";
    std::getline(c, 50);

    std::cout << "Your entered string: ";
    while(*ptr != '\0'){
        std::cout << *ptr;
        ptr++;
    }
    return 0;
}

```

Console Output:

```
Enter a string:Sadaf Fatima
```

```
Your entered string: Sadaf Fatima
```

```
Process finished with exit code 0
```

6# Write a C++ program to concatenate two strings using pointers.

Program:

```
#include<iostream>

void concatStr(char *str1, const char *str2){

    while (*str1 != '\0'){
        str1++;
    }

    while(*str2 != '\0'){ //swapping takes place till middle character(index)
        *str1 = *str2;
        str1++;
        str2++;
    }

    *str1 = '\0';
}

int main(){

    char str1[100];
    char str2[50];

    std::cout << "Enter first string: ";
    std::cin.getline(str1, 50);

    std::cout << "Enter second string: ";
    std::cin.getline(str2, 50);

    concatStr(str1,str2);

    std::cout << "Concatenated string: " << str1;

    return 0;
}
```

Console Output:

```
Enter first string:Sadaf Fatima
```

```
Enter second string: - 24CAMSA149 , GK6762
```

```
Concatenated string: Sadaf Fatima - 24CAMSA149 , GK6762
```

```
Process finished with exit code 0
```

7# Write a program for reading elements using a pointer into an array and display the values using an array.

- i. Declare a set of elements.
- ii. Declare the pointer and initialize it to the first element address of a set of elements(array).
- iii. Repeat the loop until the pointer reaches to the last element and displays each element.

Program:

```
#include <iostream>
using namespace std;

int main() {

    int n;
    cout << "Enter elements to be inserted in an array: ";
    cin >> n;

    int array[n];
    int *ptr = array;

    cout << "\nInsert array elements: ";
    for(int i = 0 ; i < n ;i++){
        cin >> *(ptr+i);
    }

    cout << "\nArray elements: ";
    for (int i = 0; i< n ;i++){
        cout << array[i] << " ";
    }

    return 0;
}
```

Console Output:

```
Enter elements to be inserted in an array:5

Insert array elements:6 5 4 1 2

Array elements: 6 5 4 1 2

Process finished with exit code 0
```

8# Write a program through a pointer variable to the sum of n elements from the array.

Program:

```
#include <iostream>

int main(){
    int n, *array = new int[n], sum = 0;
    std::cout << "Enter number of elements to be inserted in an array: ";
    std::cin >> n;

    std::cout << "\nInsert array elements: ";
    for ( int i = 0; i < n ;i++){
        std::cin >> *(array+i);
        sum += *(array+i);
```

```

    }

    std::cout << "\nSum of elements : " << sum;

    return 0;
}

```

Console Output:

Enter number of elements to be inserted in an array:5

Insert array elements:1 2 3 4 5

Sum of elements : 15

Process finished with exit code 0

9# Write a program for reading elements using a pointer into the array and display the values using an array.

Program:

```

#include <iostream>
using namespace std;

int main() {

    int n;
    cout << "Enter elements to be inserted in an array: ";
    cin >> n;

    int array[n];
    int *ptr = array;

    cout << "\nInsert array elements: ";
    for(int i = 0 ; i < n ;i++){
        cin >> *(ptr+i);
    }

    cout << "\nArray elements: ";
    for (int i = 0; i< n ;i++){
        cout << *(ptr+i) << " ";
    }

    return 0;
}

```

Console Output:

Enter elements to be inserted in an array:5

Insert array elements:6 5 4 1 2

Array elements: 6 5 4 1 2

Process finished with exit code 0

10# Write a C++ program to reverse a string using pointers.

Program:

```
#include<iostream>

void reverseStr(char *ptr) {
    char* begin = ptr, *end = ptr; // both of them will point to the first
    character

    while (*end != '\0') {
        end++;
    }
    end--;

    while (begin < end){ //swapping takes place till middle character(index)
        char temp = *begin;
        *begin = *end;
        *end = temp;

        begin++;
        end--;
    }
}

int main(){
    char str[50];

    std::cout << "Enter a string: ";
    std::cin.getline(str, 50);

    reverseStr(str);

    std::cout << "Reversed string: " << str;
    return 0;
}
```

Console Output:

Enter a string:*Sadaf Fatima - 24CAMSA149*

Reversed string: *941ASMAC42 - amitaF fadas*

Process finished with exit code 0

WEEK – 7

PROBLEM

1# Write a C++ Program to Count vowels String Using Pointer.

Program:

```
#include<iostream>
#include <cctype>

int countVowel(char *ch) {
    int count = 0;
    while(*ch != '\0') {
        *ch = tolower(*ch);

        if(*ch == 'a' || *ch == 'e' || *ch == 'i' || *ch == 'o' || *ch == 'u') {
            count++;
        }
        ch++;
    }

    return count;
}

int main() {
    char str[50];
    std::cout << "Enter a string: ";
    std::getline(str, 50);

    std::cout << "Vowel count: " << countVowel(str);
}
```

Console Output:

```
Enter a string:This is a test string to count vowels.
```

```
Vowel count: 10
```

```
Process finished with exit code 0
```

2# Write a C++ Program for Length of String Using Pointer.

Program:

```
#include<iostream>

int lenStr(const char *str) {
    int count = 0;
    while(*str != '\0') {
        str++;
        count++;
    }
    return count;
}
```

```

int main() {

    char str[50];
    std::cout << "Enter a string: ";
    std::getline(str, 100);

    std::cout << "Length of the string: " << lenStr(str);
}

```

Console Output:

Enter a string:*This is test string to find its length.*

Length of the string: 39

Process finished with exit code 0

3# Write a C++ program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

Program:

```

#include <iostream>
#include <cmath>

float sum_elements(int n, float *array) {
    float sum = 0;
    for(int i = 0; i < n; i++) {
        sum += *(array + i);
    }
    return sum;
}

float mean_elements(int n, float *array) {
    float mean = sum_elements(n, array) / n;
    return mean;
}

float standard_deviation(int n, float *array) {

    float sd = 0, diff_mean = 0, mean = mean_elements(n, array);

    for(int i = 0; i < n; i++) {
        diff_mean += powf((*(array + i)) - mean, 2);
    }

    sd = sqrtf(diff_mean / n);

    return sd;
}

int main() {
    int n;
    auto *array = new float[n];

    std::cout << "Enter elements to be inserted in an array: ";
    std::cin >> n;
}

```

```

    std::cout << "Insert array elements: ";
    for(int i = 0; i < n ;i++){
        std::cin >> *(array+i);
    }

    std::cout << "\nArray elements: ";
    for(int i = 0; i < n ;i++){
        std::cout<< *(array+i) << " ";
    }

    std::cout << "\nSum: " << sum_elements(n,array);
    std::cout << "\nMean: " << mean_elements(n, array);
    std::cout << "\nStandard Deviation: " << standard_deviation(n, array);

    return 0;
}

```

Console Output:

Enter elements to be inserted in an array:5

Insert array elements:1 2 3 4 5

Array elements: 1 2 3 4 5

Sum: 15

Mean: 3

Standard Deviation: 1.41421

Process finished with exit code 0

4# Write a C++ program to create three objects for a class named pntr_obj with data members such as roll_no & name. Create a member function set_data() for setting the data values and print() member function to print which object has invoked it using the ‘this’ pointer.

Program:

```

#include <iostream>

class pntr_obj {

private:
    int roll_no;
    std::string name;

public:

    void set_data(){
        std::cout<<"\nEnter Roll No: ";
        std::cin >> roll_no;
        std::cin.ignore();
        std::cout << "Enter Name: ";
        getline(std::cin, name);
    }
}

```

```

        std::cout << std::endl;
    }

    void print(){
        std::cout << "Object at address: " << this << std::endl;
        std::cout << "Roll No: " << roll_no << std::endl;
        std::cout << "Name: " << name << std::endl;
        std::cout << std::endl;
    }

};

int main(){

    //Creating objects

    pptr_obj obj1, obj2, obj3;

    //set data for each object by taking user input

    std::cout << "Enter data for first object";
    obj1.set_data();
    std::cout << "Enter data for second object";
    obj2.set_data();
    std::cout << "Enter data for third object";
    obj3.set_data();

    //Displaying data

    obj1.print();
    obj2.print();
    obj3.print();

    return 0;
}

```

Console Output:

Enter data for first object

Enter Roll No:**149**

Enter Name:**Sadaf Fatima**

Enter data for second object

Enter Roll No:**108**

Enter Name:**Adeeba Zakir**

Enter data for third object

Enter Roll No:**157**

Enter Name:**Aneela Usmani**

Object at address: 0x6ce33ffe10

Roll No: 149

Name: Sadaf Fatima

Object at address: 0x6ce33ffdde0

Roll No: 108

Name: Adeeba Zakir

Object at address: 0x6ce33ffdb0

Roll No: 157

Name: Aneela Usmani

5# Develop a C++ program to find the greatest of two numbers using this pointer which returns the member variable.

Program:

```
#include <iostream>

class number{
private:
    int a, b;

public:
    void setInput() {
        std::cout << "Enter two numbers: ";
        std::cin >> a >> b;
    }

    int greatestNumber() {
        if(this->a > this->b)
            return this->a;
        else
            return this->b;
    }
};

int main() {

    number obj;

    obj.setInput();
    std::cout << "\nGreatest of two numbers: " << obj.greatestNumber();

}
```

Console Output:

Enter two numbers: 6 9

Greatest of two numbers: 9

Process finished with exit code 0

6# Write a C++ program to implement flight class with data member as flight_no., source destination and fare. Write a member function to display the flight information using this pointer.

Program:

```
#include <iostream>

class flight{

private:
    int flight_no;
    std::string source;
    std::string destination;
    unsigned int fare;

public:

    void setData() {
        std::cout << "Enter flight Number: ";
        std::cin >> flight_no;

        std::cin.ignore();

        std::cout << "Enter source: ";
        getline(std::cin, source);

        std::cout << "Enter destination: ";
        getline(std::cin, destination);

        std::cout << "Enter fare: ";
        std::cin >> fare;
    }

    void displayData() {
        std::cout << "\nFlight Number: " << this->flight_no << std::endl;
        std::cout << "Source: " << this->source << std::endl;
        std::cout << "Destination: " << this->destination << std::endl;
        std::cout << "Fare: " << this->fare << std::endl;
    }
};

int main(){
    flight obj;

    obj.setData();
    obj.displayData();
}
```

Console Output:

```
Enter flight Number:123

Enter source:Delhi

Enter destination:Kerala

Enter fare:10000
```

```

Flight Number: 123
Source: Delhi
Destination: Kerala
Fare: 10000

```

```
Process finished with exit code 0
```

7# Write a C++ program to use this pointer and return the pointer reference.

Program:

```

#include <iostream>

class myClass{

private:
    int num;

public:
    myClass* set_value(int n) {
        this->num = n;
        return this;
    }

    void print_value() {
        std::cout << "Value: " << num << std::endl;
    }
};

int main(){
    myClass obj;
    int n;

    std::cout << "Enter a number: ";
    std::cin >> n;

    obj.set_value(n)->print_value(); //set value and return pointer reference;

    std::cout << "\nEnter another input: ";
    std::cin >> n;

    obj.set_value(n)->print_value();

}

```

Console Output:

```
Enter a number:12
```

```
Value: 12
```

```
Enter another input:54
```

```
Value: 54
```

```
Process finished with exit code 0
```

WEEK – 8

PROBLEM

1# Write a C++ program that uses functions to perform the following operations:

- i. To insert a sub string into a given main string from a given position.
- ii. ii. To delete n characters from a given position in a given string.

Program:

```
#include <iostream>

std::string subStr_insert(std::string str, std::string subStr, int pos) {

    std::string new_str = "";
    for(int i = 0; i < (str.length() + subStr.length()); i++) {
        if(i < pos) {
            new_str += str[i];
        }
        else if( i >= pos && i < pos + subStr.length()) {
            new_str += subStr[i - pos];
        }
        else{
            new_str += str[i - subStr.length()];
        }
    }

    return new_str;
}

std::string delete_str_char(std::string str, int nChar, int pos) {
    std::string new_str= "";

    for(int i = 0; i < (str.length() - nChar) ; i++) {
        if(i < pos) {
            new_str += str[i];
        }
        else{
            new_str += str[i + nChar];
        }
    }

    return new_str;
}

int main(){

    std::string str;
    int pos;

    int sw;
    std::cout << "Select '1' to INSERT SUBSTRING and '2' to DELETE n CHARACTERS
FROM STRING: ";
    std::cin >> sw;
    std::cin.ignore();
}
```

```

switch(sw) {
    case 1:{
        std::string subStr;

        std::cout << "\nEnter a string: ";
        getline(std::cin, str);

        std::cout << "Enter a substring to be inserted in an String: ";
        getline(std::cin, subStr);

        std::cout << "Enter position where the sub string is to be inserted:
";
        std::cin >> pos;

        std::cout << "\nResultant String: " << subStr_insert(str, subStr,
pos);

        break;
    }

    case 2:{
        int nChar;

        std::cout << "\nEnter a string: ";
        getline(std::cin, str);

        std::cout << "Enter number of characters deleted from a string: ";
        std::cin >> nChar;

        std::cout << "Enter position where the characters is to be deleted:
";
        std::cin >> pos;

        std::cout << "\nResultant String: " << delete_str_char(str, nChar,
pos);

        break;
    }
    default: {
        std::cout << "Wrong input...";
    }
}

return 0;
}

```

Console Output:

Select '1' to INSERT SUBSTRING and '2' to DELETE n CHARACTERS FROM STRING:**1**

Enter a string:**Sadaf**

Enter a substring to be inserted in an String:**123**

Enter position where the sub string is to be inserted:**2**

Resultant String: **Sa123daf**

Select '1' to INSERT SUBSTRING and '2' to DELETE n CHARACTERS FROM STRING:2

Enter a string:**Sadddaaf**

Enter number of characters deleted from a string:**3**

Enter position where the characters is to be deleted:**2**

Resultant String: Sadaf

Process finished with exit code 0

2# Write a C++ program to determine if the given string is a palindrome or not.

Program:

```
#include <iostream>

std::string newString(std::string str){
    std::string new_str = "";
    for(int i = 0; i < str.length(); i++) {
        char c = tolower(str[i]);
        int ascii = c;
        if(ascii >= 97 && ascii <= 122) {
            new_str+=c;
        }
    }
    return new_str;
}

std::string revStr(std::string str){
    std::string new_str = newString(str);
    std::string rev = "";
    for(int i = new_str.length()-1; i >= 0; i--) {
        rev+= new_str[i];
    }
    return rev;
}

bool isPalindrome(std::string str){
    std::string s = newString(str) , rev = revStr(str);
    return (s==rev);
}

int main(){
    std::string str;
    std::cout << "Enter a string: ";
    getline(std::cin, str);

    (isPalindrome(str)) ? std::cout << "\nString is Palindrome" : std::cout <<
    "\nString is not Palindrome";

    return 0;
}
```

Console Output:

```
Enter a string:Was it a car or a cat I saw?
```

```
String is Palindrome
```

```
Enter a string:Was it car or a cat I saw?
```

```
String is not Palindrome
```

```
Process finished with exit code 0
```

3# Write a C++ program to find a string within a sentence and replace it with another string.

Program:

```
#include <iostream>
#include <string>
#include <algorithm>

std::string replaceStr(std::string sentence, const std::string& toFind, const std::string& toReplace){
    size_t pos = sentence.find(toFind); //size_t is unsigned long long int
    while(pos!=std::string::npos){
        sentence.replace(pos, toFind.length(), toReplace);
        pos = sentence.find(toFind, pos + toReplace.length());
    }

    return sentence;
}

std::string toLower(const std::string& str){
    std::string result = str;
    transform(result.begin(), result.end(), result.begin(), ::tolower);
    return result;
}

int main(){
    std::string str, str_toFind, str_toReplace;

    std::cout << "Enter a sentence: ";
    getline(std::cin, str);

    std::cout << "Enter the string to find: ";
    getline(std::cin, str_toFind);

    std::cout << "Enter the string to replace: ";
    getline(std::cin, str_toReplace);

    std::cout << "\nModified Sentence: " << replaceStr(str, str_toFind, str_toReplace);
}
```

Console Output:

```
Enter a sentence:Sadaf Fatima - 24CAMSA123
Enter the string to find:123
Enter the string to replace:149
Modified Sentence: Sadaf Fatima - 24CAMSA149
Process finished with exit code 0
```

4# Write a C++ program that reads a line of text and counts all occurrence of a particular word.

Program:

```
#include <iostream>
#include <sstream>
#include <string>
#include <algorithm>

std::string removePunctuation(const std::string& line) {
    std::string result;
    for (int i = 0; i < line.length(); i++) {
        if (!ispunct(line[i])) {
            result += line[i];
        }
    }
    return result;
}

std::string toLower(const std::string& str) {
    std::string result = str;
    transform(result.begin(), result.end(), result.begin(), ::tolower);
    return result;
}

int countOccurrences(const std::string& line, const std::string& word) {

    std::string cleanLine = removePunctuation(line);
    std::stringstream ss(cleanLine);
    std::string temp;
    int count = 0;

    while (ss >> temp) {
        if (toLower(temp) == toLower(word)) {
            count++;
        }
    }
    return count;
}

int main() {
    std::string line, word;

    std::cout << "Enter a line of text: ";
    getline(std::cin, line);
```

```

    std::cout << "Enter the word to count: ";
    std::cin >> word;

    std::cout << "The word '" << word << "' occurs " << countOccurrences(line,
word) << " times." << std::endl;

    return 0;
}

```

Console Output:

Enter a line of text:*The quick brown fox, was seen by the farmer, the fox.*

Enter the word to count:*Fox*

The word 'Fox' occurs 2 times.

Process finished with exit code 0

5# Write a C++ program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.

Program:

```

#include <iostream>
#include <string>
#include <algorithm>

std::string toLower(std::string& str){
    std::string result = str;
    transform(result.begin(), result.end(), result.begin(), ::tolower);
    return result;
}

int indexStr(std::string& sentence, std::string& toFind){
    std::string str = toLower(sentence), str_toFind = toLower(toFind);
    if(!str.find(str_toFind)) {
        return 0;
    }

    return str.find(str_toFind);
}

int main(){
    std::string str, str_toFind;

    std::cout << "Enter a sentence: ";
    getline(std::cin, str);

    std::cout << "Enter a substring to be searched in an string: ";
    getline(std::cin, str_toFind);

    (indexStr(str, str_toFind)== -1)?std::cout << "1 : String is not contained
in a sentence" : std::cout << "Index of string where substring begins: " <<
indexStr(str, str_toFind);

}

```

Console Output:

```
Enter a sentence:The lazy brown fox jumps over the dog.
```

```
Enter a substring to be searched in an string:brown fox
```

```
Index of string where substring begins: 9
```

```
Enter a sentence:The lazy brown fox jumps over the dog.
```

```
Enter a substring to be searched in an string:white
```

```
1 : String is not contained in a sentence
```

```
Process finished with exit code 0
```

WEEK – 9

PROBLEM

1# Write C++ programs that use both recursive and non-recursive functions to find:

- a) The factorial of a given integer.
- b) To find the greatest common divisor of two given integers.

Program:

```
#include <iostream>

unsigned long long int factorial_recur(int n){
    if(n==0)
        return 1;

    return n * factorial_recur(n-1);
}

unsigned long long int factorial_nonRecur(int n) {
    unsigned long long int fact = 1;
    for(int i = 0; i < n; i++){
        fact+=fact*i;
    }
    return fact;
}

int gcdRecursive(int a, int b) {
    if(a%b==0)
        return b;
    return gcdRecursive(b, a%b);
}

int gcdNonRecursive(int a, int b) {
    while(b!=0) {
        int r = a%b;
        a = b;
        b = r;
    }
    return a;
}

int main() {

    int sw;
    std::cout << "Select '1' for GCD and '2' for FACTORIAL: ";
    std::cin >> sw;

    switch (sw) {
        case 1:{
            unsigned int a, b;
            std::cout << "Enter first number: ";
            std::cin >> a;

            std::cout << "Enter second number: ";
            std::cin >> b;

            std::cout << "\nGCD of " << a << " and " << b << " using RECURSIVE "
    }
}
```

```

method: " << gcdRecursive(a, b) << std::endl;
        std::cout << "\nGCD of " << a << " and " << b << " using NON-
RECURSIVE method: " << gcdNonRecursive(a,b) << std::endl;

        break;
    }

    case 2:{
        unsigned int a;
        std::cout << "Enter a number: ";
        std::cin >> a;

        std::cout << "\nFactorial of " << a << " using RECURSIVE method: "
<< factorial_recur(a)<< std::endl;
        std::cout << "\nFactorial of " << a << " using NON-RECURSIVE method:
" << factorial_nonRecur(a) << std::endl;

        break;
    }

    default:
        std::cout << "Wrong Input...";
    }

    return 0;
}

```

Console Output:

```

Select '1' for GCD and '2' for FACTORIAL:1

Enter first number:54

Enter second number:32

GCD of 54 and 32 using RECURSIVE method: 2

GCD of 54 and 32 using NON-RECURSIVE method: 2

Process finished with exit code 0

Select '1' for GCD and '2' for FACTORIAL:2

Enter a number:5

Factorial of 5 using RECURSIVE method: 120

Factorial of 5 using NON-RECURSIVE method: 120

Process finished with exit code 0

```

2# Write C programs that use both recursive and non-recursive functions to solve towers of Hanoi problem.

Program:

```
#include <iostream>
#include <cmath>

unsigned int towerOfHanoi_Recursive(unsigned int n) {

    // Base case: Only one disk
    if (n == 1)
        return 1;

    // Recursive case: Move n-1 disks twice and add one move for the nth disk
    return (2 * towerOfHanoi_Recursive(n - 1)) + 1;
}

unsigned int towerOfHanoi_nonRecursive(unsigned int n) {

    if (n >= 1)
        return std::pow(2, n) - 1;
}

int main() {

    unsigned int n; // n==disks

    std::cout << "Enter the number of disks: ";
    std::cin >>n;

    if (n <= 0) {
        std::cout << "Atleast one disk is required...";
    }

    else{
        std::cout << "\nNumber of moves using recursive method: " <<
towerOfHanoi_Recursive(n) << std::endl;
        std::cout << "\nNumber of moves using non-recursive method: " <<
towerOfHanoi_nonRecursive(n) << std::endl;
    }

    return 0;
}
```

Console Output:

Enter the number of disks:5

Number of moves using recursive method: 31

Number of moves using non-recursive method: 31

Process finished with exit code 0

3# Write a C++ program to print the transpose of a given matrix using function.

Program:

```
#include <iostream>
#include <vector>

void transposeMatrix(int m, int n, std::vector<std::vector<int>>& matrix,
                     std::vector<std::vector<int>>& transpose) {
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            transpose[i][j] = matrix[j][i];
        }
    }
}

void insertElementMatrix(int m, int n, std::vector<std::vector<int>>& matrix) {
    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            std::cin >> matrix[i][j];
        }
        std::cout << std::endl;
    }
}

void displayElementMatrix(int m, int n, std::vector<std::vector<int>>& matrix) {
    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            std::cout << matrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

int main() {
    int m, n;

    std::cout << "Enter number of rows to be inserted in a matrix: ";
    std::cin >> m;

    std::cout << "Enter number of columns to be inserted in a matrix: ";
    std::cin >> n;

    std::vector<std::vector<int>> matrix(m, std::vector<int>(n));
    std::vector<std::vector<int>> transpose(n, std::vector<int>(m));

    std::cout << "Insert matrix elements: ";
    insertElementMatrix(m, n, matrix);

    std::cout << "\nMatrix: " << std::endl;
    displayElementMatrix(m, n, matrix);

    transposeMatrix(m, n, matrix, transpose);
    std::cout << "\nTranspose of Matrix: " << std::endl;
    displayElementMatrix(n, m, transpose);

    return 0;
}
```

Console Output:

```
Enter number of rows to be inserted in a matrix:2
Enter number of columns to be inserted in a matrix:3
Insert matrix elements:1 2 3
4 5 6
```

Matrix:

```
1 2 3
4 5 6
```

Transpose of Matrix:

```
1 4
2 5
3 6
```

Process finished with exit code 0

4# Write a C++ program to swap two number by both call by value and call by reference mechanism, using two functions swap_value() and swap_reference respectively, by getting the choice from the user and executing the user's choice by switch-case.

Program:

```
#include<iostream>

void swap_value(int a, int b) {
    std::cout << "\nNumbers before swapping: " << a << " , " << b <<
std::endl;

    int temp = a;
    a = b;
    b = temp;

    std::cout << "Numbers after swapping: " << a << " , " << b <<
std::endl;
}

void swap_reference(int& a, int& b) {
    std::cout << "\nNumbers before swapping: " << a << " , " << b <<
std::endl;

    int temp = a;
    a = b;
    b = temp;

    std::cout << "Numbers after swapping: " << a << " , " << b <<
```

```

std::endl;
}

void insertValues(int& a, int& b){
    std::cout << "\nInsert two Numbers: ";
    std::cin >> a >> b;
}

int main(){
    int a, b, sw;
    std::cout << "Select '1' for CALL BY VALUE METHOD and '2' for CALL BY
REFERENCE METHOD: ";
    std::cin >> sw;

    switch(sw){
        case 1:{
            std::cout << "\nCALL BY VALUE METHOD FOR SWAPPING TWO VARIABLES" <<
std::endl;
            insertValues(a,b);
            swap_value(a,b);
            std::cout << "\nValue in the main() after swapping: " << a << " ,
" << b;
            break;
        }
        case 2:{
            std::cout << "\nCALL BY REFERENCE METHOD FOR SWAPPING TWO VARIABLES"
<< std::endl;
            insertValues(a,b);
            swap_reference(a, b);
            std::cout << "\nValue in the main() after swapping: " << a << " ,
" << b;
            break;
        }
        default:
            std::cout << "Wrong Input...";
    }

    return 0;
}

```

Console Output:

```

Select '1' for CALL BY VALUE METHOD and '2' for CALL BY REFERENCE METHOD:1

CALL BY VALUE METHOD FOR SWAPPING TWO VARIABLES

Insert two Numbers:2 3

Numbers before swapping: 2 , 3
Numbers after swapping: 3 , 2

Value in the main() after swapping: 2 , 3

Process finished with exit code 0

```

Select '1' for CALL BY VALUE METHOD and '2' for CALL BY REFERENCE METHOD:2

CALL BY REFERENCE METHOD FOR SWAPPING TWO VARIABLES

Insert two Numbers:2 3

Numbers before swapping: 2 , 3

Numbers after swapping: 3 , 2

Value in the main() after swapping: 3 , 2

Process finished with exit code 0

5# Write a C++ program to display all array elements using recursion.

Program:

```
#include <iostream>
void displayElements(int n, int arr[], int index = 0){

    if (index == n){
        return;
    }
    std::cout << arr[index] << " ";
    displayElements(n, arr, index+1);
}

void insertElements(int n, int arr[], int index = 0){

    if (index == n){
        return;
    }
    std::cin >> arr[index] ;
    insertElements(n, arr, index+1);
}

int main(){

    int n, arr[n];

    std::cout << "Insert number of elements to be inserted in an array: ";
    std::cin >> n;

    std::cout << "Insert array elements: ";
    insertElements(n, arr);

    std::cout << "\nDisplaying array elements:";
    displayElements(n, arr);
    std::cout << std::endl;

    return 0;
}
```

Console Output:

```
Insert number of elements to be inserted in an array:5
Insert array elements:2 4 2 7 6
Displaying array elements:2 4 2 7 6
Process finished with exit code 0
```

6# Write a C++ program to find sum of elements of array using recursion.

Program:

```
#include <iostream>

int sumArrayElements(int n , int arr[]){
    if(n<=0)
        return 0;
    return arr[n-1] + sumArrayElements(n-1, arr);
}

void insertElements(int n, int arr[], int index = 0){

    if (index == n){
        return;
    }
    std::cin >> arr[index] ;
    insertElements(n, arr, index+1);
}

int main() {
    int n, arr[n];
    std::cout << "Insert number of elements to be inserted in an array: ";
    std::cin >> n;
    std::cout << "Insert array elements: ";
    insertElements(n, arr);

    std::cout << "\nSum array elements:" << sumArrayElements(n, arr);
    std::cout << std::endl;
}
```

Console Output:

```
Insert number of elements to be inserted in an array:5
Insert array elements:1 2 3 4 5
Sum array elements:15
Process finished with exit code 0
```

7# Write a C++ program to find maximum and minimum elements in array using recursion.

Program:

```
#include <iostream>

int findMax(int arr[], int n) {
    if(n==1)
        return arr[0];
    return std::max(arr[n-1], findMax(arr, n-1));
}

int findMin(int arr[], int n) {
    if(n == 1)
        return arr[0];
    return std::min(arr[n-1], findMin(arr, n-1));
}

void displayElements(int n, int arr[], int index = 0) {
    if (index == n) {
        return;
    }
    std::cout << arr[index] << " ";
    displayElements(n, arr, index+1);
}

void insertElements(int n, int arr[], int index = 0) {
    if (index == n) {
        return;
    }
    std::cin >> arr[index];
    insertElements(n, arr, index+1);
}

int main() {
    int n, arr[n];
    std::cout << "Insert number of elements to be inserted in an array: ";
    std::cin >> n;
    std::cout << "Insert array elements: ";
    insertElements(n, arr);
    std::cout << "\nDisplaying array elements:";
    displayElements(n, arr);
    std::cout << std::endl;
    std::cout << "\nMaximum Value of Element: " << findMax(arr, n) <<
    std::endl;
    std::cout << "\nMinimum Value of Element: " << findMin(arr, n) << std::endl;
    return 0;
}
```

Console Output:

```

Insert number of elements to be inserted in an array:5

Insert array elements:8 3 5 1 9

Displaying array elements:8   3   5   1   9

Maximum Value of Element: 9
Minimum Value of Element: 1

Process finished with exit code 0

```

8# Consider the insurance database given below. The primary keys are made bold and the data types are specified.

PERSON (driver_id:string , name:string , address:string)
CAR(regno:string , model:string , year:int)
ACCIDENT(report_number:int , accd_date:date , location:string)
OWNS(driver_id:string , regno:string)
PARTICIPATED (driver_id:string, regno:string, report_number:int , damage_amount:int).

- i. Create the above tables by properly specifying the primary keys and foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.
- iv. Add a new accident to the database.
- v. Find the total number of people who owned cars that were involved in accidents in the year 2008.
- vi. Find the number of accidents in which cars belonging to a specific model were involved.

Program:

```

--create tables
create table person(
    driver_id varchar2(5) not null primary key,
    name varchar2(50) not null,
    address varchar2(250) not null
);

create table car(
    regno varchar2(5) not null primary key,
    model varchar2(50) not null,
    year number(4) not null
);

create table accident(
    report_number number(5) not null primary key,

```

```

    accd_date date not null,
    location varchar2(50) not null
);

create table owns(
    driver_id varchar2(5) not null,
    regno varchar2(5) not null,
    foreign key(driver_id) references person(driver_id),
    foreign key(regno) references car(regno)
);

create table participated(
    driver_id varchar2(5) not null,
    regno varchar2(5) not null,
    report_number number(5)not null,
    damage_amount number(7) not null,
    foreign key(driver_id) references person(driver_id),
    foreign key(regno) references car(regno),
    foreign key(report_number) references accident(report_number)
);

--Insert data into tables

insert into person(driver_id, name, address) values ('D0001' , 'John', 'Aligarh');
insert      into      person(driver_id,           name,           address)      values
('D0002','David','Bulandshahr');
insert into person(driver_id, name, address) values ('D0003','Ahsan','Delhi');
insert into person(driver_id, name, address) values ('D0004','Sahil' , 'Aligarh');
insert into person(driver_id, name, address) values ('D0005', 'Mujtaba', 'Agra');

insert into car values('R0001', 'Ford', '2008');
insert into car values('R0002', 'Suzuki', '2005');
insert into car values('R0003', 'Mercedes', '2006');
insert into car values('R0004', 'Ford', '2007');
insert into car values('R0005', 'Hyundai', '2004');

insert into accident values(10,'11 nov 2008','Aligarh');
insert into accident values(11,'09 december 2009','Aligarh');
insert into accident values(12,'15 january 2008','Lucknow');
insert into accident values(13,'27 may 2007','Delhi');
insert into accident values(14,'11 nov 2008','Agra');

insert into owns values('D0001', 'R0001');
insert into owns values('D0002', 'R0002');
insert into owns values('D0003', 'R0003');
insert into owns values('D0004', 'R0004');
insert into owns values('D0005', 'R0005');

insert into participated values('D0001', 'R0001', 10, 30000);
insert into participated values('D0002', 'R0002', 11, 25000);
insert into participated values('D0003', 'R0003', 12, 10600);
insert into participated values('D0004', 'R0004', 13, 12900);
insert into participated values('D0005', 'R0005', 14, 9000);

--update the damage amount for specific accident

```

```

update participated
set damage_amount = 25000
where report_number = 12 and regno = 'R0003';

-- add new accident to database

insert into accident values(15,'13 oct 2008','Agra');
insert into participated values('D0001','R0001',15,12000);

--find total number of unique people who owned cars involved in accidents in 2008

select count(distinct O.driver_id) as no_of_people from owns O
join participated P on O.regno = p. regno
join accident A on P.report_number = A.report_number
where accd_date like '%08';

--number of accidents for specific car model

select count(distinct P.report_number) as total_accidents
from participated P
join car C on P.regno = C.regno
where C.model = 'Hyundai';

```

Console Output:

```

mysql> select * from person;
+-----+-----+-----+
| driver_id | name | address |
+-----+-----+-----+
| D0001 | John | Aligarh |
| D0002 | David | Bulandshahr |
| D0003 | Ahsan | Delhi |
| D0004 | Sahil | Aligarh |
| D0005 | Mujtaba | Agra |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from car;
+-----+-----+-----+
| regno | model | year |
+-----+-----+-----+
| R0001 | Ford | 2008 |
| R0002 | Suzuki | 2005 |
| R0003 | Mercedes | 2006 |
| R0004 | Ford | 2007 |
| R0005 | Hyundai | 2004 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> select * from accident;
+-----+-----+-----+
| report_number | accd_date | location |
+-----+-----+-----+
| 10 | 2008-11-11 | Aligarh |
| 11 | 2009-12-09 | Aligarh |
| 12 | 2008-01-15 | Lucknow |
| 13 | 2007-05-27 | Delhi |
| 14 | 2008-11-11 | Agra |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from owns;
+-----+-----+
| driver_id | regno |
+-----+-----+
| D0001 | R0001 |
| D0002 | R0002 |
| D0003 | R0003 |
| D0004 | R0004 |
| D0005 | R0005 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from participated;
+-----+-----+-----+-----+
| driver_id | regno | report_number | damage_amount |
+-----+-----+-----+-----+
| D0001 | R0001 | 10 | 30000 |
| D0002 | R0002 | 11 | 25000 |
| D0003 | R0003 | 12 | 10600 |
| D0004 | R0004 | 13 | 12900 |
| D0005 | R0005 | 14 | 9000 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> -- Update the damage amount for specific accident
mysql> UPDATE participated
      → SET damage_amount = 25000
      → WHERE report_number = 12 AND regno = 'R0003';
Query OK, 1 row affected (0.18 sec)
Rows matched: 1  Changed: 1  Warnings: 0

```

```

mysql> select * from participated;
+-----+-----+-----+-----+
| driver_id | regno | report_number | damage_amount |
+-----+-----+-----+-----+
| D0001 | R0001 | 10 | 30000 |
| D0002 | R0002 | 11 | 25000 |
| D0003 | R0003 | 12 | 25000 |
| D0004 | R0004 | 13 | 12900 |
| D0005 | R0005 | 14 | 9000 |
+-----+-----+-----+-----+
5 rows in set (0.02 sec)

```

```

mysql> insert into accident values(15, '2008-10-13', 'Agra');
Query OK, 1 row affected (0.06 sec)

mysql> insert into participated values('D0001','R0001',15,12000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from accident;
+-----+-----+-----+
| report_number | accd_date | location |
+-----+-----+-----+
|      10 | 2008-11-11 | Aligarh
|      11 | 2009-12-09 | Aligarh
|      12 | 2008-01-15 | Lucknow
|      13 | 2007-05-27 | Delhi
|      14 | 2008-11-11 | Agra
|      15 | 2008-10-13 | Agra
+-----+-----+-----+
6 rows in set (0.04 sec)

mysql> select* from participated;
+-----+-----+-----+-----+
| driver_id | regno | report_number | damage_amount |
+-----+-----+-----+-----+
| D0001    | R0001 |          10 |      30000
| D0002    | R0002 |          11 |      25000
| D0003    | R0003 |          12 |      25000
| D0004    | R0004 |          13 |      12900
| D0005    | R0005 |          14 |       9000
| D0001    | R0001 |          15 |      12000
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

```

mysql> SELECT COUNT(DISTINCT O.driver_id) AS no_of_people
   → FROM owns O
   → JOIN participated P ON O.regno = P.regno
   → JOIN accident A ON P.report_number = A.report_number
   → WHERE YEAR(A.accd_date) = 2008;
+-----+
| no_of_people |
+-----+
|          3 |
+-----+
1 row in set (0.08 sec)

```

```

mysql> -- Number of accidents for specific car model
mysql> SELECT COUNT(DISTINCT P.report_number) AS total_accidents
   → FROM participated P
   → JOIN car C ON P.regno = C.regno
   → WHERE C.model = 'Hyundai';
+-----+
| total_accidents |
+-----+
|          1 |
+-----+
1 row in set (0.16 sec)

```

WEEK – 10

PROBLEM

1# Write a C++ program that uses functions to perform the following operations:

- i. Reading a complex number
- ii. Writing a complex number
- iii. Addition and subtraction of two complex numbers
- iv. Multiplication of two complex numbers. Note: represent complex number using a structure.

Program:

```
#include <iostream>

struct ComplexNum {
    int real;
    int imaginary;
};

void ReadComplexNum(ComplexNum &num) {
    std::cout << "Enter the real part of the complex number: ";
    std::cin >> num.real;
    std::cout << "Enter the imaginary part of the complex number: ";
    std::cin >> num.imaginary;
}

void DisplayComplexNum(const ComplexNum &num) {
    if (num.imaginary < 0) {
        std::cout << "\nYour Entered Number: " << num.real << num.imaginary << "i" << std::endl;
    }
    else if (num.real == 0 && num.imaginary == 0) {
        std::cout << "\nYour Entered Number: " << 0 << std::endl;
    }
    else if (num.real == 0) {
        std::cout << "\nYour Entered Number: " << num.imaginary << "i" << std::endl;
    }
    else if (num.imaginary == 0) {
        std::cout << "\nYour Entered Number: " << num.real << std::endl;
    }
    else {
        std::cout << "\nYour Entered Number: " << num.real << "+" << num.imaginary << "i" << std::endl;
    }
}

ComplexNum MulComplexNum(const ComplexNum &num1, const ComplexNum &num2) {
    ComplexNum result;
    result.real = (num1.real * num2.real) - (num1.imaginary * num2.imaginary);
    result.imaginary = (num1.real * num2.imaginary) + (num2.real * num1.imaginary);
    return result;
}

ComplexNum AddComplexNum(const ComplexNum &num1, const ComplexNum &num2) {
```

```

ComplexNum result;
result.real = num1.real + num2.real;
result.imaginary = num1.imaginary + num2.imaginary;
return result;
}

ComplexNum SubComplexNum(const ComplexNum &num1, const ComplexNum &num2) {
    ComplexNum result;
    result.real = num1.real - num2.real;
    result.imaginary = num1.imaginary - num2.imaginary;
    return result;
}

void DisplayResult(const ComplexNum &result, const std::string &operation) {
    if (result.imaginary < 0) {
        std::cout << "\nResult after " << operation << ":" << result.real <<
result.imaginary << "i" << std::endl;
    }
    else if (result.real == 0 && result.imaginary == 0) {
        std::cout << "\nResult after " << operation << ":" << 0 << std::endl;
    }
    else if (result.real == 0) {
        std::cout << "\nResult after " << operation << ":" << result.imaginary
<< "i" << std::endl;
    }
    else if (result.imaginary == 0) {
        std::cout << "\nResult after " << operation << ":" << result.real <<
std::endl;
    }
    else {
        std::cout << "\nResult after " << operation << ":" << result.real <<
"+" << result.imaginary << "i" << std::endl;
    }
}

int main() {
    ComplexNum num1, num2;
    ReadComplexNum(num1);
    ReadComplexNum(num2);
    DisplayComplexNum(num1);
    DisplayComplexNum(num2);

    ComplexNum mulResult = MulComplexNum(num1, num2);
    DisplayResult(mulResult, "Multiplication");

    ComplexNum addResult = AddComplexNum(num1, num2);
    DisplayResult(addResult, "Addition");

    ComplexNum subResult = SubComplexNum(num1, num2);
    DisplayResult(subResult, "Subtraction");

    return 0;
}

```

Console Output:

```

Enter the real part of the complex number:6
Enter the imaginary part of the complex number:2
Enter the real part of the complex number:5
Enter the imaginary part of the complex number:-2

Your Entered Number: 6+2i
Your Entered Number: 5-2i
Result after Multiplication: 34-2i
Result after Addition: 11
Result after Subtraction: 1+4i
Process finished with exit code 0

```

2# Write a C++ program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.

Program:

```

#include<iostream>
#include<string>
#include <vector>

struct Employee{
    std::string name;
    double basicPay;
    double GrossSalary;
};

void InsertEmpDetails(Employee &emp) {
    std::cout << "Enter employee Name: ";
    std::cin.ignore();
    getline(std::cin, emp.name);
    std::cout << "Enter Basic Pay: ";
    std::cin >> emp.basicPay;
}

void GrossSalary(Employee &emp) {
    double DA = 0.52 * emp.basicPay;
    emp.GrossSalary = emp.basicPay + DA;
}

```

```

}

void DisplayEmployeeDetails(const Employee &emp) {
    std::cout << "Employee Name: " << emp.name << ", Gross Salary: " <<
emp.GrossSalary << std::endl;
}

int main(){

    const int EmpNum = 5;
    std::vector<Employee> employee(EmpNum);

    //Insert Employee DisplayEmployeeDetails
    for(int i = 0 ; i < EmpNum ; i++){
        std::cout << "\nInsert Details for Employee" << i+1 << std::endl;
        InsertEmpDetails(employee[i]);
        GrossSalary(employee[i]);
    }

    //Display Details

    std::cout << "-----Gross Salary of Employees-----" << std::endl;
    std::cout << std::endl;

    for(int i = 0; i < EmpNum ; i++){
        DisplayEmployeeDetails(employee[i]);
    }
    return 0;
}

```

Console Output:

Enter number of employees data you want to insert:2

Insert Details for Employee1

Enter employee Name:*Sadaf Fatima*

Enter Basic Pay:*32000*

Insert Details for Employee2

Enter employee Name:*Saud Ahmad Khan*

Enter Basic Pay:*45000*

-----Gross Salary of Employees-----

Employee Name: Sadaf Fatima, Gross Salary: 48640

Employee Name: Saud Ahmad Khan, Gross Salary: 68400

Process finished with exit code 0

3# Create a Book structure containing book_id, title, author name and price. Write a C++ program to pass a structure as a function argument and print the book details.

Program:

```
#include<iostream>

struct Book{
    int book_id;
    std::string title;
    std::string author_name;
    double price;
};

void InsertBookDetails(Book &book) {

    std::cout << "Insert Book ID: ";
    std::cin>>book.book_id;

    std::cin.ignore();

    std::cout << "Insert Book Title: ";
    getline(std::cin, book.title);
    std::cout << "Insert Book Author: ";
    getline(std::cin, book.author_name);
    std::cout << "Insert Book Price: ";
    std::cin >> book.price;

}

void DisplayBookDetails(Book &book) {

    std::cout << "Book Id: " << book.book_id << std::endl;
    std::cout << "Book Title: " << book.title << std::endl;
    std::cout << "Book Author: " << book.author_name << std::endl;
    std::cout << "Book Price: " << book.price << std::endl;

    std::cout << std::endl;
}

int main(){

    Book book1, book2;

    std::cout << "Enter details for BOOK1" << std::endl;
    InsertBookDetails(book1);
    std::cout << "\nEnter details for BOOK2" << std::endl;
    InsertBookDetails(book2);

    std::cout << "\nBOOK1" << std::endl;
    DisplayBookDetails(book1);
    std::cout << "\nBOOK2" << std::endl;
    DisplayBookDetails(book2);

    return 0;
}
```

Console Output:

```
Enter details for BOOK1
```

```
Insert Book ID:1
```

```
Insert Book Title:The Pianist
```

```
Insert Book Author:Wladyslaw Szpilman
```

```
Insert Book Price:375
```

```
Enter details for BOOK2
```

```
Insert Book ID:2
```

```
Insert Book Title:Night
```

```
Insert Book Author:Elie Wiesel
```

```
Insert Book Price:499
```

```
BOOK1
```

```
Book Id: 1
```

```
Book Title: The Pianist
```

```
Book Author: Wladyslaw Szpilman
```

```
Book Price: 375
```

```
BOOK2
```

```
Book Id: 2
```

```
Book Title: Night
```

```
Book Author: Elie Wiesel
```

```
Book Price: 499
```

```
Process finished with exit code 0
```

4# Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C++ program to display your present address.

Program:

```
#include <iostream>
#include <cstring>

struct Address {
    char name[50];
    char home_address[100];
    char hostel_address[100];
    char city[50];
    char state[50];
    char zip[10];
};

void InsertDetails(Address &add) {
    std::cout << "Insert your name: ";
    std::cin.getline(add.name, 50);

    std::cout << "Insert your present home address (or skip if present address is hostel address): ";
    std::cin.getline(add.home_address, 100);

    std::cout << "Insert your present hostel address: (or skip if present address is hostel address):";
    std::cin.getline(add.hostel_address, 100);

    std::cout << "Insert your present city: ";
    std::cin.getline(add.city, 50);

    std::cout << "Insert your state: ";
    std::cin.getline(add.state, 50);

    std::cout << "Insert your ZIP: ";
    std::cin.getline(add.zip, 10);
}

void DisplayAddress(const Address &add) {
    std::cout << "\nPresent Address of " << add.name << std::endl;
    if (strlen(add.hostel_address) == 0) {
        std::cout << add.home_address << ", " << add.city << ", " << add.state
        << ", " << add.zip << std::endl;
    } else {
        std::cout << add.hostel_address << ", " << add.city << ", " << add.state
        << ", " << add.zip << std::endl;
    }
}

int main() {
    Address add1, add2;

    std::cout << "Enter details for Address 1:" << std::endl;
    InsertDetails(add1);

    std::cout << "\nEnter details for Address 2:" << std::endl;
    InsertDetails(add2);

    std::cout << "\n--- Displaying Addresses ---" << std::endl;
    DisplayAddress(add1);
    DisplayAddress(add2);
}
```

```
    return 0;  
}
```

Console Output:

Enter details for Address 1:

Insert your name:*Sadaf Fatima*

Insert your present home address (or skip if present address is hostel address):*Dhorra*

Insert your present hostel address: (or skip if present address is hostel address):

Insert your present city:*Aligarh*

Insert your state:*Uttar Pradesh*

Insert your ZIP:*202002*

Enter details for Address 2:

Insert your name:*Sadaf Fatima*

Insert your present home address (or skip if present address is hostel address):

Insert your present hostel address: (or skip if present address is hostel address):*Aligarh Muslim University*

Insert your present city:*Aligarh*

Insert your state:*Uttar Pradesh*

Insert your ZIP:*202002*

--- Displaying Addresses ---

Present Address of Sadaf Fatima

Dhorra, Aligarh, Uttar Pradesh, 202002

Present Address of Sadaf Fatima

Aligarh Muslim University, Aligarh, Uttar Pradesh, 202002

Process finished with exit code 0

5# Write a C++ program to define a structure named D.O.B., which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.

Program:

```
#include <iostream>
#include <cstring>

struct DOB{
    std::string name;
    int day;
    int month;
    int year;
};

struct person{
    std::string name;
    DOB dob;
};

void InsertDetails(person &p) {
    std::cout << "Enter your Name: ";
    getline(std::cin , p.name);

    std::cout << "Enter your date of birth (day month year): ";
    std::cin >> p.dob.day >> p.dob.month >> p.dob.year;
}

void displayDetails(person &p) {
    std::cout << "\nDate of Birth of " << p.name << " is " << p.dob.day << "/" <<
p.dob.month << "/" << p.dob.year;
    std::cout << std::endl;
}

int main(){
    person p1;
    InsertDetails(p1);
    displayDetails(p1);

    return 0;
}
```

Console Output:

```
Enter your Name:Sadaf Fatima

Enter your date of birth (day month year):21 07 2002

Date of Birth of Sadaf Fatima is 21/7/2002

Process finished with exit code 0
```

6# Consider the following database for a banking enterprise.

BRANCH (branch_name:string , branch_city:string , assets:real)
 ACCOUNT(accno:int , branch_name:string , balance:real)
 DEPOSITOR(customer_name:string , accno:int)
 CUSTOMER (customer_name:string, customer_street:string , customer_city: string)
 LOAN (loan_number:int , branch_name:string , amount:real)
 BORROWER(customer_name:string , loan_number:int)

- i. Create the above tables by properly specifying the primary keys and foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Find all the customers who have at least two accounts at the main branch.
- iv. Find all the customers who have an account at all the branches located in a specific city.
- v. Demonstrate how you delete all account tuples at every branch located in a specific city.

Program:

```
--create table by specifying primary key and foreign key
CREATE TABLE branch(
branch_name VARCHAR2(20),
branch_city VARCHAR2(20),
balance REAL,
PRIMARY KEY(branch_name)) ;

CREATE TABLE account(
accno NUMBER(10),
branch_name VARCHAR2(20),
balance REAL,
PRIMARY KEY(accno),
FOREIGN KEY (branch_name) REFERENCES branch(branch_name) ON DELETE CASCADE);

CREATE TABLE customer(
customer_name VARCHAR2(20),
customer_street VARCHAR2(20),
customer_city VARCHAR2(20),
PRIMARY KEY(customer_name));

CREATE TABLE depositor(
customer_name VARCHAR2(20),
accno NUMBER(10),
PRIMARY KEY(accno),
FOREIGN KEY(customer_name) REFERENCES customer(customer_name) ON DELETE CASCADE,
FOREIGN KEY(accno) REFERENCES account(accno) ON DELETE CASCADE);

CREATE TABLE loan(
loan_number NUMBER(15),
branch_name VARCHAR2(20),
amount REAL,
PRIMARY KEY(loan_number),
FOREIGN KEY (branch_name) REFERENCES branch(branch_name) ON DELETE CASCADE);
```

```

CREATE TABLE borrower(
customer_name VARCHAR2(20),
loan_number NUMBER(15),
PRIMARY KEY(customer_name,loan_number),
FOREIGN KEY(customer_name) REFERENCES customer(customer_name) ON DELETE CASCADE,
FOREIGN KEY(loan_number) REFERENCES loan(loan_number) ON DELETE CASCADE);

--insert record in a tables
-- Insert new records with different values

INSERT INTO branch VALUES('Axis','Mumbai',15000);
INSERT INTO branch VALUES('Bank of Baroda','Pune',25000);
INSERT INTO branch VALUES('IDFC','Bangalore',35000);
INSERT INTO branch VALUES('Yes Bank','Chennai',45000);
INSERT INTO branch VALUES('Kotak','Hyderabad',55000);

INSERT INTO account VALUES(15,'Axis',3500);
INSERT INTO account VALUES(25,'Bank of Baroda',4500);
INSERT INTO account VALUES(35,'IDFC',5500);
INSERT INTO account VALUES(45,'Yes Bank',6500);
INSERT INTO account VALUES(55,'Kotak',7500);

INSERT INTO customer VALUES('Ali','B17','Mumbai');
INSERT INTO customer VALUES('Zara','A14','Pune');
INSERT INTO customer VALUES('Omar','C20','Bangalore');
INSERT INTO customer VALUES('Fatima','D22','Chennai');
INSERT INTO customer VALUES('Ahmed','E25','Hyderabad');

INSERT INTO depositor VALUES('Ali',15);
INSERT INTO depositor VALUES('Zara',25);
INSERT INTO depositor VALUES('Omar',35);
INSERT INTO depositor VALUES('Fatima',45);
INSERT INTO depositor VALUES('Ahmed',55);

INSERT INTO loan VALUES(60,'Axis',15000);
INSERT INTO loan VALUES(70,'Bank of Baroda',25000);
INSERT INTO loan VALUES(80,'IDFC',35000);
INSERT INTO loan VALUES(90,'Yes Bank',45000);
INSERT INTO loan VALUES(100,'Kotak',55000);

INSERT INTO borrower VALUES('Ali',60);
INSERT INTO borrower VALUES('Zara',70);
INSERT INTO borrower VALUES('Omar',80);
INSERT INTO borrower VALUES('Fatima',90);
INSERT INTO borrower VALUES('Ahmed',100);

--find all customers who have at least two accounts in a main branch
SELECT customer.customer_name
      FROM account a,branch b,depositor d,customer c
     WHERE b.branch_name=a.branch_name AND
           a.accno=d.accno AND c.customer_name=d.customer_name and
           b.branch_city='Delhi'
    GROUP BY c.customer_name
   HAVING COUNT(distinct b.branch_name)=(
      SELECT COUNT(branch_name)
         FROM branch WHERE branch_city='Mumbai');

```

```
--find all the customers who have an account at all the branches located in a
specific city
select customer_name
from account,depositor
where depositor.accno=account.accno and account.branch_name='Mumbai'
group by depositor.CUSTOMER_NAME
having count(depositor.customer_name)>1;

--demonstrate how you delete all accounts tuples at every branch located in a
specific city
DELETE FROM account WHERE branch_name IN(SELECT branch_name FROM branch WHERE
branch_city='Mumbai');
```

Console Output:

```
mysql> SELECT*FROM branch;
+-----+-----+-----+
| branch_name | branch_city | balance |
+-----+-----+-----+
| Axis          | Mumbai      | 15000   |
| Bank of Baroda | Pune        | 25000   |
| IDFC          | Bangalore   | 35000   |
| Kotak          | Hyderabad  | 55000   |
| Yes Bank       | Chennai     | 45000   |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT*FROM account;
+-----+-----+-----+
| accno | branch_name | balance |
+-----+-----+-----+
| 15    | Axis          | 3500    |
| 25    | Bank of Baroda | 4500    |
| 35    | IDFC          | 5500    |
| 45    | Yes Bank       | 6500    |
| 55    | Kotak          | 7500    |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT*FROM customer;
+-----+-----+-----+
| customer_name | customer_street | customer_city |
+-----+-----+-----+
| Ahmed          | E25           | Hyderabad   |
| Ali            | B17           | Mumbai      |
| Fatima         | D22           | Chennai     |
| Omar           | C20           | Bangalore  |
| Zara           | A14           | Pune        |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT*FROM depositor;
+-----+-----+
| customer_name | accno |
+-----+-----+
| Ahmed          | 55    |
| Ali            | 15    |
| Fatima         | 45    |
| Omar           | 35    |
| Zara           | 25    |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT*FROM loan;
+-----+-----+-----+
| loan_number | branch_name | amount |
+-----+-----+-----+
| 60 | Axis | 15000 |
| 70 | Bank of Baroda | 25000 |
| 80 | IDFC | 35000 |
| 90 | Yes Bank | 45000 |
| 100 | Kotak | 55000 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT*FROM borrower;
+-----+-----+
| customer_name | loan_number |
+-----+-----+
| Ali | 60 |
| Zara | 70 |
| Omar | 80 |
| Fatima | 90 |
| Ahmed | 100 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT c.customer_name
    → FROM account a
    → JOIN branch b ON b.branch_name = a.branch_name
    → JOIN depositor d ON a.accno = d.accno
    → JOIN customer c ON c.customer_name = d.customer_name
    → WHERE b.branch_city = 'Mumbai'
    → GROUP BY c.customer_name
    → HAVING COUNT(DISTINCT a.branch_name) > 1;
Empty set (0.22 sec)
```

```
mysql> SELECT d.customer_name
    → FROM depositor d
    → JOIN account a ON d.accno = a.accno
    → WHERE a.branch_name IN (SELECT branch_name FROM branch WHERE branch_city = 'Mumbai')
    → GROUP BY d.customer_name
    → HAVING COUNT(DISTINCT a.branch_name) = (SELECT COUNT(*) FROM branch WHERE branch_city = 'Mumbai');
+-----+
| customer_name |
+-----+
| Ali |
+-----+
1 row in set (0.07 sec)
```

```
mysql> DELETE FROM account WHERE branch_name IN (SELECT branch_name FROM branch WHERE branch_city = 'Mumbai');
Query OK, 1 row affected (0.10 sec)
```

```
mysql> select * from account;
+-----+-----+-----+
| accno | branch_name | balance |
+-----+-----+-----+
| 25 | Bank of Baroda | 4500 |
| 35 | IDFC | 5500 |
| 45 | Yes Bank | 6500 |
| 55 | Kotak | 7500 |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

WEEK – 11

PROBLEM

1# Write a program in C++ to display your name, Branch, Year on to the computer screen without using classes and object. All information should be displayed in the separate line.

Program:

```
#include <iostream>

int main() {
    std::string name, branch;
    int year;

    std::cout << "Enter your name: ";
    getline(std::cin, name);

    std::cout << "Enter your branch: ";
    getline(std::cin, branch);

    std::cout << "Enter your year: ";
    std::cin >> year;

    std::cout << "Name: " << name << std::endl;
    std::cout << "Branch: " << branch << std::endl;
    std::cout << "Year: " << year << " Year" << std::endl;

    return 0;
}
```

Console Output:

Enter your name:*Sadaf Fatima*

Enter your branch:*MCA*

Enter your year:*1*

Name: Sadaf Fatima

Branch: MCA

Year: 1 Year

Process finished with exit code 0

2# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

Program:

```
#include <iostream>

void operation(int num1, int num2) {
    double result;
    int choice;

    std::cout << "Enter two numbers: ";
    std::cin >> num1 >> num2;

    std::cout << "\nChoose the operation:\n";
    std::cout << "1. Addition\n";
    std::cout << "2. Subtraction\n";
    std::cout << "3. Multiplication\n";
    std::cout << "4. Division\n";
    std::cout << "Enter your choice (1-4): ";
    std::cin >> choice;

    switch (choice) {
        case 1:
            result = num1 + num2;
            std::cout << "Result (Addition): " << result << std::endl;
            break;
        case 2:
            result = num1 - num2;
            std::cout << "Result (Subtraction): " << result << std::endl;
            break;
        case 3:
            result = num1 * num2;
            std::cout << "Result (Multiplication): " << result << std::endl;
            break;
        case 4:
            if (num2 != 0) {
                result = num1 / num2;
                std::cout << "Result (Division): " << result << std::endl;
            } else {
                std::cout << "Error: Division by zero is not allowed." <<
                std::endl;
            }
            break;
        default:
            std::cout << "Invalid choice! Please choose a valid operation." <<
            std::endl;
    }
}

int main() {
    int num1, num2;
    operation(num1, num2);
    return 0;
}
```

Console Output:

```
Enter two numbers:2 4

Choose the operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice (1-4):1

Result (Addition): 6

Process finished with exit code 0
```

3# Write a menu driven program in C++ that receives 4-digit integer value from the keyboard and perform following operations:

- i) Reverse of that no.
- ii) sum of number with its reverse.
- iii) sum of alternative digits (1 digit+3 digit and 2 digit+4 digit)

Program:

```
#include <iostream>

int reverseNumber(int num) {
    int reversed = 0;
    while (num > 0) {
        reversed = reversed * 10 + num % 10;
        num /= 10;
    }
    return reversed;
}

int sumWithReverse(int num) {
    return reverseNumber(num)+num;
}

void sumOfAlternativeDigits(int num) {
    int oddSum = 0;
    int evenSum = 0;
    int i = 1;
    while (num!=0) {
        int r = num%10;
        if(i%2!=0) {
            oddSum+=r;
        }
        else{
            evenSum+=r;
        }
        i++;
        num = num/10;
    }

    std::cout << "Sum of even alternative digits: " << evenSum << std::endl;
}
```

```

    std::cout << "Sum of odd alternative digits: " << oddSum << std::endl;
}

void performOperation(int choice, int num) {
    std::cout << "Enter a 4-digit integer: ";
    std::cin >> num;

    if (num < 1000 || num > 9999) {
        std::cout << "Error: Please enter a valid 4-digit number." << std::endl;
        return;
    }

    std::cout << "\nChoose an operation:\n";
    std::cout << "1. Reverse the number\n";
    std::cout << "2. Sum of the number with its reverse\n";
    std::cout << "3. Sum of alternative digits (1st+3rd and 2nd+4th)\n";
    std::cout << "Enter your choice (1-3): ";
    std::cin >> choice;
    switch (choice) {
        case 1:
            std::cout << "Reverse of the number: " << reverseNumber(num) <<
std::endl;
            break;
        case 2:
            std::cout << "Sum of the number with its reverse: " <<
sumWithReverse(num) << std::endl;
            break;
        case 3:
            sumOfAlternativeDigits(num);
            break;
        default:
            std::cout << "Invalid choice! Please select a valid option." <<
std::endl;
    }
}

int main() {
    int num, choice;

    performOperation(choice, num);

    return 0;
}

```

Console Output:

Enter a 4-digit integer:**2345**

Choose an operation:
1. Reverse the number
2. Sum of the number with its reverse
3. Sum of alternative digits (1st+3rd and 2nd+4th)
Enter your choice (1-3):**1**

Reverse of the number: 5432

Process finished with exit code 0

```
Enter a 4-digit integer:4521
```

Choose an operation:

1. Reverse the number
2. Sum of the number with its reverse
3. Sum of alternative digits (1st+3rd and 2nd+4th)

Enter your choice (1-3):2

```
Sum of the number with its reverse: 5775
```

```
Process finished with exit code 0
```

```
Enter a 4-digit integer:1234
```

Choose an operation:

1. Reverse the number
2. Sum of the number with its reverse
3. Sum of alternative digits (1st+3rd and 2nd+4th)

Enter your choice (1-3):3

```
Sum of even alternative digits: 4
```

```
Sum of odd alternative digits: 6
```

```
Process finished with exit code 0
```

4# Write a menu driven program in C++ to receive integer number and convert equivalent binary, octal, hexadecimal number.

Program:

```
#include <iostream>
#include <string>

void DecimalToBinary(int num) {
    if (num == 0) {
        std::cout << "Number after conversion to Binary: 0" << std::endl;
        return;
    }

    std::string binary;

    while (num != 0) {
        int r = num % 2;
        binary = std::to_string(r) + binary;
        num = num / 2;
    }

    std::cout << "Number after conversion to Binary: " << binary << std::endl;
}

void DecimalToOctal(int num) {
    if (num == 0) {
        std::cout << "Number after conversion to Octal: 0" << std::endl;
    }
}
```

```

        return;
    }

    std::string octal;

    while (num != 0) {
        int r = num % 8;
        octal = std::to_string(r) + octal;
        num = num / 8;
    }

    std::cout << "Number after conversion to Octal: " << octal << std::endl;
}

void DecimalToHexadecimal(int num) {
    if (num == 0) {
        std::cout << "Number after conversion to Hexadecimal: 0" << std::endl;
        return;
    }

    std::string hexa;

    while (num != 0) {
        int r = num % 16;

        if (r == 10) {
            hexa += 'A';
        } else if (r == 11) {
            hexa += 'B';
        } else if (r == 12) {
            hexa += 'C';
        } else if (r == 13) {
            hexa += 'D';
        } else if (r == 14) {
            hexa += 'E';
        } else if (r == 15) {
            hexa += 'F';
        } else {
            hexa = std::to_string(r) + hexa;
        }

        num = num / 16;
    }

    std::cout << "Number after conversion to Hexadecimal: " << hexa <<
    std::endl;
}

void performOperation(int choice, int num) {
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << "\nChoose an operation:\n";
    std::cout << "1. Decimal to Binary Conversion\n";
    std::cout << "2. Decimal to Octal Conversion\n";
    std::cout << "3. Decimal to Hexadecimal Conversion\n";
    std::cout << "Enter your choice (1-3): ";
    std::cin >> choice;

    switch (choice) {
        case 1:
            DecimalToBinary(num);
            break;
        case 2:
    }
}

```

```

        DecimalToOctal(num);
        break;
    case 3:
        DecimalToHexadecimal(num);
        break;
    default:
        std::cout << "Invalid choice! Please select a valid option." <<
std::endl;
    }
}

int main() {
    int num, choice;

    performOperation(choice, num);

    return 0;
}

```

Console Output:

Enter a number:543

Choose an operation:
1. Decimal to Binary Conversion
2. Decimal to Octal Conversion
3. Decimal to Hexadecimal Conversion
Enter your choice (1-3):1

Number after conversion to Binary: 10000011111

Process finished with exit code 0

Enter a number:324

Choose an operation:
1. Decimal to Binary Conversion
2. Decimal to Octal Conversion
3. Decimal to Hexadecimal Conversion
Enter your choice (1-3):2

Number after conversion to Octal: 504

Process finished with exit code 0

Enter a number:587

Choose an operation:
1. Decimal to Binary Conversion
2. Decimal to Octal Conversion
3. Decimal to Hexadecimal Conversion
Enter your choice (1-3):3

Number after conversion to Hexadecimal: 24B

Process finished with exit code 0

5# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values using function and switch case. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

Program:

```
#include <iostream>

double add(double a, double b) {
    return a + b;
}

double subtract(double a, double b) {
    return a - b;
}

double multiply(double a, double b) {
    return a * b;
}

double divide(double a, double b) {
    if (b != 0) {
        return a / b;
    } else {
        std::cout << "Error: Division by zero is not allowed." << std::endl;
        return 0;
    }
}

void performOperation(int choice, double num1, double num2) {
    std::cout << "Enter first number: ";
    std::cin >> num1;

    std::cout << "Enter second number: ";
    std::cin >> num2;

    std::cout << "\nChoose the operation:\n";
    std::cout << "1. Addition\n";
    std::cout << "2. Subtraction\n";
    std::cout << "3. Multiplication\n";
    std::cout << "4. Division\n";
    std::cout << "Enter your choice (1-4): ";
    std::cin >> choice;
    double result;
    switch (choice) {
        case 1:
            result = add(num1, num2);
            std::cout << "Result (Addition): " << result << std::endl;
            break;
        case 2:
            result = subtract(num1, num2);
            std::cout << "Result (Subtraction): " << result << std::endl;
            break;
        case 3:
            result = multiply(num1, num2);
            std::cout << "Result (Multiplication): " << result << std::endl;
            break;
        case 4:
            result = divide(num1, num2);
            if (num2 != 0) {
                std::cout << "Result (Division): " << result << std::endl;
            }
    }
}
```

```

        break;
    default:
        std::cout << "Invalid choice! Please choose a valid operation." <<
std::endl;
    }
}

int main() {
    double num1, num2;
    int choice;

    performOperation(choice, num1, num2);

    return 0;
}

```

Console Output:

Enter first number:54

Enter second number:21

Choose the operation:

1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice (1-4):4

Result (Division): 2.57143

Process finished with exit code 0

6# Define a class Bank Account to represent a bank account. Include the following members:

Data Members: Name of the depositor , Account Number , Type of account , Balance amount
Member Functions: To assign initial value , To deposit an amount , To withdraw an amount after checking.

Program:

```

#include <iostream>

class BankAccount{

private:
    std::string name;
    int accountNumber;
    std::string accountType;
    double balance;

public:
    void setDetails() {

```

```

    std::cout << "Enter customer Name: ";
    std::cin >> name;

    std::cout << "Enter Customer Account Number: ";
    std::cin >> accountNumber;

    std::cout << "Enter Customer Account Type: ";
    std::cin >> accountType;

    std::cout << "Enter Initial Balance: ";
    std::cin >> balance;
}

BankAccount() {
    name = "";
    accountNumber = 0;
    accountType = "";
    balance = 0.0;
}

void DisplayAccountDetails() {
    std::cout << "Name: " << name << std::endl;
    std::cout << "Account Number: " << accountNumber << std::endl;
    std::cout << "Account Type: " << accountType << std::endl;
    std::cout << "Balance: " << balance << std::endl;
}

void depositAmount() {
    double amount;
    std::cout << "\nEnter amount to be deposited: ";
    std::cin >> amount;

    if (amount > 0) {
        balance += amount;
        std::cout << "New Balance after depositing money: " << balance <<
std::endl;
    } else {
        std::cout << "Invalid deposit amount!" << std::endl;
    }
}

void withdrawAmount() {
    double amount;
    std::cout << "\nEnter amount to withdraw: ";
    std::cin >> amount;

    if (amount > 0) {
        if (balance >= amount) {
            balance -= amount;
            std::cout << "New Balance after withdrawing money: " << balance <<
std::endl;
        } else {
            std::cout << "Transaction Failed!! Insufficient balance" <<
std::endl;
        }
    } else {
        std::cout << "Invalid withdrawal amount!" << std::endl;
    }
}

int main() {

    BankAccount customer1;
    std::cout << "Enter details of CUSTOMER 1" << std::endl;
}

```

```

customer1.setDetails();
customer1.DisplayAccountDetails();
customer1.depositAmount();
customer1.withdrawAmount();
}

```

Console Output:

```

Enter details of CUSTOMER 1
Enter customer Name:Sadaf Fatima

Enter Customer Account Number:654321876

Enter Customer Account Type:Saving

Enter Initial Balance:32765

Name: Sadaf Fatima
Account Number: 654321876
Account Type: Saving
Balance: 32765

Enter amount to be deposited:10000

New Balance after depositing money: 42765

Enter amount to withdraw:25000

New Balance after withdrawing money: 17765

Process finished with exit code 0

```

7# Write a C++ program to implement the Bubble Sort algorithm to sort an array of integers in ascending order.

Program:

```

#include<iostream>

void BubbleSort(int n, int array[]){
    for(int i = 0; i < n-1; i++){
        for(int j = 0; j < n-1; j++){
            if(array[j] > array[j+1]){
                int temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
            }
        }
    }
}

void insertArrayElements(int n, int array[]){
    std::cout << "Insert Array Elements: ";

```

```

        for(int i = 0; i <n; i++){
            std::cin >> array[i];
        }

void SortedArrayElements(int n , int array[]){
    BubbleSort(n, array);
    std::cout << "Sorted Array Elements in ascending Order: ";
    for(int i =0 ; i < n;i++){
        std::cout << array[i] << "   ";
    }
    std::cout << std::endl;
}

int main(){
    int n, array[n];

    std::cout << "Enter number of elements to be inserted in array: ";
    std::cin >> n;

    insertArrayElements(n, array);
    SortedArrayElements(n, array);

}

```

Console Output:

```

Enter number of elements to be inserted in array:5

Insert Array Elements:2 1 8 4 3

Sorted Array Elements in ascending Order: 1   2   3   4   8

Process finished with exit code 0

```

8# Write a C++ program to implement the Binary Search algorithm on a sorted array of integers and return the index of a target element.

Program:

```

#include <iostream>

void BubbleSort(int n, int array[] ) {
    for (int i = 0; i < n - 1; i++) {
        bool swapped = false;
        for (int j = 0; j < n - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
                swapped = true;
            }
        }
        if (!swapped) break;
    }
}

int BinarySearch(int n, int array[], int item) {

```

```

int begin = 0, end = n - 1;

while (begin <= end) {
    int mid = begin + (end - begin) / 2;
    if (array[mid] == item) {
        return mid;
    } else if (item > array[mid]) {
        begin = mid + 1;
    } else {
        end = mid - 1;
    }
}
return -1;
}

int main() {
    int n;

    std::cout << "Enter number of elements to be inserted in array: ";
    std::cin >> n;

    int array[n];

    std::cout << "Insert Array Elements: ";
    for (int i = 0; i < n; i++) {
        std::cin >> array[i];
    }

    BubbleSort(n, array);

    int item;
    std::cout << "Enter item to be searched in an array: ";
    std::cin >> item;

    int ItemIdx = BinarySearch(n, array, item);

    if (ItemIdx == -1) {
        std::cout << "Element does not exist in the array." << std::endl;
    } else {
        std::cout << "Element " << item << " is found at index(es): ";
        for (int i = ItemIdx; i < n && array[i] == item; i++) {
            std::cout << i << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}

```

Console Output:

Enter number of elements to be inserted in array:5

Insert Array Elements:1 2 3 4 5

Enter item to be searched in an array:2

Element 2 is found at index(es): 1

Process finished with exit code 0

WEEK – 12

PROBLEM

1# Write a program in C++ to demonstrate default constructor. Create a class having two data members in the private section. Define a default constructor to initialize these data members to initial value and display these values with the help of member function.

Program:

```
#include <iostream>

class myClass{

private:
    int num1, num2;

public:
    myClass() {
        num1 = 4;
        num2 = 2;
    }

    void displayValues() {
        std::cout << "Value of num1: " << num1 << std::endl;
        std::cout << "Value of num2: " << num2 << std::endl;
    }
};

int main() {

    myClass obj1;
    obj1.displayValues();
    return 0;
}
```

Console Output:

```
Value of num1: 4
Value of num2: 2

Process finished with exit code 0
```

2# Write a program in C++ to demonstrate parameterized/constructor overloading constructor. Create a class calculator that contains four data members in it. Initialize data members with different values using parameterized constructor and perform various arithmetic operation over these values and display result on to the computer screen.

Program:

```
#include<iostream>

class calculator{
private:
    double a, b, c, d;

public:
    calculator():a(0) , b(0), c(0),d(0){
        std::cout << "Default Constructor values: " << "a = " << a << " , b = "
        << b << " , c = " << c << " , d = " << d << std::endl;
        std::cout << std::endl;
    }

    calculator(double a, double b){
        std::cout << "Arithmetic Operations using two parameters" << std::endl;
        std::cout << "\nAddition: " << a + b << std::endl;
        std::cout << "Subtraction: " << a - b << std::endl;
        std::cout << "Multiplication: " << a*b << std::endl;
        if(b!=0)
            std::cout << "Division: " << a/b << std::endl;
        else
            std::cout << "Cannot be defined" << std::endl;
        std::cout << std::endl;
    }

    calculator(double a, double b, double c){
        std::cout << "Arithmetic Operations using three parameters" <<
std::endl;
        std::cout << "\nAddition: " << a + b + c << std::endl;
        std::cout << "Multiplication: " << a * b * c << std::endl;
        std::cout << std::endl;
    }

    calculator(double a, double b, double c, double d){
        std::cout << "Arithmetic Operations using three parameters" <<
std::endl;
        std::cout << "\nAddition: " << a + b + c + d << std::endl;
        std::cout << "Multiplication: " << a * b * c * d << std::endl;
    }
};

int main(){
    int a, b, c, d;

    std::cout << "Enter four numbers: ";
    std::cin >> a >> b >> c >> d;

    calculator calc1;
    calculator calc2(a,b);
    calculator calc3(a,b,c);
    calculator calc4(a,b,c,d);

    return 0;
}
```

Console Output:

```

Enter four numbers:8 5 -4 3

Default Constructor values: a = 0 , b = 0 , c = 0 , d = 0

Arithmetic Operations using two parameters

Addition: 13
Subtraction: 3
Multiplication: 40
Division: 1.6

Arithmetic Operations using three parameters

Addition: 9
Multiplication: -160

Arithmetic Operations using three parameters

Addition: 12
Multiplication: -480

Process finished with exit code 0

```

3# Create a class called Triangle that stores the length of the base and height of a right triangle in two private instance variables. Include a constructor that sets these values. Define two functions. The first is hypo(), which returns the length of the hypotenuse. The second is area (), which returns the area of the triangle.

Program:

```

#include <iostream>
#include<cmath>

class triangle{

private:
    double base, height;

public:
    triangle(double b, double h) : base(b) , height(h) {}

    double hypo(){
        return sqrt(pow(base,2)+pow(height,2));
    }

    double area(){
        return (0.5 * base * height);
    }
};

```

```

int main(){
    double base, height;

    std::cout << "Enter base of a right-angled triangle: ";
    std::cin >> base;
    std::cout << "Enter height of a right-angled triangle: ";
    std::cin >> height;

    triangle right_angled(base, height);
    std::cout << "Hypotenuse of a right angled triangle with base " << base << " and height " << height << " is: " << right_angled.hypo() << std::endl;
    std::cout << "Area of a triangle: " << right_angled.area() << std::endl;

    return 0;
}

```

Console Output:

Enter base of a right-angled triangle:5

Enter height of a right-angled triangle:12

Hypotenuse of a right angled triangle with base 5 and height 12 is: 13
Area of a triangle: 30

Process finished with exit code 0

4# Create a class for counting the number of objects created and destroyed within various block using constructor and destructors.

Program:

```

#include <iostream>

class ObjectCounter {

public:
    static int createdCount;
    static int destroyedCount;

    // Constructor
    ObjectCounter() {
        createdCount++;
        std::cout << "Object created. Total objects created: " << createdCount
        << std::endl;
    }

    // Destructor
    ~ObjectCounter() {
        destroyedCount++;
        std::cout << "Object destroyed. Total objects destroyed: " <<
destroyedCount << std::endl;
    }
};

// Initialize static members
int ObjectCounter::createdCount = 0;

```

```
int ObjectCounter::destroyedCount = 0;

int main() {
    std::cout << "Entering main block." << std::endl;
    ObjectCounter obj1;

    {
        std::cout << "Entering inner block." << std::endl;
        ObjectCounter obj2, obj3;
        std::cout << "Exiting inner block." << std::endl;
    }

    std::cout << "Exiting main block." << std::endl;
    return 0;
}
```

Console Output:

```
Entering main block.
Object created. Total objects created: 1

Entering inner block.
Object created. Total objects created: 2
Object created. Total objects created: 3

Exiting inner block.
Object destroyed. Total objects destroyed: 1
Object destroyed. Total objects destroyed: 2

Exiting main block.
Object destroyed. Total objects destroyed: 3

Process finished with exit code 0
```

5# Create an inter University Database with the following relations. Include at least four attributes for each table.

- University
 - Department
 - Program
 - Course
 - Syllabus
 - Faculty(Teacher)
- a) Create above tables and mention primary keys and foreign keys.
 - b) Also create secondary index for each table.
 - c) Insert at least 5 relevant records in each of the created tables.
 - d) Write following SQL queries based on above created database:
 - i) List of Universities situated at Delhi.
 - ii) List of all Departments of AMU.
 - iii) Find the location of JNU.
 - iv) List of all Programs run by University of Jammu.
 - v) List of Universities that run Program "MCA".
 - vi) List of Courses of "MCA" run by AMU.
 - vii) List of Faculties specialized in "Information Security" across different universities.
 - viii) Syllabus of "Computer Architecture" of different Universities.

Program:

```

CREATE TABLE UNIVERSITY(
    UNIVERSITY_ID    NUMBER(10) PRIMARY KEY NOT NULL,
    UNIVERSITY_NAME VARCHAR2(100)           NOT NULL,
    ADDRESS          VARCHAR2(100)          NOT NULL,
    STATE            VARCHAR2(100)          NOT NULL);

CREATE TABLE DEPARTMENT(
    DEPARTMENT_ID    NUMBER(10) PRIMARY KEY NOT NULL,
    DEPARTMENT_NAME VARCHAR2(100)           NOT NULL,
    TOTAL_MEMBERS    NUMBER(10) DEFAULT NULL,
    UNIVERSITY_ID    NUMBER(10)           NOT NULL,
    FOREIGN KEY (UNIVERSITY_ID) REFERENCES UNIVERSITY (UNIVERSITY_ID));

CREATE TABLE PROGRAM(
    PROGRAM_ID      NUMBER(10) PRIMARY KEY NOT NULL,
    PROGRAM_NAME    VARCHAR2(100)           NOT NULL,
    DURATION        NUMBER(10) DEFAULT NULL,
    TOTAL_SEATS     NUMBER(10)           NOT NULL,
    SEMESTERS       NUMBER(2)             NOT NULL,
    UNIVERSITY_ID   NUMBER(10)           NOT NULL,
    DEPARTMENT_ID   NUMBER(10)           NOT NULL,
    FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT (DEPARTMENT_ID),
    FOREIGN KEY (UNIVERSITY_ID) REFERENCES UNIVERSITY (UNIVERSITY_ID));
  
```

```

CREATE TABLE COURSE(
    COURSE_ID      NUMBER(10) PRIMARY KEY NOT NULL,
    COURSE_NAME    VARCHAR2(100)          NOT NULL,
    COURSE_CODE    VARCHAR2(100)          NOT NULL,
    COURSE_CREDITS NUMBER(2)             NOT NULL,
    PROGRAM_ID     NUMBER(2)              NOT NULL,
    TOTAL_LECTURES NUMBER(2)             NOT NULL,
    FOREIGN KEY (PROGRAM_ID) REFERENCES PROGRAM (PROGRAM_ID));

CREATE TABLE SYLLABUS(
    SYLLABUS_ID    NUMBER(10) PRIMARY KEY NOT NULL,
    SYLLABUS_NAME  VARCHAR2(100)          NOT NULL,
    TOTAL_UNITS    NUMBER(2)              NOT NULL,
    TOTAL_COURSES NUMBER(2)              NOT NULL,
    FOREIGN KEY (SYLLABUS_ID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE);

CREATE TABLE FACULTY(
    FACULTY_ID     NUMBER(10) PRIMARY KEY NOT NULL,
    FACULTY_NAME   VARCHAR2(200)          NOT NULL,
    TOTAL_STAFF    NUMBER(3)              NULL,
    ADDRESS        VARCHAR(200)           NOT NULL,
    FACULTY_SPECIALIZATION VARCHAR(200) NOT NULL,
    DEPARTMENT_ID  NUMBER(10)             NOT NULL,
    FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT (DEPARTMENT_ID));

CREATE TABLE TEACHER(
    TEACHER_ID     INT PRIMARY KEY NOT NULL,
    TEACHER_NAME   VARCHAR(100)          NOT NULL,
    FACULTY_ID    NUMBER(10)             NOT NULL,
    FOREIGN KEY (FACULTY_ID) REFERENCES FACULTY (FACULTY_ID) ON DELETE CASCADE);

--Create secondary index for each table.
CREATE INDEX UNIVERSITY_NAME ON UNIVERSITY (UNIVERSITY_NAME);
CREATE INDEX DEPARTMENT_NAME ON DEPARTMENT (DEPARTMENT_NAME);
CREATE INDEX PROGRAM_NAME ON PROGRAM (PROGRAM_NAME);
CREATE INDEX COURSE_NAME ON COURSE (COURSE_NAME);
CREATE INDEX SYLLABUS_NAME ON SYLLABUS (SYLLABUS_NAME);
CREATE INDEX FACULTY_NAME ON FACULTY (FACULTY_NAME);
CREATE INDEX TEACHER_NAME ON TEACHER (TEACHER_NAME);

--Insert at least 5 relevant records in each of the created tables.

INSERT INTO UNIVERSITY VALUES (001, 'Aligarh Muslim University', 'Aligarh', 'Uttar Pradesh');
INSERT INTO UNIVERSITY VALUES (002, 'Jamia Millia Islamia', 'Jamia Nagar, Okhla, New Delhi, 110025', 'Delhi');
INSERT INTO UNIVERSITY VALUES (003, 'Jawaharlal Nehru University', 'New Mehrauli Road, JNU Ring Rd, New Delhi, 110067', 'Delhi');
INSERT INTO UNIVERSITY VALUES (004, 'Delhi University', 'Benito Juarez Marg, South Campus, South Moti Bagh, New Delhi, 110021', 'Delhi');
INSERT INTO UNIVERSITY VALUES (005, 'IIT Madras', 'Chennai', 'Tamil Nadu');
INSERT INTO UNIVERSITY VALUES (006, 'University of Jammu', 'Baba Saheb Ambedkar Road, Jammu Tawi (J&K)-180006 (INDIA)', 'Delhi');
INSERT INTO UNIVERSITY VALUES (007, 'University of Hyderabad', 'Gachibowli, Hyderabad, Telangana 500046', 'Telangana');
INSERT INTO UNIVERSITY VALUES (008, 'IIT Bombay', 'Mumbai', 'Maharashtra');

```

```

INSERT INTO UNIVERSITY VALUES (009, 'IIT Delhi', 'New Delhi', 'Delhi');
INSERT INTO UNIVERSITY VALUES (010, 'IIT Kharagpur', 'Kharagpur', 'West Bengal');
INSERT INTO UNIVERSITY VALUES (011, 'IIT Kanpur', 'Kanpur', 'Uttar Pradesh');
INSERT INTO UNIVERSITY VALUES (012, 'IIT Roorkee', 'Roorkee', 'Uttarakhand');

INSERT INTO DEPARTMENT VALUES (001, 'Computer Science', 100, 001);
INSERT INTO DEPARTMENT VALUES (002, 'Electronics', 100, 001);
INSERT INTO DEPARTMENT VALUES (003, 'Mechanical', 100, 001);
INSERT INTO DEPARTMENT VALUES (004, 'Civil', 100, 001);
INSERT INTO DEPARTMENT VALUES (005, 'Electrical', 100, 007);
INSERT INTO DEPARTMENT VALUES (006, 'Chemical', 100, 001);
INSERT INTO DEPARTMENT VALUES (007, 'Mathematics', 100, 003);
INSERT INTO DEPARTMENT VALUES (008, 'Physics', 100, 001);
INSERT INTO DEPARTMENT VALUES (009, 'Chemistry', 100, 009);
INSERT INTO DEPARTMENT VALUES (010, 'Computer Science', 100, 004);

INSERT INTO PROGRAM VALUES (001, 'B.Tech', 4, 100, 8, 001, 001);
INSERT INTO PROGRAM VALUES (002, 'M.Tech', 4, 100, 8, 001, 002);
INSERT INTO PROGRAM VALUES (003, 'M.Tech', 4, 100, 8, 001, 003);
INSERT INTO PROGRAM VALUES (004, 'MCA', 4, 100, 8, 001, 001);
INSERT INTO PROGRAM VALUES (005, 'BCA', 4, 100, 8, 001, 005);
INSERT INTO PROGRAM VALUES (006, 'MBA', 4, 100, 8, 001, 006);
INSERT INTO PROGRAM VALUES (007, 'B.Tech', 4, 100, 8, 002, 001);
INSERT INTO PROGRAM VALUES (008, 'M.Tech', 4, 100, 8, 002, 002);
INSERT INTO PROGRAM VALUES (009, 'M.Tech', 4, 100, 8, 002, 003);
INSERT INTO PROGRAM VALUES (010, 'MCA', 4, 100, 8, 002, 004);
INSERT INTO PROGRAM VALUES (011, 'BCA', 4, 100, 8, 002, 005);
INSERT INTO PROGRAM VALUES (012, 'MBA', 4, 100, 8, 002, 006);
INSERT INTO PROGRAM VALUES (013, 'B.Tech', 4, 100, 8, 003, 001);
INSERT INTO PROGRAM VALUES (014, 'M.Tech', 4, 100, 8, 003, 002);
INSERT INTO PROGRAM VALUES (015, 'M.Tech', 4, 100, 8, 003, 003);
INSERT INTO PROGRAM VALUES (016, 'MCA', 4, 100, 8, 003, 004);
INSERT INTO PROGRAM VALUES (017, 'BCA', 4, 100, 8, 003, 005);
INSERT INTO PROGRAM VALUES (018, 'MBA', 4, 100, 8, 003, 006);
INSERT INTO PROGRAM VALUES (019, 'B.Tech', 4, 100, 8, 004, 001);
INSERT INTO PROGRAM VALUES (020, 'M.Tech', 4, 100, 8, 004, 002);
INSERT INTO PROGRAM VALUES (021, 'M.Tech', 4, 100, 8, 004, 003);
INSERT INTO PROGRAM VALUES (022, 'MCA', 4, 100, 8, 004, 004);
INSERT INTO PROGRAM VALUES (023, 'BCA', 4, 100, 8, 006, 005);
INSERT INTO PROGRAM VALUES (024, 'MBA', 4, 100, 8, 006, 006);

INSERT INTO COURSE VALUES (001, 'CS101', 'Introduction to Computer Science', 4, 001, 001);
INSERT INTO COURSE VALUES (002, 'CS102', 'Data Structures', 4, 001, 002);
INSERT INTO COURSE VALUES (003, 'CS103', 'Algorithms', 4, 003, 003);
INSERT INTO COURSE VALUES (004, 'CS104', 'Operating Systems', 4, 001, 004);
INSERT INTO COURSE VALUES (005, 'CS105', 'Computer Networks', 4, 004, 005);
INSERT INTO COURSE VALUES (006, 'CS106', 'Database Systems', 4, 001, 006);
INSERT INTO COURSE VALUES (007, 'CS107', 'Artificial Intelligence', 4, 001, 007);
INSERT INTO COURSE VALUES (008, 'CS108', 'Compiler Design', 4, 004, 008);
INSERT INTO COURSE VALUES (009, 'CS109', 'Computer Graphics', 4, 001, 009);
INSERT INTO COURSE VALUES (010, 'CS110', 'Software Engineering', 4, 001, 010);
INSERT INTO COURSE VALUES (011, 'CS111', 'Computer Architecture', 4, 001, 011);
INSERT INTO COURSE VALUES (012, 'CS112', 'Computer Organization', 4, 001, 012);
INSERT INTO SYLLABUS VALUES (001, 'Operating System', 4, 5);

```

```

INSERT INTO SYLLABUS VALUES (002, 'Computer Networks', 4, 5);
INSERT INTO SYLLABUS VALUES (003, 'Data Structures', 4, 5);
INSERT INTO SYLLABUS VALUES (004, 'Algorithms', 4, 5);
INSERT INTO SYLLABUS VALUES (005, 'Database Management System', 4, 5);
INSERT INTO SYLLABUS VALUES (006, 'Artificial Intelligence', 4, 5);
INSERT INTO SYLLABUS VALUES (007, 'Compiler Design', 4, 5);
INSERT INTO SYLLABUS VALUES (008, 'Computer Graphics', 4, 5);
INSERT INTO SYLLABUS VALUES (009, 'Software Engineering', 4, 5);
INSERT INTO SYLLABUS VALUES (010, 'Computer Architecture', 4, 5);
INSERT INTO SYLLABUS VALUES (011, 'Computer Organization', 4, 5);

INSERT INTO FACULTY VALUES (001, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 001);
INSERT INTO FACULTY VALUES (002, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 002);
INSERT INTO FACULTY VALUES (003, 'Faculty of Science', 100, 'Aligarh', 'Information Security', 003);
INSERT INTO FACULTY VALUES (004, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 004);
INSERT INTO FACULTY VALUES (005, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 005);
INSERT INTO FACULTY VALUES (006, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 006);
INSERT INTO FACULTY VALUES (007, 'Faculty of Science', 100, 'Aligarh', 'Information Security', 007);
INSERT INTO FACULTY VALUES (008, 'Faculty of Science', 100, 'Aligarh', 'Computer Science', 008);
INSERT INTO FACULTY VALUES (009, 'Faculty of Science', 100, 'Delhi', 'Computer Graphics', 010);

INSERT INTO TEACHER VALUES (001, 'Dr. A', 001);
INSERT INTO TEACHER VALUES (002, 'Dr. B', 002);
INSERT INTO TEACHER VALUES (003, 'Dr. C', 003);
INSERT INTO TEACHER VALUES (004, 'Dr. D', 004);
INSERT INTO TEACHER VALUES (005, 'Dr. E', 005);

--List of Universities situated at Delhi.
SELECT UNIVERSITY_NAME, STATE
FROM UNIVERSITY
WHERE STATE = 'Delhi';

-- ii) List of all the departments in Aligarh Muslim University.
SELECT DEPARTMENT_ID,
       DEPARTMENT_NAME,
       UNIVERSITY_NAME
  FROM DEPARTMENT,
       UNIVERSITY
 WHERE DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND UNIVERSITY.UNIVERSITY_NAME = 'Aligarh Muslim University';
-- iii) Find the location of JNU
SELECT UNIVERSITY_NAME, ADDRESS, STATE
  FROM UNIVERSITY
 WHERE UNIVERSITY_NAME = 'Jawaharlal Nehru University';
-- iv) List of all Programs run by University of Jammu.

```

```

SELECT UNIVERSITY_NAME, PROGRAM_NAME, DURATION
  FROM PROGRAM, UNIVERSITY
 WHERE PROGRAM.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND UNIVERSITY.UNIVERSITY_NAME = 'University of Jammu';

-- vi) List of Courses of "MCA" run by AMU.
SELECT UNIVERSITY_NAME, PROGRAM_NAME, COURSE_NAME, COURSE_CODE
  FROM COURSE,
       PROGRAM,
       UNIVERSITY
 WHERE COURSE.PROGRAM_ID = PROGRAM.PROGRAM_ID
   AND PROGRAM.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND UNIVERSITY.UNIVERSITY_NAME = 'Aligarh Muslim University'
   AND PROGRAM_NAME = 'MCA';

-- vii) List of Faculties specialized in "Information Security" across different
universities.
SELECT UNIVERSITY_NAME, FACULTY_NAME, FACULTY_SPECIALIZATION
  FROM FACULTY,
       UNIVERSITY,
       DEPARTMENT
 WHERE FACULTY.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
   AND DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND FACULTY.FACULTY_SPECIALIZATION = 'Information Security';

-- viii) Syllabus of "Computer Architecture" of different Universities.
SELECT UNIVERSITY_NAME, SYLLABUS_NAME, SEMESTERS, TOTAL_UNITS, TOTAL_COURSES
  FROM SYLLABUS,
       COURSE,
       DEPARTMENT,
       UNIVERSITY,
       PROGRAM
 WHERE SYLLABUS.SYLLABUS_ID = COURSE.COURSE_ID
   AND COURSE.PROGRAM_ID = PROGRAM.PROGRAM_ID
   AND PROGRAM.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
   AND DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND SYLLABUS_NAME = 'Computer Architecture';

-- ix) List of all faculties of Department of Computer Science of Delhi University.
SELECT FACULTY_NAME, FACULTY_SPECIALIZATION, DEPARTMENT_NAME, UNIVERSITY_NAME
  FROM FACULTY,
       DEPARTMENT,
       UNIVERSITY
 WHERE FACULTY.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
   AND DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   AND UNIVERSITY.UNIVERSITY_NAME = 'Delhi University'
   AND DEPARTMENT_NAME = 'Computer Science';

-- x) Find the University which has maximum Departments.
SELECT UNIVERSITY_NAME, COUNT(DEPARTMENT_NAME) AS DEPARTMENT_COUNT
  FROM DEPARTMENT,
       UNIVERSITY
 WHERE DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
 GROUP BY UNIVERSITY_NAME
 ORDER BY DEPARTMENT_COUNT DESC
  FETCH FIRST 1 ROWS ONLY;

```

Console Output:

```

mysql> -- Create secondary indexes
mysql> CREATE INDEX UNIVERSITY_NAME_IDX ON UNIVERSITY (UNIVERSITY_NAME);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX DEPARTMENT_NAME_IDX ON DEPARTMENT (DEPARTMENT_NAME);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX PROGRAM_NAME_IDX ON PROGRAM (PROGRAM_NAME);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX COURSE_NAME_IDX ON COURSE (COURSE_NAME);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX SYLLABUS_NAME_IDX ON SYLLABUS (SYLLABUS_NAME);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX FACULTY_NAME_IDX ON FACULTY (FACULTY_NAME);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX TEACHER_NAME_IDX ON TEACHER (TEACHER_NAME);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

```

mysql> SELECT * FROM UNIVERSITY;
+-----+-----+-----+-----+
| UNIVERSITY_ID | UNIVERSITY_NAME | ADDRESS | STATE |
+-----+-----+-----+-----+
| 1 | Aligarh Muslim University | Aligarh | Uttar Pradesh |
| 2 | Jamia Millia Islamia | Jamia Nagar, Okhla, New Delhi, 110025 | Delhi |
| 3 | Jawaharlal Nehru University | New Mehrauli Road, JNU Ring Rd, New Delhi, 110067 | Delhi |
| 4 | Delhi University | Benito Juarez Marg, South Campus, South Moti Bagh, New Delhi, 110021 | Delhi |
| 5 | IIT Madras | Chennai | Tamil Nadu |
| 6 | University of Jammu | Baba Saheb Ambedkar Road, Jammu Tawi (J&K)-180006 (INDIA) | Delhi |
| 7 | University of Hyderabad | Gachibowli, Hyderabad, Telangana 500046 | Telangana |
| 8 | IIT Bombay | Mumbai | Maharashtra |
| 9 | IIT Delhi | New Delhi | Delhi |
| 10 | IIT Kharagpur | Kharagpur | West Bengal |
| 11 | IIT Kanpur | Kanpur | Uttar Pradesh |
| 12 | IIT Roorkee | Roorkee | Uttarakhand |
+-----+-----+-----+-----+
12 rows in set (0.05 sec)

```

```

mysql> SELECT * FROM DEPARTMENT;
+-----+-----+-----+-----+
| DEPARTMENT_ID | DEPARTMENT_NAME | TOTAL_MEMBERS | UNIVERSITY_ID |
+-----+-----+-----+-----+
| 1 | Computer Science | 100 | 1 |
| 2 | Electronics | 100 | 1 |
| 3 | Mechanical | 100 | 1 |
| 4 | Civil | 100 | 1 |
| 5 | Electrical | 100 | 7 |
| 6 | Chemical | 100 | 1 |
| 7 | Mathematics | 100 | 3 |
| 8 | Physics | 100 | 1 |
| 9 | Chemistry | 100 | 9 |
| 10 | Computer Science | 100 | 4 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

```
mysql> SELECT * FROM PROGRAM;
+-----+-----+-----+-----+-----+-----+-----+
| PROGRAM_ID | PROGRAM_NAME | DURATION | TOTAL_SEATS | SEMESTERS | UNIVERSITY_ID | DEPARTMENT_ID |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | B.Tech | 4 | 100 | 8 | 1 | 1 |
| 2 | M.Tech | 4 | 100 | 8 | 1 | 2 |
| 3 | M.Tech | 4 | 100 | 8 | 1 | 3 |
| 4 | MCA | 4 | 100 | 8 | 1 | 1 |
| 5 | BCA | 4 | 100 | 8 | 1 | 5 |
| 6 | MBA | 4 | 100 | 8 | 1 | 6 |
| 7 | B.Tech | 4 | 100 | 8 | 2 | 1 |
| 8 | M.Tech | 4 | 100 | 8 | 2 | 2 |
| 9 | M.Tech | 4 | 100 | 8 | 2 | 3 |
| 10 | MCA | 4 | 100 | 8 | 2 | 4 |
| 11 | BCA | 4 | 100 | 8 | 2 | 5 |
| 12 | MBA | 4 | 100 | 8 | 2 | 6 |
| 13 | B.Tech | 4 | 100 | 8 | 3 | 1 |
| 14 | M.Tech | 4 | 100 | 8 | 3 | 2 |
| 15 | M.Tech | 4 | 100 | 8 | 3 | 3 |
| 16 | MCA | 4 | 100 | 8 | 3 | 4 |
| 17 | BCA | 4 | 100 | 8 | 3 | 5 |
| 18 | MBA | 4 | 100 | 8 | 3 | 6 |
| 19 | B.Tech | 4 | 100 | 8 | 4 | 1 |
| 20 | M.Tech | 4 | 100 | 8 | 4 | 2 |
| 21 | M.Tech | 4 | 100 | 8 | 4 | 3 |
| 22 | MCA | 4 | 100 | 8 | 4 | 4 |
| 23 | BCA | 4 | 100 | 8 | 6 | 5 |
| 24 | MBA | 4 | 100 | 8 | 6 | 6 |
+-----+-----+-----+-----+-----+-----+-----+
24 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM COURSE;
```

```
+-----+-----+-----+-----+-----+-----+
| COURSE_ID | COURSE_NAME | COURSE_CODE | COURSE_CREDITS | PROGRAM_ID | TOTAL_LECTURES |
+-----+-----+-----+-----+-----+-----+
| 1 | CS101 | Introduction to Computer Science | 4 | 1 | 1 |
| 2 | CS102 | Data Structures | 4 | 1 | 2 |
| 3 | CS103 | Algorithms | 4 | 3 | 3 |
| 4 | CS104 | Operating Systems | 4 | 1 | 4 |
| 5 | CS105 | Computer Networks | 4 | 4 | 5 |
| 6 | CS106 | Database Systems | 4 | 1 | 6 |
| 7 | CS107 | Artificial Intelligence | 4 | 1 | 7 |
| 8 | CS108 | Compiler Design | 4 | 4 | 8 |
| 9 | CS109 | Computer Graphics | 4 | 1 | 9 |
| 10 | CS110 | Software Engineering | 4 | 1 | 10 |
| 11 | CS111 | Computer Architecture | 4 | 1 | 11 |
| 12 | CS112 | Computer Organization | 4 | 1 | 12 |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM SYLLABUS;
```

```
+-----+-----+-----+-----+
| SYLLABUS_ID | SYLLABUS_NAME | TOTAL_UNITS | TOTAL_COURSES |
+-----+-----+-----+-----+
| 1 | Operating System | 4 | 5 |
| 2 | Computer Networks | 4 | 5 |
| 3 | Data Structures | 4 | 5 |
| 4 | Algorithms | 4 | 5 |
| 5 | Database Management System | 4 | 5 |
| 6 | Artificial Intelligence | 4 | 5 |
| 7 | Compiler Design | 4 | 5 |
| 8 | Computer Graphics | 4 | 5 |
| 9 | Software Engineering | 4 | 5 |
| 10 | Computer Architecture | 4 | 5 |
| 11 | Computer Organization | 4 | 5 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM FACULTY;
```

FACULTY_ID	FACULTY_NAME	TOTAL_STAFF	ADDRESS	FACULTY_SPECIALIZATION	DEPARTMENT_ID
1	Faculty of Science	100	Aligarh	Computer Science	1
2	Faculty of Science	100	Aligarh	Computer Science	2
3	Faculty of Science	100	Aligarh	Information Security	3
4	Faculty of Science	100	Aligarh	Computer Science	4
5	Faculty of Science	100	Aligarh	Computer Science	5
6	Faculty of Science	100	Aligarh	Computer Science	6
7	Faculty of Science	100	Aligarh	Information Security	7
8	Faculty of Science	100	Aligarh	Computer Science	8
9	Faculty of Science	100	Delhi	Computer Graphics	10

9 rows in set (0.00 sec)

```
mysql> SELECT * FROM TEACHER;
```

TEACHER_ID	TEACHER_NAME	FACULTY_ID
1	Dr. A	1
2	Dr. B	2
3	Dr. C	3
4	Dr. D	4
5	Dr. E	5

5 rows in set (0.00 sec)

mysql> -- i) List of Universities situated at Delhi.

```
mysql> SELECT UNIVERSITY_NAME, STATE
    → FROM UNIVERSITY
    → WHERE STATE = 'Delhi';
```

UNIVERSITY_NAME	STATE
Jamia Millia Islamia	Delhi
Jawaharlal Nehru University	Delhi
Delhi University	Delhi
University of Jammu	Delhi
IIT Delhi	Delhi

5 rows in set (0.21 sec)

mysql> -- ii) List of all the departments in Aligarh Muslim University.

```
mysql> SELECT DEPARTMENT_ID, DEPARTMENT_NAME, UNIVERSITY_NAME
    → FROM DEPARTMENT
    → JOIN UNIVERSITY ON DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
    → WHERE UNIVERSITY.UNIVERSITY_NAME = 'Aligarh Muslim University';
```

DEPARTMENT_ID	DEPARTMENT_NAME	UNIVERSITY_NAME
1	Computer Science	Aligarh Muslim University
2	Electronics	Aligarh Muslim University
3	Mechanical	Aligarh Muslim University
4	Civil	Aligarh Muslim University
6	Chemical	Aligarh Muslim University
8	Physics	Aligarh Muslim University

6 rows in set (0.02 sec)

```
mysql> -- iii) Find the location of JNU.
mysql> SELECT UNIVERSITY_NAME, ADDRESS, STATE
   → FROM UNIVERSITY
   → WHERE UNIVERSITY_NAME = 'Jawaharlal Nehru University';
+-----+-----+-----+
| UNIVERSITY_NAME | ADDRESS          | STATE |
+-----+-----+-----+
| Jawaharlal Nehru University | New Mehrauli Road, JNU Ring Rd, New Delhi, 110067 | Delhi |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> -- iv) List of all Programs run by University of Jammu.
mysql> SELECT UNIVERSITY_NAME, PROGRAM_NAME, DURATION
   → FROM PROGRAM
   → JOIN UNIVERSITY ON PROGRAM.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   → WHERE UNIVERSITY.UNIVERSITY_NAME = 'University of Jammu';
+-----+-----+-----+
| UNIVERSITY_NAME | PROGRAM_NAME | DURATION |
+-----+-----+-----+
| University of Jammu | BCA           |        4 |
| University of Jammu | MBA           |        4 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> -- vi) List of Courses of "MCA" run by AMU.
mysql> SELECT UNIVERSITY_NAME, PROGRAM_NAME, COURSE_NAME, COURSE_CODE
   → FROM COURSE
   → JOIN PROGRAM ON COURSE.PROGRAM_ID = PROGRAM.PROGRAM_ID
   → JOIN UNIVERSITY ON PROGRAM.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   → WHERE UNIVERSITY.UNIVERSITY_NAME = 'Aligarh Muslim University' AND PROGRAM_NAME = 'MCA';
+-----+-----+-----+-----+
| UNIVERSITY_NAME | PROGRAM_NAME | COURSE_NAME | COURSE_CODE |
+-----+-----+-----+-----+
| Aligarh Muslim University | MCA           | CS105       | Computer Networks |
| Aligarh Muslim University | MCA           | CS108       | Compiler Design  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> -- vii) List of Faculties specialized in "Information Security" across different universities.
mysql> SELECT UNIVERSITY_NAME, FACULTY_NAME, FACULTY_SPECIALIZATION
   → FROM FACULTY
   → JOIN DEPARTMENT ON FACULTY.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
   → JOIN UNIVERSITY ON DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
   → WHERE FACULTY.FACULTY_SPECIALIZATION = 'Information Security';
+-----+-----+-----+
| UNIVERSITY_NAME | FACULTY_NAME | FACULTY_SPECIALIZATION |
+-----+-----+-----+
| Aligarh Muslim University | Faculty of Science | Information Security |
| Jawaharlal Nehru University | Faculty of Science | Information Security |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```

mysql> -- viii) Syllabus of "Computer Architecture" of different Universities.
mysql> SELECT UNIVERSITY_NAME, SYLLABUS_NAME, SEMESTERS, TOTAL_UNITS, TOTAL_COURSES
    → FROM SYLLABUS
    → JOIN COURSE ON SYLLABUS.SYLLABUS_ID = COURSE.COURSE_ID
    → JOIN PROGRAM ON COURSE.PROGRAM_ID = PROGRAM.PROGRAM_ID
    → JOIN DEPARTMENT ON PROGRAM.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
    → JOIN UNIVERSITY ON DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
    → WHERE SYLLABUS_NAME = 'Computer Architecture';
+-----+-----+-----+-----+-----+
| UNIVERSITY_NAME | SYLLABUS_NAME | SEMESTERS | TOTAL_UNITS | TOTAL_COURSES |
+-----+-----+-----+-----+-----+
| Aligarh Muslim University | Computer Architecture | 8 | 4 | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

```

mysql> -- ix) List of all faculties of Department of Computer Science of Delhi University.
mysql> SELECT FACULTY_NAME, FACULTY_SPECIALIZATION, DEPARTMENT_NAME, UNIVERSITY_NAME
    → FROM FACULTY
    → JOIN DEPARTMENT ON FACULTY.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
    → JOIN UNIVERSITY ON DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
    → WHERE UNIVERSITY.UNIVERSITY_NAME = 'Delhi University' AND DEPARTMENT_NAME = 'Computer Science';
+-----+-----+-----+-----+
| FACULTY_NAME | FACULTY_SPECIALIZATION | DEPARTMENT_NAME | UNIVERSITY_NAME |
+-----+-----+-----+-----+
| Faculty of Science | Computer Graphics | Computer Science | Delhi University |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

```

mysql> -- x) Find the University which has the maximum number of Departments.
mysql> SELECT UNIVERSITY_NAME, COUNT(DEPARTMENT_NAME) AS DEPARTMENT_COUNT
    → FROM DEPARTMENT
    → JOIN UNIVERSITY ON DEPARTMENT.UNIVERSITY_ID = UNIVERSITY.UNIVERSITY_ID
    → GROUP BY UNIVERSITY_NAME
    → ORDER BY DEPARTMENT_COUNT DESC
    → LIMIT 1;
+-----+-----+
| UNIVERSITY_NAME | DEPARTMENT_COUNT |
+-----+-----+
| Aligarh Muslim University | 6 |
+-----+-----+
1 row in set (0.02 sec)

```

6# Write a C++ program to create a singly linked list and perform basic operations: insertion (at the beginning, end, and a given position) and deletion (at the beginning, end, and a given position).

Program:

```

#include <iostream>

class Node{
public:
    int key;
    int data;
    Node* next;

    Node() {
        key = 0;
        data = 0;
        next = NULL;
    }
}

```

```

Node(int k, int d) {
    key = k;
    data = d;
}
};

class SinglyLinkedList{

public:
    Node* head;

    SinglyLinkedList() {
        head = NULL;
    }

    SinglyLinkedList(Node *n) {
        head = n; //Pass by Address
    }

    //1. Checks if particular Node exists using key value.
    Node* nodeExists(int k){
        Node* temp = NULL;

        Node* ptr = head;
        while(ptr != NULL) {
            if(ptr->key == k) {
                temp = ptr;
            }
            ptr = ptr->next;
        }

        return temp;
    }

    //2. Append node to the list ---> append always happens at end

    void appendNode(Node *n) {
        if(nodeExists(n->key) !=NULL) {
            std::cout << "Node already exists with key value: " << n->key << ".\n";
            Append another node with different key value." << std::endl;
        }

        else{
            if(head == NULL) {
                head = n;
                std::cout << "Node Appended" << std::endl;
            }
            else{
                Node* ptr = head;
                while(ptr->next != NULL) {
                    ptr = ptr->next;
                }
                ptr->next = n;
                std::cout << "Node Appended." << std::endl;
            }
        }
    }

    //3. Prepend Node - Attach a node at the start

    void prependNode(Node* n) {
        if(nodeExists(n->key) !=NULL) {
            std::cout << "Node already exists with key value: " << n->key << ".\n";
            Append another node with different key value." << std::endl;
        }
    }
};

```

```

    }
else{
    n->next = head;
    head = n;
    std::cout << "Node Prepended. " << std::endl;
}
}

//4. Insert a Node after a particular node in the list

void insertNodeAfter(int k, Node* n){
    Node* ptr = nodeExists(k);
    if(ptr == NULL){
        std::cout << "No node exists with key value: " << k << std::endl;
    }
    else{
        if(nodeExists(n->key) !=NULL){
            std::cout << "Node already exists with key value: " << n->key <<
". Append another node with different key value." << std::endl;
        }
        else{
            n->next = ptr->next;
            ptr->next = n;
            std::cout << "Node Inserted" << std::endl;
        }
    }
}

//5. Delete node by unique key.
void deleteNodeByKey(int k){

    if(head == NULL){
        std::cout << "Singly Linked List already Empty. Cannot Delete" <<
std::endl;
    }

    else if(head!=NULL) {
        if(head->key==k) {
            head = head->next;
            std::cout << "Node Unlinked with key value: " << k << std::endl;
        }
        else{
            Node* temp = NULL;
            Node* prevptr = head;
            Node* currentptr = head->next;
            while(currentptr!=NULL) {
                if(currentptr->key == k) {
                    temp = currentptr;
                    currentptr = NULL;
                }

                else{
                    prevptr = prevptr->next;
                    currentptr = currentptr->next;
                }
            }

            if(temp!=NULL) {
                prevptr->next = temp->next;
                std::cout << "Node Unlinked with key value: " << k <<
std::endl;
            }
            else{
                std::cout << "Node doesn't exists with key value: " << k <<
            }
        }
    }
}

```

```

    std::endl;
}
}
}

//6. Update Node By key

void updateNodeByKey(int k, int d){
    Node* ptr = nodeExists(k);

    if(ptr!=NULL) {
        ptr->data = d;
        std::cout << "Node Data Updated Successfully" << std::endl;
    }
    else{
        std::cout << "Node doesn't exist with key value: " << k <<
std::endl;
    }
}

//7. printing linked list

void displayList(){
    if(head==NULL) {
        std::cout << "No nodes in Singly Linked List";
    }
    else{
        std::cout << std::endl << "Singly Linked List Values: ";
        Node* temp = head;

        while(temp!=NULL) {
            std::cout << "(" << temp->key << "," << temp->data << ") --> ";
            temp = temp->next;
        }
    }
}

};

int main(){

    SinglyLinkedList s;
    int option;
    int key1,k1,data1;

    do{
        std::cout << std::endl;
        std::cout<<"Operations that can be performed." << std::endl;
        std::cout << "1. Append Node" << std::endl;
        std::cout << "2. Prepend Node" << std::endl;
        std::cout << "3. Insert Node after a particular Node" << std::endl;
        std::cout << "4. Delete Node by key" << std::endl;
        std::cout << "5. Update Node by key" << std::endl;
        std::cout << "6. Display data" << std::endl;
        std::cout << "7. Clear Screen" << std::endl;
        std::cout << "Select option number for the operation you want to
perform. Enter 0 to exit: " ;
        std::cin>>option;
        Node* n1 = new Node();

        switch(option){
            case 0:
                break;

```

```

        case 1:
            std::cout << "Append Node Operation \nEnter Key and data of the
Node to be Appended: ";
            std::cin >> key1 >> data1;
            n1->key = key1;
            n1->data = data1;
            s.appendNode(n1);
            std::cout << std::endl;
            break;

        case 2:
            std::cout << "Prepend Node Operation \nEnter Key and data of the
Node to be Appended: ";
            std::cin >> key1 >> data1;
            n1->key = key1;
            n1->data = data1;
            s.prependNode(n1);
            std::cout << std::endl;
            break;

        case 3:
            std::cout << "Insert Node after Operation \nEnter Key of
existing Node after which you want to insert this New Node: ";
            std::cin >> k1;
            std::cout << "Enter key and data of the new Node: ";
            std::cin >> key1 >> data1;
            n1->key = key1;
            n1->data = data1;
            s.insertNodeAfter(k1,n1);
            std::cout << std::endl;
            break;

        case 4:
            std::cout << "Delete Node By Key Operation \nEnter Key of the
node to be deleted: ";
            std::cin >> k1;
            s.deleteNodeByKey(k1);
            std::cout << std::endl;
            break;

        case 5:
            std::cout << "Update Node By Key Operation \nEnter Key and New
data to be updated: ";
            std::cin >> key1 >> data1;
            s.updateNodeByKey(key1, data1);
            std::cout << std::endl;
            break;

        case 6:
            s.displayList();
            std::cout << std::endl;
            break;

        case 7:
            system("cls");
            break;

        default:
            std::cout << "Wrong Input...!" << std::endl;
    }
} while(option!=0);

return 0;
}

```

Console Output:

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:**1**

Append Node Operation

Enter Key and data of the Node to be Appended:**1 10**

Node Appended

Singly Linked List Values: (1,10) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:**2**

Prepend Node Operation

Enter Key and data of the Node to be Prepended:**2 20**

Node Prepended.

Singly Linked List Values: (2,20) --> (1,10) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:**1**

Append Node Operation

Enter Key and data of the Node to be Appended:**3 30**

Node Appended.

Singly Linked List Values: (2,20) --> (1,10) --> (3,30) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:3

Insert Node after Operation

Enter Key of existing Node after which you want to insert this New Node:1

Enter key and data of the new Node:4 40

Node Inserted

Singly Linked List Values: (2,20) --> (1,10) --> (4,40) --> (3,30) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:4

Delete Node By Key Operation

Enter Key of the node to be deleted:2

Node Unlinked with key value: 2

Singly Linked List Values: (1,10) --> (4,40) --> (3,30) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:4

Delete Node By Key Operation

Enter Key of the node to be deleted:3

Node Unlinked with key value: 3

Singly Linked List Values: (1,10) --> (4,40) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:1

Append Node Operation

Enter Key and data of the Node to be Appended:2 20

Node Appended.

Singly Linked List Values: (1,10) --> (4,40) --> (2,20) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:4

Delete Node By Key Operation

Enter Key of the node to be deleted:4

Node Unlinked with key value: 4

Singly Linked List Values: (1,10) --> (2,20) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:5

Singly Linked List Values: (1,10) --> (2,20) -->

Operations that can be performed.

1. Append Node
2. Prepend Node
3. Insert Node after a particular Node
4. Delete Node by key
5. Display data

Select option number for the operation you want to perform. Enter 0 to exit:0

Process finished with exit code 0