CHAPTER-12
STRUCTURED DATA TYPE: ARRAY
TYPE A : VERY SHORT ANSWER QUESTIONS

1.	**What do you understand by an array? What is the significance of array?**
Ans.	An array is a collection of variables of the same type that are referenced by a common name. Arrays are very useful in a case where quite many elements of the same types to be stored and processed.
2.	**What is meant by index of an element? How are indices numbered in C++?**
Ans.	The element's number is referred to as an index. An array index in C++ starts with number 0 not 1. That is, array A[5] will be having elements: A[0], A[1], A[2], A[3], A[4].
3.	**Determine the total bytes required to store B[17], a char array.**
Ans.	As B[17] is a character array, the size of each element is 1 byte. The total bytes required to store B[17] is: 1 x 17 = 17
4.	**What is the data type of element of an array called?**
Ans.	The data type of array elements is known as the base type of the array.
5.	**How is one-dimensional array represented in memory?**
Ans.	The elements of an one-dimensional array are stored in contiguous memory location in their index order. For example, an array *grade* of type *char* with 8 elements declared as `char grade[8];` will have the element grade[0] at the first allocated memory location, grade[1] at the next contiguous memory location, grade[2] at the next, and so forth.
6.	**Determine the number of bytes required to store an int array A[23], if the int size is 4 bytes.**
Ans.	The size of int is 4 bytes and there are 23 elements in array. So the number of bytes required to store an int array is: 4 x 23 = 92.
7.	**An array element is accessed using** **(a) a first-in-first-out approach (b) the dot operator** **(c) an element name (d) an index number**
Ans.	(d) an index number
8.	**Write a statement that defines a one-dimensional array called amount of type double that holds two elements.**
Ans.	`double amount[2];`
9.	**The elements of a twenty-element array are numbered from _____ to _____.**
Ans.	`0 to 19.`
10.	**Write a statement that takes element k of array amount and writes it to cout<< with the insertion operator.**
Ans.	`cout<<amount[k];`
11.	**Element amount[9] is which element of the array?** **(a) the eighth (b) the ninth (c) the tenth (d) impossible to tell.**
Ans.	(c) the tenth
12.	**Write a statement that defines a char array to store coins' types and initializes it to the values 10-paise, 25-paise, 50-paise, 1-rupee, 2-rupee, 5-rupee, 100-rupee.**
Ans.	`char coins[7][15] = { "10-paise", "25-paise", "50-paise", "1-rupee", "2-rupee", "5-rupee", "100-rupee" };`
13.	**How does C++ view a string as? Which character marks the end of a string?**
Ans.	C++ views a string as single-dimensional character array. The null character '\0' marks the end of a string.
14.	**Write a statement that defines a string variable school to hold a string of up to 25 characters.**
Ans.	`char school[26];`
15.	**Write a statement that defines a string constant, called dextrose, that has the value "C6H1206-H20".**
Ans.	`char dextrose[20] = { "C6H1206-H20 " }`
16.	**Declare following arrays:** **(i) figures of 30 char (ii) check of 100 short (iii) balance of 26 float (iv) budget of 58 double**

Ans.	`(i) char figures[30];`
	`(ii) short check[100];`
	`(ii) float balance[26];`

	`(iv) double budget[58];`
17.	**Declare an array of 5 ints and initialize it to the first five even numbers.**
Ans.	`int A[5] = { 2, 4, 6, 8, 10 };`
18.	**Write a statement that assign the sum of the first, second and last elements of the array n question 17 to the variable threesum.**
Ans.	`threesum=(A[0] + A[2] + A[4]);`
19.	**Write a statement that displays the value of the second element in the long array balance.**
Ans.	`cout<<A[1];`
20.	**Declare an array of char and initialize it to the string "Pizzas".**
Ans.	`char string[10] = "Pizzas";`
21.	**For a multidimensional array X[5][24], find the number of bytes required.**
Ans.	No. rows = 5, No. of columns = 24, Size of each elements = 2 bytes. Total bytes = number-of-rows x number-of-columns x size of (base type) = 5 x 24 x 2 =240 bytes.
22.	**For the same array mentioned in question 21, find the total number of elements in X.**
Ans.	Total elements = number-of-rows x number-of-columns = 5 x 24 = 120
23.	**For a multidimensional array B[9][15] find the total number of elements in B.**
Ans.	No. rows = 9, No. of columns = 15, Size of each elements = 2 bytes. Total bytes = number-of-rows x number-of-columns = 9 x 15 =135
24.	**How are the 2-D arrays stored in the memory?**
Ans.	Two-dimensional arrays are stored in a row-column matrix, where the first index indicated the row and the second indicates the columns.
25.	**Given two 2-D arrays, what is the condition to calculate the sum of the given arrays?**
Ans.	To calculate the sum of the given 2-D arrays, both arrays should be identical. For instance, A[n][m] and B[p][q] are the two given matrices, then, in order to perform sum operation, following condition must be true: n = p and m = q
26.	**Given two 2-D arrays, what condition must be satisfied to calculate the difference of the two arrays?**
Ans.	To calculate the sum of the given 2-D arrays, both arrays should be identical. For instance, A[n][m] and B[p][q] are the two given matrices, then, in order to perform difference operation, following condition must be true: n = p and m = q
27.	**To obtain the product of the two 2-D arrays, what condition must be true?**
Ans.	Suppose, A[n][m] and B[p][q] are the two given matrices, then, in order to perform obtain the product, following condition must be true: m= p
28.	**Given the following 2-D arrays:** **A= 1 0 0 / 0 1 0 / 0 0 1** **B= 0 1 1 / 1 0 1 / 1 1 0** **What will be the sum of the A and B?**
Ans.	A + B = 1 1 1 / 1 1 1 / 1 1 1
29.	**Given the following 2-D arrays:** **A= 1 0 / 1 1 / 1 1** **B= 1 1 1 / 2 2 2 / 3 3 3** **What will be the sum of the A and B?**

Ans.	The sum of the A and b is not possible as A and B are not identical.
30.	**Given the following 2-D arrays:** **X= 2 1 2 / 3 1 3 / 4 1 4** **y= 1 2 3 / 4 5 6** **Is the product of X and Y possible?**
Ans.	Yes, the product of X and Y is possible.
31.	**Given the following array:** **A= 1 2 3 / 4 5 6 / 7 8 9** **Write its diagonal elements.**
Ans.	`Left diagonal = 1 5 9` `Right diagonal = 3 5 7`
32.	**For the same array A, mentioned in question 31, calculate the sum of diagonal elements of a.**
Ans.	The sum of the diagonal elements is (1 + 5 + 9 + 3 + 5 + 7) = 30.
33.	**What is a vector? Can a 2-D array be called as a vector?**
Ans.	A vector is a mathematical term that refers to collection of analogous numbers. A 2-D array with one row can be called as a vector.
34.	**Can any dimension of an array be skipped at its declaration time?**
Ans.	Only the first dimension can be skipped at its declaration time, the second dimension must be given.
35.	**How does C++ calculate the size of unsized array?**
Ans.	To skip the size of array, we must give list of initializers and C++ calculates the size of the unsized array by counting them.
36.	**Define a two dimensional array MAT of (4 x 5) of type integer. Also initialize all the elements of MAT with the value 0(zero). Construct one statement to assign the value 100 to cell in row 3 and column 4.**

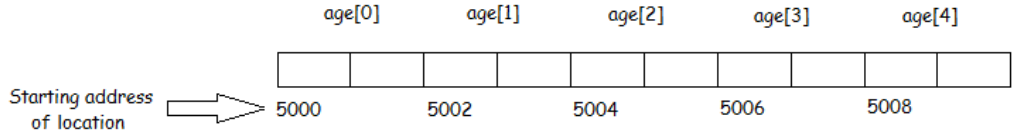
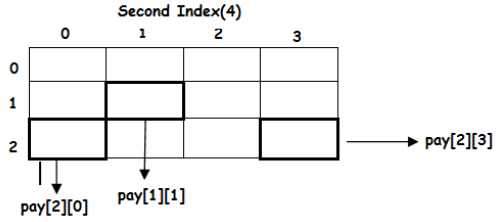
Ans.	`int MAT[4][5] = { 0, 0, 0, 0, 0`
	`0, 0, 0, 0, 0`
	`0, 0, 0, 0, 0`
	`0, 0, 0, 0, 0 };`
	`MAT[3][4] = 100;`

TYPE B : SHORT ANSWER QUESTIONS

1.	**What is an array? What is the need for arrays?**
Ans.	An array is a collection of variables of the same type that are referenced by a common name. Arrays are very useful in a case where quite many elements of the same types to be stored and processed. For example, to store the marks of 50 students we required 50 variables. But if we use array than we can store all 50 student's mark in one array that have 50 elements. Elements of these arrays will be referred to as arrayname[n], where n is the element number in the array. Writing such a program would not only be simple but also very easy to code and understand.
2.	**What are different types of arrays? Give example of each array type.**

Ans.	There are three different types of arrays as following:
	1. One-dimensional array: The array that has only one subscript is known as a one-dimensional array. For example,
	`int marks[50];`
	The above statement declare array marks has 50 elements, mark[0] to mark[49].
	2. Two-dimensional array: The array that has two subscripts is known as a two-dimensional array. For example,
	`int sales[5][12];`
	Above statement declare an array sales that has 5 rows and 12 columns.
	3. Multi-dimensional array: The array that has more than one dimension is known as multi-dimensional array.
	`int A[3][3][3];`
	Above statement declare an A array that have three dimensions.

3.	**Write a note on how single-dimension array are represented in memory.**
Ans.	Suppose an array age of type char with 5 elements declared as `int age[5];`

	Its each element will have sizeof(int) bytes for its storage. On a system with 2 bytes integer size, the array age's element will be having 2 bytes for its storage. If the starting memory location of array age is 5000, then it will be represented in the memory as shown below: 
4.	**How does the amount of storage (in bytes) depend upon type and size of an array? Explain with the help of an example.**
Ans.	The amount of storage required to hold an array is directly related to its type and size. For a single-dimensional array, the total size (in bytes) can be computed according to following formula: `total bytes = sizeof(type) * size_of_array` For example, of we have an array age of type int that has 5 elements then the amount of storage required is 2 * 5 = 10.
5.	**How are strings manipulated in C++? Support your answer with examples.**
Ans.	In C++ strings are manipulated as a single-dimensional character array. A string is defined as a character array that is terminated by a null character '\0'. For this reason, the character arrays are declared one character longer than the largest string they can hold. For instance, to declare an array str that holds a 10-chracter string, we would write `char str[11];` this makes room for the null at the end of the string.
6.	**What do you understand by two-dimensional arrays? State some situations that can be easily represented by two-dimensional arrays.**
Ans.	A two-dimensional array is an array in which each element is itself an array. For instance, an array A[m][n] is an M by N table with M rows and N columns containing M x N elements. Situations that can be represented by two-dimensional arrays: ✓ A two-dimensional array can be used to store objects, which is especially convenient for programming sketches that involve some sort of "grid" or "board." ✓ We might write a program using a two-dimensional array to draw a grayscale image.
7.	**How are two-dimensional arrays represented in memory?**
Ans.	Two-dimensional arrays are stored in row-column matrix, where the first index indicates the row and the second indicates the column. Suppose we have declared an array pay as follows: `float pay[3][4];` it will be having 3 x 4 = 20 elements which will be represented in memory as shown below: 
8.	**Can you think of a difference between an array of strings and other two-dimensional arrays? What is it? Support your answer with examples.**

Ans.

Array of strings	Two-dimensional array
An array of string is a two-dimensional character array.	A two-dimensional array is an array in which each element is itself an array.
Example: `char strings[10][51];` The above can contain 10 strings, each of 51 characters long.	Example: `int Arr[2][3];` The above has 2 rows and 3 columns.
The first index refers the number of strings and the second index refers to the maximum length of each string.	The first index refers the number of rows and the second index refers to the number of columns in the array.
An array of string is terminated with null character '\0'.	Two-dimensional array is not terminated with null

	character.

9. **Consider a 1-D array 'A' containing 100 integers. Develop a program to do the following:**
(i) remove all occurrences of a given integer.
(ii) shift the elements of the array to the right so that used space is available at the left end.
(ii) fill the unused spaces by 0 (zero).
(For example, the array 10 | 20 | 15 | 4 | 20 | 2 | 20 after execution of the program for given integer 20 should become

0	0	0	10	15	4	2

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      int A[100],no,val,found;
      clrscr();
      cout<<"Enter number of elements you want to insert ";
      cin>>no;
      for(int i=0;i<no;i++)
      {
            cout<<"Enter element "<<i+1<<":";
            cin>>A[i];
      }

      cout<<"Enter the number you want to search ";
      cin>>val;

      for(int j=0; j<no; j++)
      {
            if(A[j]==val)
                  A[j]=0;
      }

      for(int k=0; k<no; k++)
      {
            cout<<A[k]<<"   ";
      }

      int divider = -1;
      for (int m = 0; m < no; m++)
      {
            if (A[m] == 0)
            {
                  // swap A[i] and A[divider+1].
                  int temp = A[m];
                  A[m] = A[divider + 1];
                  A[divider + 1] = temp;
                  //increment divider.
                  divider ++;
            }
      }
getch();
}
```

10. **Suppose A, B, C are arrays of integers of sizes m, n, m + n respectively. Give a program to produce a third array C, containing all the data of array A and B.**

Ans.

```
#include<iostream.h>
#include<conio.h>
```

```
void main()
{
      int A[100], B[100], C[200],n,m,k=0;
      int I=0, J=0;
      cout<<"\nEnter number of elements you want to insert in first array ";
      cin>>n;
      cout<<"Enter element in first array\n";
      for(int i=0;i<n;i++)
      {
            cout<<"Enter element "<<i+1<<":";
            cin>>A[i];
      }
      cout<<"\nEnter number of elements you want to insert in second array ";
      cin>>m;
      cout<<"Enter element in second array\n";
      for(i=0;i<m;i++)
      {
            cout<<"Enter element "<<i+1<<":";
            cin>>B[i];
      }
       for (int T=I;T<n;T++)
           C[k++]=A[T];
       for (T=J;T<m;T++)
           C[k++]=B[T];
      cout<<"The Merged Array is:"<<endl;
      for(i=0;i<k;i++)
      {
            cout<<C[i]<<" ";
      }
      getch();
      return 0;
}
```

11. **Write a short program that adds two matrices X[3][3] and Y[3][3].**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
     clrscr();
      int i,j,n,a[3][3],b[3][3],c[3][3];
      cout<<"enter value of elements of matrix A"<<"\n";
      for(i=0;i<3;i++)
      {
            for(j=0;j<3;j++)
            {
                  cin>>a[i][j];
            }
      }
      cout<<"enter values of elements of matrix B"<<"\n";
      for(i=0;i<3;i++)
      {
            for(j=0;j<3;j++)
            {
                  cin>>b[i][j];
            }
      }
      cout<<"addition of matrix A and B is"<<"\n";
```

```
      for(i=0;i<3;i++)
      {
            cout<<"\n";
            for(j=0;j<3;j++)
            {
                  c[i][j]=a[i][j]+b[i][j];
                  cout<<c[i][j]<<"\t";
            }
            cout<<"\n";
      }
      getch();
}
```

12. **Write a short program that subtracts two matrices X[4][4] and Y[4][4].**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
     clrscr();
      int i,j,n,a[3][3],b[3][3],c[3][3];
      cout<<"enter value of elements of matrix A"<<"\n";
      for(i=0;i<3;i++)
      {
            for(j=0;j<3;j++)
            {
                  cin>>a[i][j];
            }
      }
      cout<<"enter values of elements of matrix B"<<"\n";
      for(i=0;i<3;i++)
      {
            for(j=0;j<3;j++)
            {
                  cin>>b[i][j];
            }
      }
      cout<<"subtraction of matrix A and B is"<<"\n";
      for(i=0;i<3;i++)
      {
            cout<<"\n";
            for(j=0;j<3;j++)
            {
                  c[i][j]=a[i][j]-b[i][j];
                  cout<<c[i][j]<<"\t";
            }
            cout<<"\n";
      }
      getch();
}
```

13. **Write a short program that doubles every element of a matrix A[4][4].**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      int i,j,n,a[4][4];
      cout<<"enter value of elements of matrix A"<<"\n";
```

```
      for(i=0;i<4;i++)
      {
            for(j=0;j<4;j++)
            {
                  cin>>a[i][j];
            }
      }
      cout<<"addition of matrix A and B is"<<"\n";
      for(i=0;i<4;i++)
      {
            cout<<"\n";
            for(j=0;j<4;j++)
            {
                  a[i][j]*=a[i][j];
                  cout<<a[i][j]<<"\t";
            }
            cout<<"\n";
      }
      getch();
}
```

14. **Write an algorithm to print all those elements of a matrix X[4][4] that are not diagonal elements.**

Ans. Students I am giving you the program for printing Non Diagonal elements of a matrix X[4x4], try to convert this code into algorithm.

```
#include<conio.h>
#include<iostream.h>
    void accept(int a[4][4],int size)
    {
       cout<<"Diagonal One:";
       for (int i=0;i<size;i++)
           for(int j=0;j<size;j++)
     if (i != j && i != size-j-1)
          cout<<a[i][j];
}
void main()
{
      int a[4][4]={{5,4,3,4},{6,7,9,1},{8,0,3,7},{2,4,5,9}};
      clrscr();
      accept(a,4);
      getch();
}
```

15. **Let A(n x n) that are not diagonal array. Write a program to find the sum of all the elements which lie on either diagonal. For example, for the matrix shown below, your program should output 68 = (1 + 6 + 11 + 16 + 4 + 7 + 10 + 13):**

1	**2**	**3**	**4**
5	**6**	**7**	**8**
9	**10**	**11**	**12**
13	**14**	**15**	**16**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{     clrscr();
      int i,j,n,m,a[10][10];
      int sum=0;
      cout<<"Enter m:";
      cin>>m;
```

```
      cout<<"Enter n:";
      cin>>n;
      cout<<"enter value of elements of matrix A"<<"\n";
      for(i=0;i<m;i++)
      {
            for(j=0;j<n;j++)
            {
                  cin>>a[i][j];
            }
      }
      for(i=0;i<m;i++)
            {
                  for(j=0;j<n;j++)
                  {
                        if(i==j)              //elements on first diagonal
                        sum+= a[i][j];
                        if((i+j)==(m-1))      // elements on off-diagonal
                        sum+= a[i][j];
                  }
            }
      cout<<"\t"<<sum;
      getch();
}
```

16. **Write a program to count number of spaces in a string.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main( )
{
      char str[50];
      int c=0;
      clrscr();
      cout<<"Enter string: ";
      gets(str);
      for(int i=0;i<strlen(str);i++)
      {
            if(str[i]==' ')
                  c++;
      }
      cout<<"the no. of spaces: "<<c;
      getch( );
}
```

17. **Write a program to count number of occurrences of a given character in a string.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main( )
{
      char str[50],ch;
      int c=0;
      clrscr();
      cout<<"Enter string: ";
      cin>>str;
      cout<<"Enter character: ";
      cin>>ch;
```

```
      for(int i=0;i<strlen(str);i++)
      {
            if(str[i]==ch)
                  c++;
      }
      cout<<"the no. of occurences is: "<<c;
      getch( );
}
```

18. **Write a program to compare two strings for equality.**

Ans.

```
int compare(char str1[], char str2[])
{
      if ( strcmp (str1,str2) == 0)
            return 0;
      else
            return -1;
}
void main()
{
      char str1[20];
      char str2[20];
      int res;
      cout<<" first string:\n1: ";
      cin>>str1 ;
      cout<<" second string:\n 2: ";
      cin>>str2;
      res=compare(str1,str2);
      if(res==0)
            cout<<"Equal strings";
      else
            cout<<"Not equal";
      getch();
}
```

TYPE C : LONG ANSWER QUESTIONS

1. **From a two-dimensional array A[4][4], write a program to prepare a one-dimensional array B[16] that will have all the elements of A if they are stored in row-major form. For example, for the following array**

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

The resultant array should be 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Ans.

```
void main( )
{
int B[16];
int A[4][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};

clrscr();
//Print the array
cout<<"\n The Original matrix : \n\n";
for(int i = 0; i < 4 ; i++)
{
 for(int j = 0; j < 4 ; j++)
    cout<< A[i][j]<<"\t";
 cout<< "\n";
}
```

```
int n=0;
// Convert 2 - D array into 1-D array by row-major rule
for(int p = 0; p < 4 ; p++)
{
   for(int q = 0; q < 4 ; q++)
   {
      B[n] = A[p][q];
      cout<< B[n] << "  ";
      n++;
   }
}
//print the converted 1D array
cout<<"\n \n";
for(int k=0; k<16; k++)
   cout<< B[k] << "  ";

getch( );
}
```

2. **Write a program to count the number of words in a string as well as the number of characters other than a space.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
      char sen[50];
      int word_count = 0, chr = 0, space = 0;
      cout<< "\n Enter A String :";
      gets(sen);
      for(int i = 0; sen[i] != '\0' ; i++)
      {   if (sen[i] == ' ')
          space++; }
       chr = strlen(sen) - space;
        word_count = space + 1;
        cout<< "\n The Total no. of words = " << word_count;
        cout << "\n The total characters without spaces = " << chr;
        getch();
}
```

3. **Write a program to print the upper and lower triangle of a matrix.**

Ans.

```
#include<iostream.h>
#include<conio.h>
# define n 10
void main( )
{
      int mat[n][n];
      int d;
      // Input elements
      cout<<  "\n  Enter dimension of square matrix:";
      cin >> d;
      for(int i = 0; i < d ; i++)
      for( int j = 0; j < d ; j++)
      {cout<<"\n Enter elements for "<<  i+1 << "," << j+1 <"location :";
       cin >> mat[i][j];
      }
      clrscr();
      //Print the array
      cout<<"\n The Original matrix : \n\n";
```

```
      for( i = 0; i < d ; i++)
      {     for( j = 0; j < d ; j++)
               cout<<  mat[i][j]<<"\t";
            cout<< "\n";
      }
      //upper half of left diagonal ----
      cout<<"\n The Upper half of the matrix : \n\n";
      for( i = 0; i < d ; i++)
      {
            for( j = 0; j < d ; j++)
                  {   if(i < j)
                        cout <<  mat [i][j] << " " ;
                      else
                        cout << " " << " ";
                  }
            cout << "\n ";
      }
      //lower half of left diagonal -----
      cout<<"\n The Lower half of the matrix : \n\n";
      for( i = 0; i < d ; i++)
      {
            for( j = 0; j < d ; j++)
                  {   if(i >  j)
                        cout <<  mat [i][j] << " " ;
                      else
                        cout << " " << " ";
                  }
            cout << "\n ";
      }
      getch ( );
}
```

4. **Write a program to reverse a string (i) using another array, (ii) without another array.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{    char str[50], rev[50];
     cout<< "\n Enter a string :";
     gets(str);
     int l = strlen(str);
     /* Reverse String using another array */
     for( int i = l-1, k = 0 ; i>= 0 ; i--, k++)
          rev[k] = str[i];
          rev[k] = '\0';

     cout<< "\n Original String is "<< str;
     cout<< "\n Reverse String is "<< rev ;
     cout<< "\n Reverse String is ";
     /* Reverse String without using another array */
     for(  i = l-1 ; i>=0 ; i--)
      cout<< str[i];
      getch();
}
```

5. **Write a function that checks whether or not two arrays (of characters) are identical, that is, whether they have same characters and all characters in corresponding position are equal.**

Ans.

```
#include<iostream.h>
```

```
#include<conio.h>
#include<string.h>
void main()
{
      char sen1[50], sen2[50];
      void chk_eq(char str1[], char str2[]);
      cout<< "\n Enter String1 :";
      gets(sen1);

      cout<<"\n Enter String 2:";
      gets(sen2);
      chk_eq(sen1, sen2);
      getch();
}
void chk_eq(char str1[],char str2[])
{
      int l1,l2;
      l1 = strlen(str1);
      l2 = strlen(str2);
      if(l1 == l2)
      {
            cout<<" \ n Two String are equal ";
            getch();
            exit(1);
      }
      for( int i = 0; i<= l1 - 1 ; i++)
      {
        if(str1[i]!=str2[i])
         { cout<<" \ n Two String are NOT equal ";
           getch();
           exit(1);
         }
      }
      cout<< "\n Two Strings are EQUAL lengthwise , character wise and case
      wise ";
      getch();
}
```

6. **Write a program that reads an array of strings and counts the number of strings of each length that occurs, as well as the total number of strings. The program should then print the mean (average) entirely of (uppercase and lowercase) letters and is terminated by '.'.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
      clrscr();
      char str[255] = " Hi \n Enter Strings \n Thanks ";
      cout<< "\n Enter a String (~ to terminate ): ";
      cin.getline(str, 255, '~');
      /* Count no. of strings */
      int no_f_strings = 0;
      for(int i = 0; str[i] != '\0' ; i++)
         if(str[i] == '\n')
            no_f_strings ++;
      /*Count total number of strings */
      int tot_strings = 0;
```

```
      for( i = 0 ; str[i] != '\0' ; i++)
      {
      /*space or new-line or fullstop means new word starts */
      if( str[i] == ' ' || str [i] == '\n' || str[i] == '.')
        {tot_strings++;
        while(str[i] == ' ')
         i++;}
       if(str[i] == '\0')
         i--;
      }
      tot_strings++; //for the last word in the string array
      cout<< "\n The  number of strings of each length : " << no_f_strings;
      cout<<"\n The total number of strings (words) : " << tot_strings;
      int len = strlen(str);
      int ucas=0, lcas = 0, fullst = 0;
      for( i=0; str[i] != '\0' ; i++)
      {
           if(str[i]>='A' && str[i]<='Z')
             ucas++;
           if(str[i]>='a' && str[i]<='z')
            lcas++;
           if(str[i]=='.')
           fullst++;
      }
      float avg_ucas = (float) (ucas/len);
      float avg_lcas = (float) (lcas/len);
      cout<< "\n The Mean  of lowercase letters in strings of array  = " <<
      avg_lcas;
      cout<< "\n The Mean  of uppercase letters in strings of array = " <<
      avg_ucas;
      cout<< "\n The total number of  terminating fullstop= " << fullst;
      getch();
}
```

7. **Write a program to obtain the product of the following matrices:**

$$A = \begin{matrix} 5 & 9 & -2 \\ 8 & 0 & 5 \\ 0 & 2 & 8 \end{matrix} \qquad B = \begin{matrix} 2 & 4 \\ 0 & 1 \\ 1 & 3 \end{matrix}$$

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{     clrscr();
      int A[3][3] = {{5,9,-2},
                     {8,0,5},
                     {0,2,8}};
      int B[3][2] = {{2,4},
                     {0,1},
                     {1,3}};
      int P[3][2] , i, j, k;
      cout<< "\n Matrix A : \n";
for( i =0; i< 3 ; i++)
{  for(j = 0; j< 3 ; j++)
        cout<<  A[i][j]<<"\t";
   cout<< endl;
}
cout<< "\n Matrix B: \n";
for( i =0; i< 3 ; i++)
```

```
{ for(j = 0; j< 2 ; j++)
     cout<<  B[i][j]<<"\t";
  cout<< endl;
}
/*  ......Product ...... */
cout<< "\n  Product AXB  : \n";
for(i = 0; i< 3 ; i++)
{   cout<<"\n";
    for(j = 0 ; j< 2 ; j++)
    {
          P[i][j] = A[i][j] * B[i][j];
          cout<<P[i][j]<<"\t";
    }
}
getch();
}
```

8. **Write a program to find the sum of squares of elements on the diagonal of a square matrix n x n.**

Ans.

```
#include<iostream.h>
#include<conio.h>
# define n 10
void main( )
{
      int mat[n][n];
      int d;
      long  left_diag = 0, right_diag = 0;
      // Input elements
      cout<<  "\n  Enter dimension of square matrix:";
      cin >> d;
      for(int i = 0; i < d ; i++)
      for( int j = 0; j < d ; j++)
      {cout<<"\n Enter elements for "<<  i+1 << "," << j+1<<"location :";
       cin >> mat[i][j];
      }
      clrscr();
      //Print the array
      for( i = 0; i < d ; i++)
           {  for( j = 0; j < d ; j++)
                 cout<<  mat[i][j]<<"\t";
              cout<< "\n";
           }
      //left diagonal ----
      for( i = 0; i < d ; i++)
       for( j = 0; j < d ; j++)
            if(i==j)
               left_diag += mat[i][j] * mat [i][j];
      //right diagonal ----
      for(i = 0; i < d ; i++)
           for( j = 0;j<d ;j++)
              if((i+j)==(d-1))
                 right_diag += mat[i][j] * mat [i][j];
      cout<< "\n Sum of the squares of left diagonal elements = "<<left_diag;
      cout<< "\n Sum of the squares of right diagonal elements = "<<right_diag;
      getch ( );
}
```

9. **Write a program to check equality of two given matrices with p x q and m x n dimensions.**

Ans.

```
#include<iostream.h>
```

```
#include<conio.h>
void main()
{    int mat1[10][10] , mat2[10][10];
     int m,n,p,q;
     // Input elements (matrix 1 ) ------
     cout<< "\n Enter no. of rows for matrix 1:";
     cin >> m;
     cout<< "\n Enter no. of columns :";
     cin >> n;
     for(int i = 0; i < m ; i++)
      for( int j = 0; j < n  ; j++)
     { cout<<"\n Enter elements for "<< i+1 << "," << j+1 <<"location :";
       cin >> mat1[i][j];
     }
     // Input elements (matrix 2 ) ------
     cout<< "\n Enter no. of rows for matrix 2:";
     cin >> p;
     cout<< "\n Enter no. of columns :";
     cin >> q;
    for(i = 0; i < p ; i++)
     for(j = 0; j < q ; j++)
     { cout<<"\n Enter elements for "<< i+1 << "," << j+1 <<"location :";
        cin >> mat2[i][j];
     }
     // check dimensions
     if (( m != n )||(p != q))
     {
          cout<< "\n Two matrices are unequal . ";
          getch ( );
          exit (1);
     }
     //check individual elements
     for( i = 0; i < m ; i++)
      for( j = 0; j < n; j++)
          if(mat1[i][j] != mat2[i][j])
           {
                cout<< "\n Two matrices are unequal . ";
                getch ( );
                exit (1);
            }
     cout<< " \n Two matrices are exactly equal . ";
     //Print the array
     cout<<"\n The matrix : \n\n";
     for( i = 0; i < m ; i++)
     {for( j = 0; j < n ; j++)
     cout<< mat1[i][j]<<"\t";
     cout<< "\n";
     }
     getch ( );
}
```