UNIT 7

ARRAY

A. One-Dimensional Array

1. WAP that reads 10 float numbers from user and displays the entered numbers in the screen.

```
#include<stdio.h>

int main(){

        float num[10];

        printf("Enter 10 numbers:");

        for(int i=0;i<10;i++){

                scanf("%f",&num[i]);

        }

        printf("Entered numbers are:\n");

        for(int i=0;i<10;i++){

                printf("num[%d] = %f\n",i,num[i]);

        }

        return 0;

}
```

2. WAP that accepts marks for 5 subjects from the keyboard and find sum and average.

```
#include<stdio.h>
int main(){
	int marks[5],sum=0;
	float average;
	printf("Enter marks in 5 subjects:");
	for(int i=0;i<5;i++){
		scanf("%d",&marks[i]);
		sum+=marks[i];
	}
	average=sum/5;
	printf("Sum = %d \nAverage = %.2f",sum,average);
	return 0;
}
```

3. WAP that accepts the price and stock of 5 different bulbs, calculate the total stock value.

```
#include<stdio.h>
int main(){
	int price[5],stock[5],total=0;
	for(int i=0;i<5;i++){
		printf("Enter the stock of bulb %d:",i+1);
		scanf("%d",&stock[i]);
		printf("Enter the price of bulb %d:",i+1);
		scanf("%d",&price[i]);
```

```
        total=total+stock[i]*price[i];
    }
    printf("The total stock value is %d.",total);
    return 0;
}
```

4. WAP to find the smallest and the largest element in the array.

```
#include<stdio.h>
int main(){
    int num[50],n,small,large;
    printf("Enter number of elements in array:");
    scanf("%d",&n);
    printf("Enter %d numbers:",n);
    for(int i=0;i<n;i++){
        scanf("%d",&num[i]);
    }
    printf("The elements of an array are:");
    for(int i=0;i<5;i++){
        printf("\t%d",num[i]);
    }
    small=num[0];
    large=num[0];
    for(int i=1;i<n;i++){
        if(small>num[i]){
            small=num[i];
        }
        if(large<num[i]){
            large=num[i];
        }
    }
    printf("\nSmallest number = %d \nLargest number = %d",small,large);
    return 0;
}
```

5. WAP for sorting elements of one-dimensional array in ascending order using selection sort.

```
#include<stdio.h>
int main(){
    int a[10],n,temp;
    printf("Enter number of elements in an array:");
    scanf("%d",&n);
    printf("Enter %d numbers:",n);
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Elements before sorting:");
    for(int i=0;i<n;i++){
        printf("\t%d",a[i]);
    }
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if(a[i]>a[j]){
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    printf("\nElements after sorting:\t");
    for(int i=0;i<n;i++){
        printf("\t%d",a[i]);
    }
    return 0;
}
```

6. WAP for sorting elements of one-dimensional array in ascending order using bubble sort.

```
#include<stdio.h>
int main(){
    int a[10],n,temp;
    printf("Enter number of elements in an array:");
    scanf("%d",&n);
    printf("Enter %d numbers:",n);
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Elements before sorting:");
    for(int i=0;i<n;i++){
        printf("\t%d",a[i]);
    }
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if(a[j]>a[j+1]){
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    printf("\nElements after sorting:\t");
    for(int i=0;i<n;i++){
        printf("\t%d",a[i]);
    }
    return 0;
}
```

7. WAP to illustrate the searching of an item in an array using sequential search that is searching in unordered array.

```
#include<stdio.h>
int main(){
    int a[10],n,key,i;
    printf("Enter numbers of elements in array:");
    scanf("%d",&n);
```

```
        printf("Enter %d numbers:",n);
        for(i=0;i<n;i++){
                scanf("%d",&a[i]);
        }
        printf("The entered numbers are:");
        for(i=0;i<n;i++){
                printf("\t%d",a[i]);
        }
        printf("\nEnter a number to be searched:");
        scanf("%d",&key);
        for(i=0;i<n;i++){
                if(a[i]==key){
                        printf("%d found at position %d.",key,i+1);
                        break;
                }
        }
        if(i==n){
                printf("%d was not found.",key);
        }
        return 0;
}
```

B. Two-Dimensional Array

8. WAP to demonstrate the loading or storing of data in two-dimensional array.

```
#include<stdio.h>

int main(){

    int a[3][4],i,j;

    for(i=0;i<3;i++){

        for(j=0;j<4;j++){

            a[i][j]=i+j;

        }

    }

    for(i=0;i<3;i++){

        for(j=0;j<4;j++){

            printf("a[%d][%d] = %d\t",i,j,a[i][j]);

        }

        printf("\n");

    }

    return 0;

}
```

9. WAP to read a matrix of size M*N from user and display it on screen.

```
#include<stdio.h>

int main(){

    int a[10][10],i,j,M,N;

    printf("Enter rows of matrix:");

    scanf("%d",&M);

    printf("Enter columns of matrix:");

    scanf("%d",&N);

    printf("Enter the elements of matrix:\n");

    for(i=0;i<M;i++){

        for(j=0;j<N;j++){

            scanf("%d",&a[i][j]);

        }

    }

    printf("Entered matrix is:\n");

    for(i=0;i<M;i++){

        for(j=0;j<N;j++){

            printf("%d\t",a[i][j]);

        }

        printf("\n");

    }

    return 0;

}
```

10. WAP to read two M*N matrices and display their sum or difference.

```
#include<stdio.h>

#define M 3

#define N 3

int main(){

        int a[M][N],b[M][N],i,j;

        printf("Enter the elements of 1st matrix:\n");

        for(i=0;i<M;i++){

                for(j=0;j<N;j++){

                        scanf("%d",&a[i][j]);

                }

        }

        printf("The 1st matrix is:\n");

        for(i=0;i<M;i++){

                for(j=0;j<N;j++){

                        printf("%d\t",a[i][j]);

                }

                printf("\n");

        }

        printf("Enter the elements of 2nd matrix:\n");

        for(i=0;i<M;i++){

                for(j=0;j<N;j++){

                        scanf("%d",&b[i][j]);

                }
        }
        printf("The 2nd matrix is:\n");
        for(i=0;i<M;i++){
                for(j=0;j<N;j++){
                        printf("%d\t",b[i][j]);
                }
                printf("\n");
        }
        printf("The sum of matrices is:\n");
        for(i=0;i<M;i++){
                for(j=0;j<N;j++){
                        printf("%d\t",a[i][j]+b[i][j]);
                }
                printf("\n");
        }
        return 0;
}
```

11. WAP to find transpose of a matrix.

```
#include<stdio.h>
#define M 3
#define N 3
int main(){
        int a[M][N],b[M][N],i,j;
        printf("Enter the elements matrix:\n");
```

```
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            scanf("%d",&a[i][j]);
        }
    }
    printf("The matrix to be transposed is:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    printf("The transposed matrix is:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            printf("%d\t",a[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

12. WAP to find the sum of squares in a diagonal of a square matrix.

```
#include<stdio.h>

#include<stdlib.h>

int main(){

	int a[10][10],i,j,M,N,sum=0;

	printf("Enter order of matrix:");

	scanf("%d%d",&M,&N);

	if(M!=N){

		printf("Not square matrix.\n");

		exit(0);

	}

	if(M>10 || N>10){

		printf("The order is out of range.\n");

		exit(0);

	}

	printf("Enter the elements matrix:\n");

	for(i=0;i<M;i++){

		for(j=0;j<N;j++){

			scanf("%d",&a[i][j]);

			if(i==j){

				sum+=a[i][j]*a[i][j];

			}

		}

	}
```

```
    printf("Entered matrix is:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    printf("Sum of squares in a diagonal is %d.",sum);
    return 0;
}
```

13. WAP to read two matrices from user and multiply them if possible.

```
#include<stdio.h>
int main(){
    int a[10][10],b[10][10],s[10][10];
    int m,n,x,y,i,j,k;
    printf("Enter number of rows in 1st matrix:");
    scanf("%d",&m);
    printf("Enter number of columns in 1st matrix:");
    scanf("%d",&n);
    printf("Enter number of rows in 2nd matrix:");
    scanf("%d",&x);
    printf("Enter number of columns in 2nd matrix:");
    scanf("%d",&y);
    if(n!=x){
        printf("Matrix multiplication is not possible.");
    }
    else{
        printf("Enter the elements of 1st matrix:\n");
        for(i=0;i<m;i++){
            for(j=0;j<n;j++){
                scanf("%d",&a[i][j]);
            }
        }
        printf("The 1st matrix is:\n");
        for(i=0;i<m;i++){
            for(j=0;j<n;j++){
                printf("%d\t",a[i][j]);
            }
            printf("\n");
        }
        printf("Enter the elements of 2nd matrix:\n");
        for(i=0;i<x;i++){
            for(j=0;j<y;j++){
                scanf("%d",&b[i][j]);
            }
        }
        printf("The 2nd matrix is:\n");
        for(i=0;i<x;i++){
            for(j=0;j<y;j++){
                printf("%d\t",b[i][j]);
            }
            printf("\n");
        }
        for(i=0;i<m;i++){
            for(j=0;j<y;j++){
                s[i][j]=0;
                for(k=0;k<n;k++){
                    s[i][j]=s[i][j]+a[i][k]*b[k][j];
                }
            }
        }
        printf("The multiplication of matrices is:\n");
        for(i=0;i<m;i++){
            for(j=0;j<y;j++){
                printf("%d\t",s[i][j]);
            }
            printf("\n");
        }
    }
    return 0;
}
```

14. Write a menu-driven program to input a 3*3 matrix and display the menu.

Menu:

- **Print the input matrix**
- **Sum of even values of elements**
- **Sum of all diagonal elements**
- **Exit**

```
#include<stdio.h>

#define M 3

#define N 3

int main(){

        int a[3][3],sum;

        int i,j,choice;

        printf("Enter the elements of matrix:\n");

        for(i=0;i<M;i++){

                for(j=0;j<N;j++){

                        scanf("%d",&a[i][j]);

                }

        }

        printf("\nEnter 1 to display matrix.");

        printf("\nEnter 2 to display sum of even values of elements.");

        printf("\nEnter 3 to display sum of all diagonal elements.");

        printf("\nEnter 4 to exit.");

        do{

                printf("\nEnter your choice:");

                scanf("%d",&choice);
```

```
switch(choice){
    case 1:
        printf("The matrix is:\n");
            for(i=0;i<M;i++){
                for(j=0;j<N;j++){
                    printf("%d\t",a[i][j]);
                }
            printf("\n");
            }
            break;
    case 2:
        sum=0;
        for(i=0;i<M;i++){
            for(j=0;j<N;j++){
                if(a[i][j]%2==0){
                    sum+=a[i][j];
                }
            }
        }
        printf("Sum of even values of elements = %d",sum);
        break;
    case 3:
        sum=0;
        for(i=0;i<M;i++){
```

```
                    for(j=0;j<N;j++){
                        if(i==j){
                            sum+=a[i][j];
                        }
                    }
                }
                printf("Sum of all diagonal elements = %d",sum);
                break;
            case 4:
                break;
            default:
                printf("Invalid option. Please try again.");
        }
    }while(choice!=4);
    return 0;
}
```

15. 25 numbers are entered through the keyboard into an array; WAP to find out how many of them are even and how many of them are odd.

```
#include<stdio.h>

#define SIZE 25

int main(){

        int a[SIZE];

        int i,evencount=0,oddcount=0;

        printf("Enter 25 numbers:\n");

        for(i=0;i<25;i++){

                scanf("%d",&a[i]);

                if(a[i]%2==0){

                        evencount++;

                }

                else{

                        oddcount++;

                }

        }

        printf("Even numbers = %d \nOdd numbers = %d",evencount,oddcount);

        return 0;

}
```

16. WAP that accepts the elements of 3*3 matrix and calculate the sum of all elements of the matrix.

```
#include<stdio.h>

#define M 3

#define N 3
```

```
int main(){
    int a[M][N],sum=0;
    int i,j;
    printf("Enter the elements of matrix:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            scanf("%d",&a[i][j]);
            sum+=a[i][j];
        }
    }
    printf("Sum of all elements = %d",sum);
    return 0;
}
```

17. WAP to read 4*4 matrix and find the sum of each row.

```
#include<stdio.h>
#define M 4
#define N 4
int main(){
    int a[M][N],sum;
    int i,j;
    printf("Enter the elements of matrix:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<M;i++){
        sum=0;
        for(j=0;j<N;j++){
            sum+=a[i][j];
        }
        printf("Sum of elements of %d th row is %d\n",i+1,sum);
    }
    return 0;
}
```

18. WAP to convert a 4*4 matrix to upper triangular and display the result in matrix form.

```
#include<stdio.h>
#define M 4
#define N 4
int main(){
    int a[M][N];
    int i,j;
    printf("Enter the elements of matrix:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            if(i>j){
                a[i][j]=0;
            }
        }
    }
    printf("Upper triangular matrix is:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

19. WAP to convert a 4*4 matrix to lower triangular and display the result in matrix form.

```
#include<stdio.h>
#define M 4
#define N 4
int main(){
    int a[M][N];
    int i,j;
    printf("Enter the elements of matrix:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            if(i<j){
                a[i][j]=0;
            }
        }
    }
    printf("Upper triangular matrix is:\n");
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

C. Strings

20. WAP to demonstrate string initialization.

```
#include<stdio.h>

int main(){

        char city[9]={'N','E','W',' ','Y','O','R','K','\0'};

        int i=0;

        while(city[i]!='\0'){

                printf("%c\t",city[i]);

                i++;

        }

        return 0;

}
```

21. WAP to demonstrate scanf() function which terminates its input on the first white space it encounters.

```
#include<stdio.h>

int main(){

        char name[20];

        printf("Enter your name:");

        scanf("%s",name);

        printf("Your name is:%s",name);

        return 0;

}
```

22. WAP to read and display capital letters only of name.

```
#include<stdio.h>
int main(){
	char name[20];
	printf("Enter your name in uppercase:");
	scanf("%[A-Z]",name);
	printf("Your name is: %s",name);
	return 0;
}
```

23. WAP that reads strings with blank spaces.

```
#include<stdio.h>
int main(){
	char name[20];
	printf("Enter your name:");
	scanf("%[^\n]",name);
	printf("Your name is: %s",name);
	return 0;
}
```

24. WAP to illustrate the use of strcpy() function.

```
#include<stdio.h>
#include<string.h>
int main(){
	char a[12]="Hello World";
	char b[50];
```

```
        strcpy(b,a);
        printf("Copied string = %s",b);
        return 0;
}
```

25. WAP to find the length of a string using library function strlen().

```
#include<stdio.h>
#include<string.h>
int main(){
        char name[50];
        int length;
        printf("Enter your name:");
        scanf("%s",name);
        length=strlen(name);
        printf("Length of name = %d.",length);
        return 0;
}
```

26. WAP to concatenate two strings using strcat() library functions.

```
#include<stdio.h>
#include<string.h>
int main(){
        char s1[10]="Happy ";
        char s2[]="New Year";
        printf("s1 = %s\ns2 = %s",s1,s2);
        printf("\nAfter concatenating = %s",strcat(s1,s2));
        return 0;
}
```

27. WAP to read a string from keyboard (until enter key) and count the number of vowels, consonants, spaces, semicolons and commas in that string.

```
#include<stdio.h>
#include<string.h>
int main(){
        char string[100];
        int i,vowels=0,consonants=0,spaces=0,semicolons=0,commas=0;
        printf("Enter a string:\n");
        gets(string);
        strlwr(string);
        for(i=0;string[i]!='\0';i++){

        if(string[i]=='a'||string[i]=='e'||string[i]=='i'||string[i]=='o'||string[i]=='u'){
                        vowels++;
                }
                else if(string[i]==' '){
                        spaces++;
                }
                else if(string[i]==';'){
                        semicolons++;
                }
                else if(string[i]==','){
                        commas++;
```

```
            }
            else{
                consonants++;
            }
        }
        printf("\nNumber of vowels = %d",vowels);
        printf("\nNumber of consonants = %d",consonants);
        printf("\nNumber of spaces = %d",spaces);
        printf("\nNumber of semicolons = %d",semicolons);
        printf("\nNumber of commas = %d",commas);
        return 0;
}
```

28. WAP to reverse a string using library functions:

```
#include<stdio.h>
#include<string.h>
int main(){
        char string[100];
        printf("Enter a string:\n");
        gets(string);
        strrev(string);
        printf("String after reversing = %s",string);
        return 0;
}
```

29. WAP to read a string and count the number of vowels and consonants in it.

```
#include<stdio.h>

#include<string.h>

int main(){

        char string[100];

        int i,vowels=0,consonants=0;

        printf("Enter a string:\n");

        gets(string);

        strlwr(string);

        for(i=0;string[i]!='\0';i++){


        if(string[i]=='a'||string[i]=='e'||string[i]=='i'||string[i]=='o'||string[i]=='u'){

                        vowels++;

                }

                else{

                        consonants++;

                }

        }

        printf("Number of vowels = %d\nNumber of consonants =
%d",vowels,consonants);

        return 0;

}
```

30. WAP to check whether a given string is a palindrome or not.

```
#include<stdio.h>

#include<string.h>

int main(){

    char string1[100],string2[100];

    printf("Enter a string:\n");

    gets(string1);

    strcpy(string2,string1);

    strrev(string1);

    if(strcmp(string1,string2)==0){

        printf("Palindrome.");

    }

    else{

        printf("Not Palindrome.");

    }

    return 0;

}
```

31. WAP to convert a string to uppercase using strupr() library functions.

```
#include<stdio.h>
#include<string.h>
int main(){
	char string[100];
	printf("Enter a string:\n");
	gets(string);
	strupr(string);
	printf("String in uppercase = %s",string);
	return 0;
}
```

32. WAP to convert a string to lowercase using strlwr() library function.

```
#include<stdio.h>
#include<string.h>
int main(){
	char string[100];
	printf("Enter a string in uppercase:\n");
	gets(string);
	strlwr(string);
	printf("String in lowercase = %s",string);
	return 0;
}
```

33. WAP to compare two strings using strcmp() library function.

```
#include<stdio.h>

#include<string.h>

int main(){

        char string1[100],string2[100];

        int result;

        printf("Enter 1st string:\n");

        gets(string1);

        printf("Enter 2nd string:\n");

        gets(string2);

        result=strcmp(string1,string2);

        if(result==0){

                printf("Two strings are identical.");

        }

        else if(result>0){

                printf("%s is greater than %s.",string1,string2);

        }

        else{

                printf("%s is greater than %s.",string2,string1);

        }

        return 0;

}
```