Module 3

Array

An array in C is a collection of items stored at contiguous memory locations and elements can be accessed randomly using indices of an array. They are used to store similar type of elements as in the data type must be the same for all elements. They can be used to store collection of primitive data types such as int, float, double, char, etc of any particular type.

A	Indices	0	1	2	3
	Array elements	10	20	30	40

Why do we need arrays?

We can use normal variables (v1, v2, v3) when we have a small number of objects, but if we want to store a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.

Different ways of declaring an Array

1. **Array declaration by specifying size**

   Eg: int A[10];

2. **Array declaration by initializing elements**

   Eg: int A[] = { 10, 20, 30, 40 }

3. **Array declaration by specifying size and initializing elements**

   Eg: int A[6] = { 10, 20, 30, 40 }

Advantages of an Array:

1. Random access of elements using array index.
2. Use of less line of code as it creates a single array of multiple elements.
3. Easy access to all the elements.
4. Traversal through the array becomes easy using a single loop.
5. Sorting becomes easy as it can be accomplished by writing less line of code.

Disadvantages of an Array:

- It allows us to enter only fixed number of elements into it. We cannot alter the size of the **array** once **array** is declared. Hence if we need to insert more number of records than declared then it is not possible. We should know **array** size at the compile time itself.
- Inserting and deleting the records from the array would be costly since we add / delete the elements from the array, we need to manage memory space too.
- It does not verify the indexes while compiling the array. In case there is any indexes pointed which is more than the dimension specified, then we will get run time errors rather than identifying them at compile time.

Eg: int a[3]={1,2,3}
printf("%d",A[3]);
Output: 3245431 (Garbage Value)

1) **Write a program to read and display an array of size n.**

```
#include<stdio.h>
void main()
{
        int n,A[20],i;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        for(i=0;i<n;i++)
        {
                printf("%d\t",A[i]);
        }
}
```

2) **Write a program to read and display an array of size n in reverse order.**

```
#include<stdio.h>
void main()
{
        int n,A[20],i;
        printf("Enter the size of Array");
        scanf("%d",&n);
```

```
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        for(i=n-1;i>=n;i--)
        {
                printf("%d\t",A[i]);
        }
}
```

3) Write a program to find sum and average of an array content of size n.

```
#include<stdio.h>
void main()
{
        int n,A[20],i,sum;
        float avg;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        sum=0;
        for(i=0;i<n;i++)
        {
                sum = sum + i;
        }
        avg = (float)sum/n;
        printf("Sum=%d",sum);
        printf("Average=%f",avg);
}
```

4) Write a program to read and display even display numbers in an array of size n.

```
#include<stdio.h>
void main()
{
        int n,A[20],i;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
```

```
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        for(i=0;i<n;i++)
        {
                if(A[i]%2 == 0)
                {
                        printf("%d\t",A[i]);
                }
        }
}
```

5) Write a program to read and display prime numbers in an array of size n.

```
#include<stdio.h>
void main()
{
        int n,A[20],i,flag,j;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        for(i=0;i<n;i++)
        {
                flag=0;
                for(j=2;j<=A[i]/2;j++)
                {
                        if(A[i]%j == 0)
                        {
                                flag=1;
                                break;
                        }
                }
                if(flag == 0)
                {
                        printf("%d\t",A[i]);
                }
        }
}
```

6) Write a program to search an element in array of size n.

```
#include<stdio.h>
void main()
{
        int n,A[20],i,index;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
        printf("Enter the element to search");
        scanf("%d",&key);
        index = -1;
        for(i=0;i<n;i++)
        {
                if(A[i] == key)
                {
                        index = i;
                        break;
                }
        }
        if(index == -1)
        {
                printf("No search element found");
        }
        else
        {
                prinft("Search Element found at index %d",index);
        }
}
```

7) Write a program to find largest element in array of size n.

```
#include<stdio.h>
void main()
{
        int n,A[20],i,large;
        printf("Enter the size of Array");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the number");
                scanf("%d",&A[i]);
        }
```

```
        large=A[0];
        for(i=0;i<n;i++)
        {
                if(large < A[i])
                {
                        large = A[i]
                }
        }
        printf("Large =%d",large);
}
```

8) Write a program to sort element in array of size n.

```
#include<stdio.h>
void main()
{
    int n,A[20],i,j,temp;
    printf("Enter the size of Array");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
            printf("Enter the number");
            scanf("%d",&A[i]);
    }
    printf("Before Sorting");
    for(i=0;i<n;i++)
    {
            printf("%d\t",A[i]);
    }
    for(i=0;i<n-1;i++)
    {
            for(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("Before Sorting");
    for(i=0;i<n;i++)
    {
            printf("%d\t",A[i]);
    }
```

}

String

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character '\0'.

Index	0	1	2	3	4	5
Character Array	'H'	'E'	'L'	'L'	'O'	
String	'H'	'E'	'L'	'L'	'O'	'\0'

Declaration of strings: Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string.

char str[20];

In the above syntax str is any name given to the string variable and 20 is used define the length of the string, i.e. the number of characters strings will store. Please keep in mind that there is an extra terminating character which is the Null character ('\0') used to indicate termination of string which differs strings from normal character arrays.

Initializing a String:

A string can be initialized in different ways.

1. *char str[] = "Programming";*
2. *char str[50] = "Programming";*
3. *char str[] = {'P','r','o','g','r','a','m','m','i','n','g','\0'};*
4. *char str[12] = {'P','r','o','g','r','a','m','m','i','n','g','\0'};*

Read a character from User

1) scanf()

- It is used to read the input (character, string, numeric data) from the standard input (keyboard).
- It is used to read the input until it encounters a whitespace, newline or End Of File (EOF).

```
#include <stdio.h>
void main()
{
   char str[20];
   printf("Enter the String\n");
   scanf("%s", str);
   printf("String: %s\n", str);

}
```

Output

Enter the String : Hello World

String: Hello

Issue with scanf: There is a whitespace after Hello. So it read the input till Hello and store it in str.

Solution: use % [^\n] instead of %s.

```
#include <stdio.h>
void main()
{
   char str[20];
   printf("enter the string\n");
   scanf("%[^\n]", str);
   printf("String: %s\n", str);

}
```

2) Gets

- It is used to read input from the standard input (keyboard).
- It is used to read the input until it encounters newline or End Of File (EOF).

```
#include <stdio.h>
void main()
{
   char str[20];
   printf("Enter the String\n");
   gets( str);
   printf("String: %s\n", str);

}
```

Output

Enter the String: Hello World

String: Hello World

Issue: it reads string from standard input and prints the entered string, but it suffers from Buffer Overflow as gets() doesn't do any array bound testing. gets() keeps on reading until it sees a newline character.

Display String value – printf and puts

#include <stdio.h>
void main()
{
char str[20];
printf("Enter the String\n");
gets(str);
printf("%s\n", str);
}
	#include <stdio.h>
	void main()
	{
	char str[20];
	printf("Enter the String\n");
	gets(str);
	puts(str);
	}

Output Enter the String: Hello world Hello world	**Output** Enter the String: Hello world Hello world

In-built String Function

Function	Work of Function
strlen()	computes string's length
strcpy()	copies a string to another
strcat()	concatenates(joins) two strings
strcmp()	compares two strings case sensitive checking
strcmpi()	compares two strings case insensitive checking
strlwr()	converts string to lowercase
strupr()	converts string to uppercase

strlen vs sizeof

strlen returns the length of the string stored in array, however sizeof returns the total allocated size assigned to the array.

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str[]="Hello World";
        int len=strlen(str);
        int size=sizeof(str);
        printf("Value=%d",len*size);
}
```

len value will the length of the string which is 11. But size value will be 12 including the end of character.

Output

Value=132

strcat

It concatenates two strings and returns the concatenated string.

Syntax: strcat(string1,string2) – concatenate string1 with string2.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char s1[10] = "Hello";
    char s2[10] = "World";
    strcat(s1,s2);
    printf("After concatenation: %s", s1);
}
```

Output:

After concatenation: HelloWorld

strcpy

It copies the string str2 into string str1, including the end character (terminator char '\0').

Syntax: strcpy(string1,string2) - copy the content of string2 to string1

```
#include <stdio.h>
#include <string.h>
void main()
```

```
{
    char s1[30];
    char s2[30] = "Hello World";
    strcpy(s1,s2);
    printf("String s1: %s", s1);
}
```

Output:

String s1 is: Hello World

strcmp

It compares the two strings and returns an integer value. If both the strings are same (equal) then this function would return 0 otherwise it may return a negative or positive value based on the comparison.

strcmp(string1,string2) -- Compares the content of string1 and string2

- If string1 < string2 OR string1 is a substring of string2 then it would result in a negative value.
- If string1 > string2 then it would return positive value.
- If string1 == string2 then you would get 0(zero) when you use this function for compare strings.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char s1[20] = "Hello";
    char s2[20] = "World";
    if (strcmp(s1, s2) ==0)
    {
      printf("string 1 and string 2 are equal");
    }
  else
  {
      printf("string 1 and 2 are different");
  }
}
```

Output:

string 1 and 2 are different

- **Write a program to display the content of string in upper and lower case.**

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str[20];
        printf("Enter the string");
        gets(str);
        printf("Upper Case String:%s",strupr(str));
        printf("Lower Case String:%s",strlwr(str));
}
```

Output

Enter the string Hello World

Upper Case String: HELLO WORLD

Lower Case String: hello world

- **Write a program to display reverse the content of string.**

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str[20];
        int len,i,j;
        printf("Enter the string");
        gets(str);
        i=strlen(str)-1;
        j=0;
        while(i>=0)
        {
                rev[j]=str[i];
                j++;
                i--;
        }
        rev[j]='\0';
        printf("Reverse=%s",rev);
}
```

- **Write a program to check whether a string is palindrome or not without using in built functions.**

```
#include<stdio.h>
```

```
#include<string.h>
void main()
{
	char str[20];
	int len,i,j,flag;
	printf("Enter the string");
	gets(str);
	// Code to find string length
	for(len=0;str[len]!='\0';len++);
	// Code to check the Palindrome
	flag=0;
	for(i=0;i<len/2;i++)
	{
		if(str[i] != str[len-1-i])
		{
			flag=1;
			break;
		}
	}
	if(flag == 0)
	{
		printf("Palindrome");
	}
	else
	{
		printf("Not Palindrome");
	}
}
```

- **Write a program to count number of vowels in a given string.**

```
#include<stdio.h>
#include<string.h>
void main()
{
	char str[20];
	int i,count;
	printf("Enter the string");
	gets(str);
	count=0;
	for(i=0;i<strlen(str);i++)
	{
		switch(str[i])
		{
			case 'A':
			case 'a':
```

```
                        case 'E':
                        case 'e':
                        case 'i':
                        case 'I':
                        case 'O':
                        case 'o':
                        case 'U':
                        case 'u':count++;
                                        break;
                }
        }
        printf("Count of vowels=%d",count);
}
```

- **Write a program to length of string without in built function.**

```
#include<stdio.h>
void main()
{
        char str[30];
        int i=0;
        printf("Enter the String:");
        scanf("%s",str);
        while(str[i] != '\0')
        {
                i++;
        }
        printf("Length = %d",i);
}
```

Output
Enter the String:Sangeeth
Length = 8

- **Write a program to copy the content of a string to another location without using in built function.**

```
#include<stdio.h>
void main()
{
        char s1[30],s2[30];
        int i=0;
        printf("Enter the String:");
        scanf("%s",s1);
        while(s1[i] != '\0')
```

```
        {
                s2[i]=s1[i];
                i++;
        }
        s2[i]='\0';
        printf("Copied String = %s",s2);
}
```

Output
Enter the String:Sangeeth
Copied String = Sangeeth

- **Write a program to concatenate the content of a string with another location without using in built function.**

```
#include<stdio.h>
void main()
{
        char s1[30],s2[30],s3[50];
        int i=0,j=0;
        printf("Enter the String 1:");
        scanf("%s",s1);
        printf("Enter the String 2:");
        scanf("%s",s2);
        while(s1[i] != '\0')
        {
                s3[i]=s1[i];
                i++;
        }
        while(s2[j] != '\0')
        {
                s3[i]=s2[j];
                i++;
                j++;
        }
        s3[i]='\0';
        printf("Concatenated String = %s",s3);
}
```

Output

Enter the String 1:Sangeeth
Enter the String 2:Nagarajan
Concatenated String = SangeethNagarajan

Two Dimensional Array

An array of arrays is known as 2D array. The two dimensional (2D) array is also known as matrix. A matrix can be represented as a table of rows and columns. The syntax to declare the 2D array is given below.

data_type array_name[rows][columns];

Initialization of Two Dimensional Array

There are two ways to initialize a two Dimensional arrays during declaration.

1. int A[2][3] = { {10, 11, 12},{14, 15, 16}};
   int A[][3] = { {10, 11, 12},{14, 15, 16}};
2. int A[2][3] = { 10, 11, 12, 14, 15, 16};
   int A[][3] = { 10, 11, 12, 14, 15, 16};

When we initialize a 1 dimensional array during declaration, we need not to specify the size of it. However that's not the case with 2D array, you must always specify the second dimension even if you are specifying elements during the declaration.

/* Valid declaration*/

int A[2][2] = {1, 2, 3 ,4 }

/* Valid declaration*/

int A[][2] = {1, 2, 3 ,4 }

/* Invalid declaration – you must specify second dimension*/

int A[][] = {1, 2, 3 ,4 }

/* Invalid because of the same reason mentioned above*/

int A[2][] = {1, 2, 3 ,4 }

Structure of two dimensional array of size - A[3][5]

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]
A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[1][4]
A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]

1. **Write a program to read a matrix of size mXn.**

```
#include<stdio.h>
void main()
{
	int m,n,A[20][20],i,j;
	printf("Enter the order of matrix");
	scanf("%d%d",&m,&n);

	// Row
	for(i=0;i<m;i++)
	{
		//Column
		for(j=0;j<n;j++)
		{
			printf("Enter the element");
			scanf("%d",&A[i][j]);
		}
	}
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%d\t",A[i][j]);
		}
		printf("\n");
	}
}
```

2. **Write a program to read a matrix of size mXn and display its transpose.**

```
#include<stdio.h>
void main()
{
	int m,n,A[20][20],i,j;
	printf("Enter the order of matrix");
	scanf("%d%d",&m,&n);
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Enter the element");
			scanf("%d",&A[i][j]);
		}
	}
```

```
        printf("Transpose of Matrix A\n");
        for(j=0;j<n;j++)
        {
                for(i=0;i<m;i++)
                {
                        printf("%d\t",A[i][j]);
                }
                printf("\n");
        }
}
```

3. Write a program to read a matrix of size mXn and display its sum of diagonal.

```
#include<stdio.h>
void main()
{
        int m,n,A[20][20],i,j;
        printf("Enter the order of matrix");
        scanf("%d %d",&m,&n);
        for(i=0;i<m;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("Enter the element");
                        scanf("%d",&A[i][j]);
                }
        }
        for(i=0;i<m;i++)
        {
                sum = sum + A[i][i];
        }
        printf("Sum=%d",sum);
}
```

4. Write a program to add the content of two matrices.

```
#include<stdio.h>
#include<stdlib.h> // used for exit(1)
void main()
{
        int r1,c1,r2,c2,A[20][20],B[20][20],C[20][20],i,j;
        printf("Enter the order of matrices:");
        scanf("%d%d%d%d",&r1,&c1,&r2,&c2);
        if(r1 != r2 || c1 != c2)
        {
                printf("Matrices Addition Not possible");
```

```
                exit(1);
        }
        printf("Enter the Matrix A\n");
        for(i=0;i<r1;i++)
        {
                for(j=0;j<c1;j++)
                {
                        printf("Enter the element A[%d][%d]:",i,j);
                        scanf("%d",&A[i][j]);
                }
        }
        printf("Enter the Matrix B\n");
        for(i=0;i<r2;i++)
        {
                for(j=0;j<c2;j++)
                {
                        printf("Enter the element B[%d][%d]:",i,j);
                        scanf("%d",&B[i][j]);
                }
        }
        printf("Added Matrix C:\n");
        for(i=0;i<r1;i++)
        {
                for(j=0;j<c1;j++)
                {
                        C[i][j]=A[i][j] + B[i][j];
                        printf("%d\t",C[i][j]);
                }
                printf("\n");
        }
}
```

Output

```
Enter the order of matrices: 2 2 2 2
Enter the Matrix A
Enter the element A[0][0]:1
Enter the element A[0][1]:2
Enter the element A[1][0]:3
Enter the element A[1][1]:4
Enter the Matrix B
Enter the element B[0][0]:5
Enter the element B[0][1]:6
Enter the element B[1][0]:7
Enter the element B[1][1]:8
Added Matrix C:
```

6 8
10 12

5. Write a program to perform matrix multiplication.

```
#include<stdio.h>
#include<stdlib.h> // used for exit(1)
void main()
{
    int r1,c1,r2,c2,A[20][20],B[20][20],C[20][20],i,j,k;
    printf("Enter the order of matrices:");
    scanf("%d%d%d%d",&r1,&c1,&r2,&c2);
    if(c1 != r2)
    {
        printf("Matrices Multiplication Not possible");
        exit(1);
    }
    printf("Enter the Matrix A\n");
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            printf("Enter the element A[%d][%d]:",i,j);
            scanf("%d",&A[i][j]);
        }
    }
    printf("Enter the Matrix B\n");
    for(i=0;i<r2;i++)
    {
        for(j=0;j<c2;j++)
        {
            printf("Enter the element B[%d][%d]:",i,j);
            scanf("%d",&B[i][j]);
        }
    }
    printf("Matrix C:\n");
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c2;j++)
        {
            C[i][j]= 0;
            for(k=0;k<c1;k++)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
            printf("%d\t",C[i][j]);
```

```
            }
            printf("\n");
        }
    }
```

Output

```
Enter the order of matrices:2 2 2 2
Enter the Matrix A
Enter the element A[0][0]:1
Enter the element A[0][1]:2
Enter the element A[1][0]:3
Enter the element A[1][1]:4
Enter the Matrix B
Enter the element B[0][0]:5
Enter the element B[0][1]:6
Enter the element B[1][0]:7
Enter the element B[1][1]:8
Matrix C:
19      22
43      50
```

Example of Two dimensional Character Array

- **Write a program to read name of n persons and display those names.**

```
#include<stdio.h>
#include<string.h>
void main()
{
        int n,i,j;
        char names[30][30],temp[30];
        printf("Enter the value of n:");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the name:");
                scanf("%s",names[i]);
        }
        printf("Names\n");
        for(i=0;i<n;i++)
        {
                printf("%s\n",names[i]);
        }
}
```

Output
Enter the value of n:4
Enter the name:Sangeeth
Enter the name:Karthi
Enter the name:Poornima
Enter the name:Chandrika
Names
Sangeeth
Karthi
Poornima
Chandrika

- **Write a program to read name of n persons and display those name in alphabetical order.**

```
#include<stdio.h>
#include<string.h>
void main()
{
	int n,i,j;
	char names[30][30],temp[30];
	printf("Enter the value of n:");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("Enter the name:");
		scanf("%s",names[i]);
	}
	for(i=0;i<n-1;i++)
	{
		for(j=0;j<n-i-1;j++)
		{
			if(strcmp(names[j],names[j+1])>0)
			{
				strcpy(temp,names[j]);
				strcpy(names[j],names[j+1]);
				strcpy(names[j+1],temp);
			}
		}
	}
	printf("Names in Ascending order\n");
	for(i=0;i<n;i++)
	{
		printf("%s\n",names[i]);
	}
}
```

Output

```
Enter the value of n:4
Enter the name:Sangeeth
Enter the name:Karthi
Enter the name:Poornima
Enter the name:Chandrika
Names in Ascending order
Chandrika
Karthi
Poornima
Sangeeth
```