Module 5:

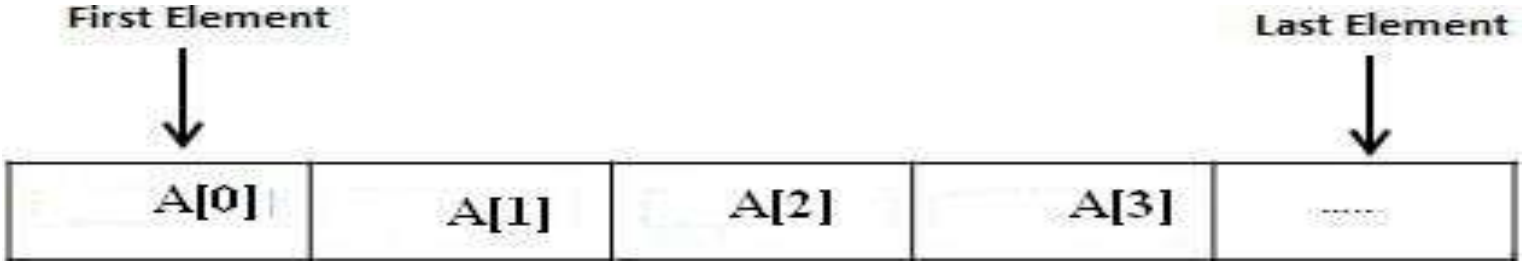
Array and Strings

Array

- **An array**
  - Is a derived data type
  - Collection of similar type of elements
  - Stored in continuous memory locations
  - Remain the same size once they are created
    - *Fixed-length* entries

Array

- □ Instead of declaring individual variables, such as A0, A1,A2,….A99, you declare one array variable such as A[100] and use A[0], A[1], and ..., A[99] to represent individual variables.
- □ A specific element in an array is accessed by an index.
- □ Array index starts with 0 and ends with size-1.
- □ All arrays consist of contiguous memory locations.
- □ The lowest address corresponds to the first element and the highest address to the last element.



Array Declaration

- To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:

    - datatype arrayName [arraySize];

- This is called a *single-dimensional* array. The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type.

Eg:- double balance[10];
int A[5];
char Name[10];

Initializing Arrays

- You can initialize array in C either one by one or using a single statement as follows:

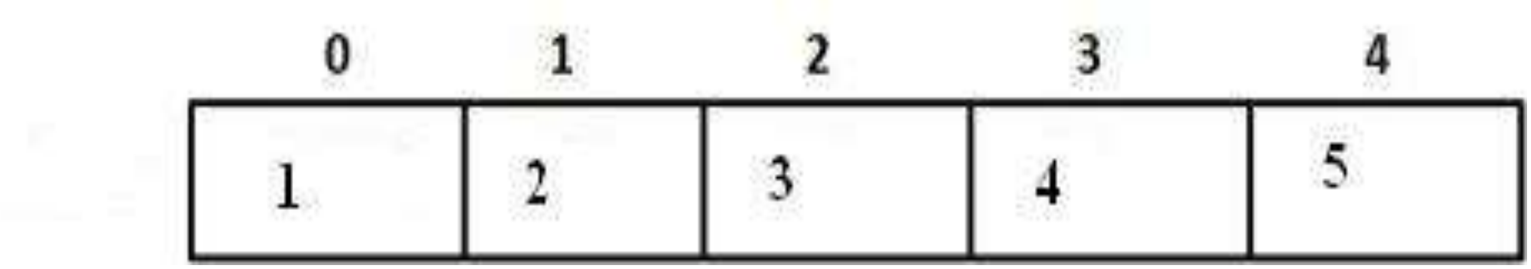
  double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};

  int A[5]={1,2,3,4,5};

- The number of values between braces { } can not be larger than the number of elements that we declare for the array between square brackets [ ].
- If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write:

  int A[]={1,2,3,4,5};

Here, Size of array is 5.



0	1	2	3	4
1	2	3	4	5

Accessing Array Elements

- An element is accessed by indexing the array name.

```
int num = A[4];
```

- To read array elements

```
for(i=0;i<size;i++)
    {
            scanf(“%d”,&A[i]);
}
```

- To display array elements

```
for(i=0;i<size;i++)
    {
            printf(“%d\t”,A[i]);
    }
```

WAP to read and display 10 integer type array elements.

```
#include <stdio.h>
void main ()
{
int n[ 10 ]; // n is an array of 10 integers
int i,j; // initialize elements of array n to 0
for ( i = 0; i < 10; i++ )
{
n[ i ] = i + 1 //set element at location i to i + 1
}
 /* output each array element's value */
for (j = 0; j < 10; j++ )
{
printf("Element[%d] = %d\n", j, n[j] );
}
}
```

Output

```
Element[0]=1
Element[1]=2
Element[2]=3
Element[3]=4
Element[4]=5
Element[5]=6
Element[6]=7
Element[7]=8
Element[8]=9
Element[9]=10
```

WAP to calculate sum and average of array elements

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,a[100],sum=0;
    float avg;
    clrscr();
    printf("Enter the number of elements:");
    scanf("%d",&n);
    for(i=0;i<=n-1;i++)
    {
            printf("Enter a value:");
            scanf("%d",&a[i]);
    }
for(i=0;i<=n-1;i++)
    {
            sum=sum+a[i];
    }
    avg=sum/n;
    printf("The  sum is %d and average
    is  %f",sum,avg);
    getch();
}
```

```
Output
Enter the number of elements: 5
Enter a value:10
Enter a value:20
Enter a value:30
Enter a value:40
Enter a value:50
The sum is 150 and average is 30
```

WAP to search an element in array

```
#include<stdio.h>
#include<conio.h>
void main()
{    int n,i,a[100],x,index=0;
     clrscr();
     printf("Enter the number of elements:");
     scanf("%d",&n);
printf("Enter array elements:");
     for(i=0;i<=n-1;i++)
     {
              scanf("%d",&a[i]);
     }
     printf("Enter the element to be searched:");
     scanf("%d",&x);
```

```
for(i=0;i<=n-1;i++)
     {
              if(x==a[i])
              {
                            index=1;
                            break;
     }
     }
if(index!=0)
{     printf("The element is found at
     position %d",(i+1));}
     else
{    printf("Not Found");
}    getch();
}
```

WAP to search an element in array

Output

```
Enter the number of elements: 5
Enter array elements:
10
11
12
13
14
Enter the element to be searched: 14
The element is found at position 5
```

WAP to sort array elements in ascending order

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,j,a[10],temp;
    clrscr();
    for(i=0;i<10;i++)
    {
           printf("Enter array elements:");
           scanf("%d",&a[i]);
    }
```

```
for(i=0;i<10;i++)
{    for(j=i+1;j<10;j++)
     {
            if(a[i]>a[j])
            {          temp=a[i];
                       a[i]=a[j];
                       a[j]=temp;
            }
     }
}
printf("Sorted array is:\n:");
for (i = 0; i < n; i++ )
{
printf(" %d\t", a[i] );
}
}
```

WAP to sort array elements in ascending order

Output

```
Enter the number of elements: 5
Enter array elements:
5
3
2
1
4
Sorted array is: 1 2 3 4 5
```

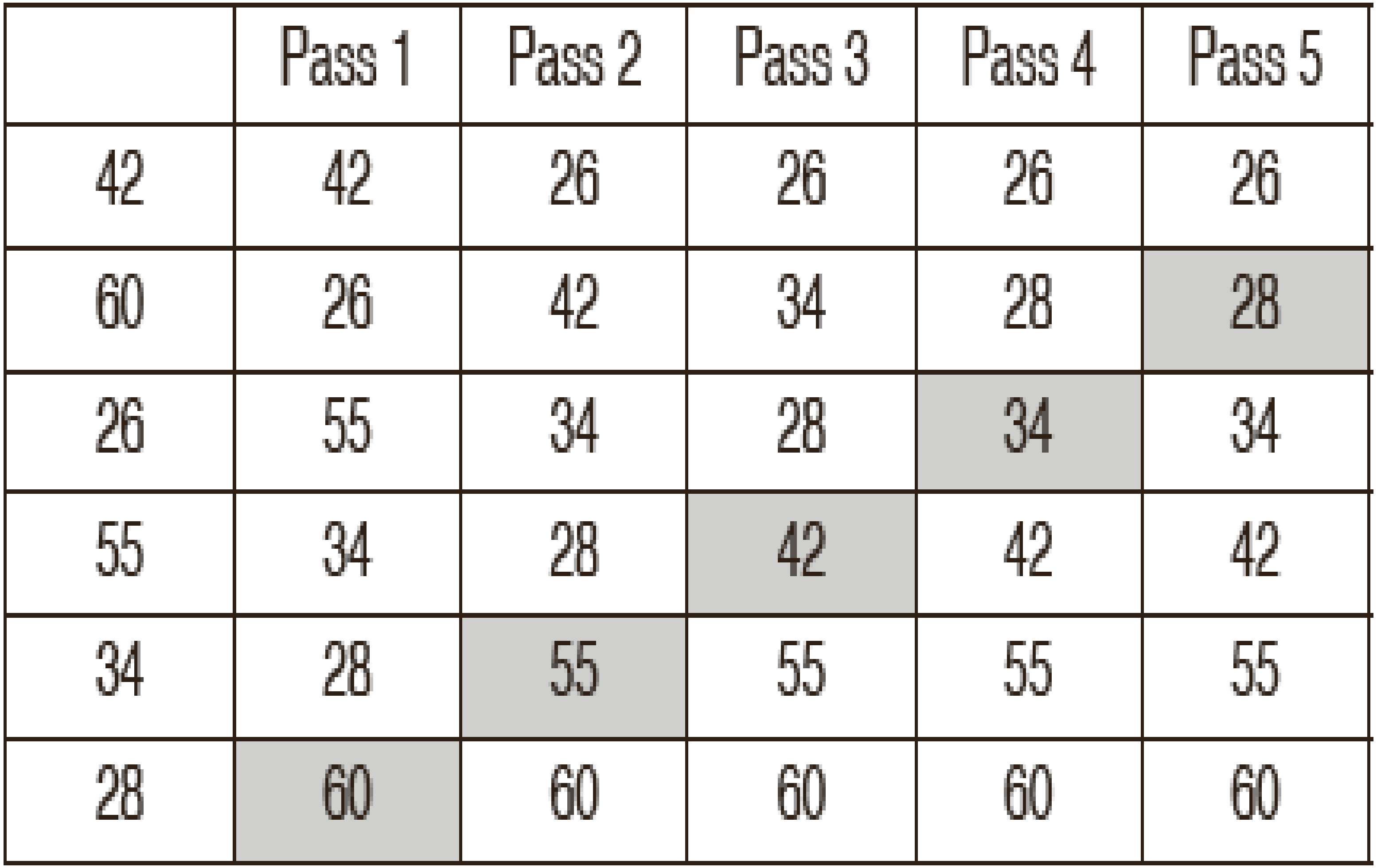
Bubble sort

- ***A bubble sort compares adjacent array elements*** and exchanges their values if they are out of order.
- In this way, the smaller values ‘bubble’ to the top of the array (towards element 0), while the larger values sink to the bottom of the array.
- This sort continues until no exchanges are performed in a pass.

Bubble sort example

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
42	42	42	42	42	42
60	60	26	26	26	26
26	26	60	55	55	55
55	55	55	60	34	34
34	34	34	34	60	28
28	28	28	28	28	60

Bubble sort example



	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5
42	42	26	26	26	26
60	26	42	34	28	28
26	55	34	28	34	34
55	34	28	42	42	42
34	28	55	55	55	55
28	60	60	60	60	60

WAP to sort array elements in ascending order using bubble sort

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,j,a[100],temp;
    clrscr();
    printf("Enter the number of elements:");
    scanf("%d",&n);
    for(i=0;i<=n-1;i++)
    {
            printf("Enter array Element:");
            scanf("%d",&a[i]);
    }
for(i=0;i<n;i++)
{
    for(j=0;j<=n-1;j++)
    {
            if(a[j]>a[j+1])
            {
                    temp=a[j];
                    a[j]=a[j+1];
                    a[j+1]=temp;
            }
    }
}
printf("Sorted array is:\n:");
for (i = 0; i < n; i++ )
{
printf(" %d\t", a[i] );
}
}
```

WAP to sort array elements in ascending order using bubble sort

Output

```
Enter the number of elements: 5
Enter array elements:
5
3
2
1
4
Sorted array is: 1 2 3 4 5
```

Multi-dimensional array

- C programming language allows multidimensional arrays. Here is the general form of a multidimensional array declaration:

  datatype arrayname[size1][size2]...[sizeN];

Eg:-

int A[3][3];

char name[10][5];

float B[3][3];

2-Dimensional array

```
int A[3][3];
```

COL

ROW

	0	**1**	**2**
0	A[0][0]	A[0][1]	A[0][2]
1	A[1][0]	A[1][1]	A[1][2]
2	A[2][0]	A[2][1]	A[2][2]

Memory Representation of 2-D array

A[0][0]	**A[0][1]**	**A[0][2]**	**A[1][0]**	**A[1][1]**	**A[1][2]**	**A[2][0]**	**A[2][1]**	**A[2][2]**
ROW 0			ROW 1			ROW 2		

2-D Array initialization

```
int A[][]={{1,2,3},
           {4,5,6},
           {7,8,9}};
```

Size of A is 3X3.

```
int B[][3]={{1,1,1},
            {2,2,2}};
```

Size of B is 2X3.

Note- size of row is optional

Accessing 2-D Array Elements

To read 2-D Array elemnts

```
for(i=0;i<row;i++)
   {
         for(j=0;j<col;j++)
         {
         scanf(“%d”,&A[i][j]);
         }
}
```

To display 2-D Array elements

```
for(i=0;i<row;i++)
   {
         for(j=0;j<col;j++)
         {
         printf(“%d\t”,A[i][j]);
         }
   printf(“\n”);
   }
```

WAP to read and display elements of integer type 2-D array.

```
#include <stdio.h>
void main ()
{
int A[3][3]; // A is an array of 3X3 size
int i,j;
Printf(“ENTER ARRAY
    ELEMENTS:\n”);
for(i=0;i<3;i++)
    {
            for(j=0;j<3;j++)
            {
            scanf(“%d”,&A[i][j]);
            }
}
```

```
/* output each array element's
    value */
for(i=0;i<3;i++)
    {
            for(j=0;j<3j++)
            {

            printf(“Element[%d][%d]=
    %d\t”,i,j,A[i][j]);
            }
    printf(“\n”);
    }
```

Output

```
ENTER ARRAY ELEMENTS:
1 2 3 4 5 6 7 8 9


Element[0][0]=1        Element[0][1]=2        Element[0][2]=3
Element[1][0]=4        Element[1][1]=5        Element[1][2]=6
Element[2][0]=7        Element[2][1]=8        Element[2][2]=9
```

WAP to display addition of two matrices of size mxn

```
#include <stdio.h>
void main ()
{
int A[3][3],B[3][3],C[3][3];
int i,j,m,n;
printf("ENTER ARRAY
    ELEMENTS FOR A:\n");
```

```
for(i=0;i<3;i++)
    {       for(j=0;j<3;j++)
            {
            scanf("%d",&A[i][j]);
            }
}
printf("ENTER ARRAY
    ELEMENTS FOR B:\n");
for(i=0;i<3;i++)
    {       for(j=0;j<3;j++)
            {
            scanf("%d",&B[i][j]);
            }
}
```

```
for(i=0;i<3;i++)
    {
           for(j=0;j<3;j++)
           {
           C[i][j]=A[i][j]+B[i][j];
           }
}
printf(“Matrix addition is\n”);
```

```
for(i=0;i<3;i++)
    {
           for(j=0;j<3;j++)
           {
           printf(“%d”,C[i][j]);
           }
printf(“\n”);
}
getch();
}
```

OUTPUT

```
Enter size of matrices 3 3
ENTER ARRAY ELEMENTS FOR A:
1         1         1
1         1         1
1         1         1
ENTER ARRAY ELEMENTS FOR B:
1         1         1
1         1         1
1         1         1
Matrix addition is
2         2         2
2         2         2
2         2         2
```

WAP to perform transpose of a matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
int A[3][3],i,j,temp;
clrscr();
printf("ENTER ARRAY ELEMENTS\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&A[i][j]);
    }
}
```

```
for(i=0;i<3;i++)
{
 for(j=0;j<i;j++)
 {          temp=A[i][j];
            A[i][j]=A[j][i];
            A[j][i]=temp;
 }
}
printf("transpose of Matrix is\n\n");
for(i=0;i<3;i++)
{
 for(j=0;j<3;j++)
 {
            printf("%d\t",A[i][j]);
 }
printf("\n");
}
getch();
}
```

output

```
ENTER ARRAY ELEMENTS
1 2 3
1 2 3
1 2 3
transpose of Matrix is
1       1       1
2       2       2
3       3       3
```

WAP to perform addition of column elements of a matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
int A[3][3],i,j,sum[3]={0};
clrscr();
printf("ENTER ARRAY
    ELEMENTS\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
           scanf("%d",&A[i][j]);
    }
}
```

```
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
           sum[i]=sum[i]+A[j][i];
    }
}
printf("sum of column elements
    is\n\n");
for(i=0;i<3;i++)
{
    printf("%d\t",sum[i]);
}

getch();
}
```

output

```
ENTER ARRAY ELEMENTS
1 2 3
1 2 3
1 2 3
sum of column elements is
3       6       9
```

WAP to perform addition of diagonal elements of a matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
int A[3][3],i,j,sum=0;
clrscr();
printf("ENTER ARRAY ELEMENTS\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&A[i][j]);
    }
}
```

```
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(i==j||i+j==2)
        {
            sum=sum+A[i][j];
        }
    }
}
printf("sum of diagonal elements is\n\n");

    printf("%d\t",sum);


getch();
}
```

output

```
ENTER ARRAY ELEMENTS
1 2 3
1 2 3
1 2 3
sum of diagonal elements is
10
```

WAP to perform matrix multiplication

```
#include<stdio.h>
#include<conio.h>
void main()
{
int A[3][3],B[3][3],C[3][3],i,j,k;
clrscr();
printf("ENTER ARRAY
    ELEMENTS FOR A\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
            scanf("%d",&A[i][j]);
    }
}
printf("ENTER ARRAY
    ELEMENTS FOR B\n");
```

```
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {       scanf("%d",&B[i][j]);
    }
}
Printf(“Matrix multiplication is:\n);
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {       C[i][j]=0;
            for(k=0;k<3;k++)
            {
            C[i][j]+=A[i][k]*B[k][j];
            }
            printf("%d\t",C[i][j]);
    }
    printf("\n");
}getch();}
```

output

```
ENTER ARRAY ELEMENTS FOR A
1 1 1
1 1 1
1 1 1
ENTER ARRAY ELEMENTS FOR B
2 2 2
2 2 2
2 2 2
Matrix multiplication is:
6       6       6
6       6       6
6       6       6
```

program to cyclically rotate the elements in array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,a[10],k,n,temp,size;
char ch;
 size=5;
 clrscr();
 printf("Enter array elements:\n");
 for(i=0;i<size;i++)
 {
   scanf("%d",&a[i]);
 }
 printf("how many times to rotate the
     bit");
  scanf("%d",&n);
printf("Enter L for Left Rotation and R
     for Right Rotation:");
```

```
ch=getche();
 switch(ch)
      {
 case 'L':
 case 'l':
          for(k=0;k<n;k++)
        {
         i=0;
         temp=a[i++];
         for(j=i;j<size;j++)
         {
           a[j-1]=a[j];
          }
          a[j-1]=temp;
          }
         break;
```

```
case 'R':
case 'r':
        for(k=0;k<n;k++)
       {
       i=size-1;
       temp=a[i--];
       for(j=i;j>=0;j--)
       {
        a[j+1]=a[j];
        }
       a[0]=temp;
       }
       break;
```

```
default:
                printf("Invalid choice..........");
    }

 for(i=0;i<size;i++)
 {
    printf("%d ",a[i]);
   }

 getch();
 }
```

output

```
Enter array elements:

1 2 3 4 5

how many times to rotate the bit2

Enter L for Left Rotation and R for Right Rotation:l3 4 5 1 2

Enter array elements:
2 3 4 5 7
how many times to rotate the bit3
Enter L for Left Rotation and R for Right Rotation:R4 5 7 2 3
```

STRINGS

- Strings in C are represented by **One- dimensional Character Arrays**
- An array formed by characters is a string in C.
- The end of the string is marked with a special character, the *null character*
- The null character is represented by character escape sequence, ‘\0’.

"A String"

A		S	t	r	i	n	g	\0

Declaration of a String

- Strings can be declared like one-dimensional arrays.

```
datatype  stringname[size]
```

  - For example,

```
char str[6];
char text[10];
```

str[0]	**str[1]**	**str[2]**	**str[3]**	**str[4]**	
H	E	L	L	O	\0

Initializing Strings

- Allocate an array of a size large enough to hold the string (plus 1 extra value for the delimiter)
- Examples (with initialization):

```
char str1[6] = “Hello”;

char str2[] = “Hello”;

char str3[6] = {‘H’,’e’,’l’,’l’,’o’,’\0’};
```

STRING INPUT AND OUTPUT

INPUT

```
char c[10];
printf("ENTER STRING: ");
scanf("%s“,c);
```

OR

```
printf("ENTER STRING: ");
gets(c);
```

OUTPUT

```
printf(“\n String= %s",c);
```

OR

```
printf(“\nString=“);
puts(c);
```

PROGRAM FOR STRING INPUT AND OUTPUT using scanf() and printf()

```
#include<stdio.h>
#include<conio.h>
void main()
{
char c[10];
clrscr();
printf("ENTER STRING: ");
scanf("%s",c);
printf("\nString=%s",c);
getch();
}
```

```
OUPPUT
ENTER STRING: Hi Hello
String=Hi
```

PROGRAM FOR STRING INPUT AND OUTPUT using gets() and puts()

```
#include<stdio.h>
#include<conio.h>
void main()
{
char c[10];
clrscr();
puts("ENTER STRING: ");
gets(c);
puts(“String=");
puts(c);
getch();}
```

```
OUPPUT
ENTER STRING:
Hi Hello
String=
Hi Hello
```

Write a program to calculate length of a string

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[10];
int i=0;
clrscr();
printf("ENTER STRING: ");
gets(a);
while(a[i]!='\0')
{
    i++;
}
printf("Length of a String=%d",i);
getch();}
```

```
OUTPUT
ENTER STRING: HI HELLO
Length of a String= 8
```

Write a program to copy one string into another

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[10],b[10];
int i=0;
clrscr();
printf("ENTER STRING: ");
gets(a);
while(a[i]!='\0')
{
    b[i]=a[i];
    i++;
}
b[i]='\0';
printf(“Copied String=%s",b);
getch();
}
```

```
OUTPUT
ENTER STRING: HELLO
Copied  String= HELLO
```

Write a program to concatenate one string into another

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[20],b[10];
int i=0,j=0;
clrscr();
printf("ENTER FIRST STRING: ");
gets(a);
printf("ENTER SECOND STRING: ");
gets(b);
while(a[i]!='\0')
{
    i++;
}
while(b[j]!='\0')
{
    a[i]=b[j];
    i++;
    j++;
}
a[i]='\0';
printf("Concatenated String=%s",a);
getch();
}
```

```
OUTPUT
ENTER FIRST STRING : Hello
ENTER SECOND STRING : Hi
Concatenated String= HelloHi
```

Write a program to reverse a given string

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[10],b[10];
int i=0,j=0;
clrscr();
printf("ENTER STRING: ");
gets(a);

while(a[i]!='\0')
{
    i++;
}

i--;
while(i>=0)
{
    b[j]=a[i];
    j++;
    i--;
}
b[j]='\0';
printf("Reverse String=%s",b);
getch();
}
```

```
OUTPUT
ENTER STRING : Hello
Reverse String=olleH
```

Write a program to check given string is palindrome or not.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[10];
int i=0,j=0,k=0;
int flag=0;
clrscr();
printf("ENTER STRING: ");
gets(a);
while(a[i]!='\0')
{
    i++;
}
k=i/2;
i--;
```

```
while(j<k)
{
    if(a[j]!=a[i])
    {
    flag=1;
    break;
    }
    j++;
    i--;
}
if(flag==0)
printf("String %s  is a palindrome", a);
else
printf("String %s is not a palindrome",
    a);
getch();
}
```

OUTPUT

```
ENTER STRING : malayalam
String malayalam is a palindrome
```

Write a program to compare two strings

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[20],b[20];
int i=0, count=1;
clrscr();
printf("ENTER FIRST STRING: ");
gets(a);
printf("ENTER SECOND STRING: ");
gets(b);
while(a[i]!='\0’&&b[i]!='\0')
{       if(a[i]!=b[i])
    {
             count=0;
             break;
    }
    i++;
}
if(count==1)
printf("\nSTRINGS ARE EQUAL");
else
printf("\nSTRINGS ARE NOT EQUAL");
getch();
}
```

```
OUTPUT
ENTER FIRST STRING : Hello
ENTER SECOND STRING : Hi

STRINGS ARE NOT EQUAL
```

Built-in Functions from string.h

Function	Purpose	Example
strcpy	Makes a copy of a string	strcpy(s1,s2)
strcat	Appends a string to the end of another string	strcat(s1,s2)
strcmp	Compare two strings alphabetically	strcmp(s1, s2)
strcmpi	Compare two strings by ignoring case	strcmpi(s1, s2)
strlen	Returns the length of a string	strlen(s1)
strlwr	Converts the string to lower case	strlwr(s1)
strupr	Converts the string to upper case	strupr(s1)
strrev	Reverse the string	strrev(s1)

strcpy(arg1,arg2)

- copies the string in the second argument into the first argument.
- The **null character** is appended at the end automatically

Example

```
char s1[]=“hi”
char s2[]=“hello”
strcpy(s1,s2);
```

s1=

h	**e**	**l**	**l**	**o**	**\0**

strcat(arg1,arg2)

- concatenate the string in the second argument at the end of first argument.
- The **null character** is appended at the end automatically

Example

char s1[]=“hi”

char s2[]=“hello”

strcat(s1,s2);

s1=

h	**i**	**h**	**e**	**l**	**l**	**o**	**\0**

strlen (arg)

- returns length of a string in terms of number of characters before null character

Example

char s1[] = “Hello”

int n=strlen(s1)

s1=

h	**e**	**l**	**l**	**o**	**\0**

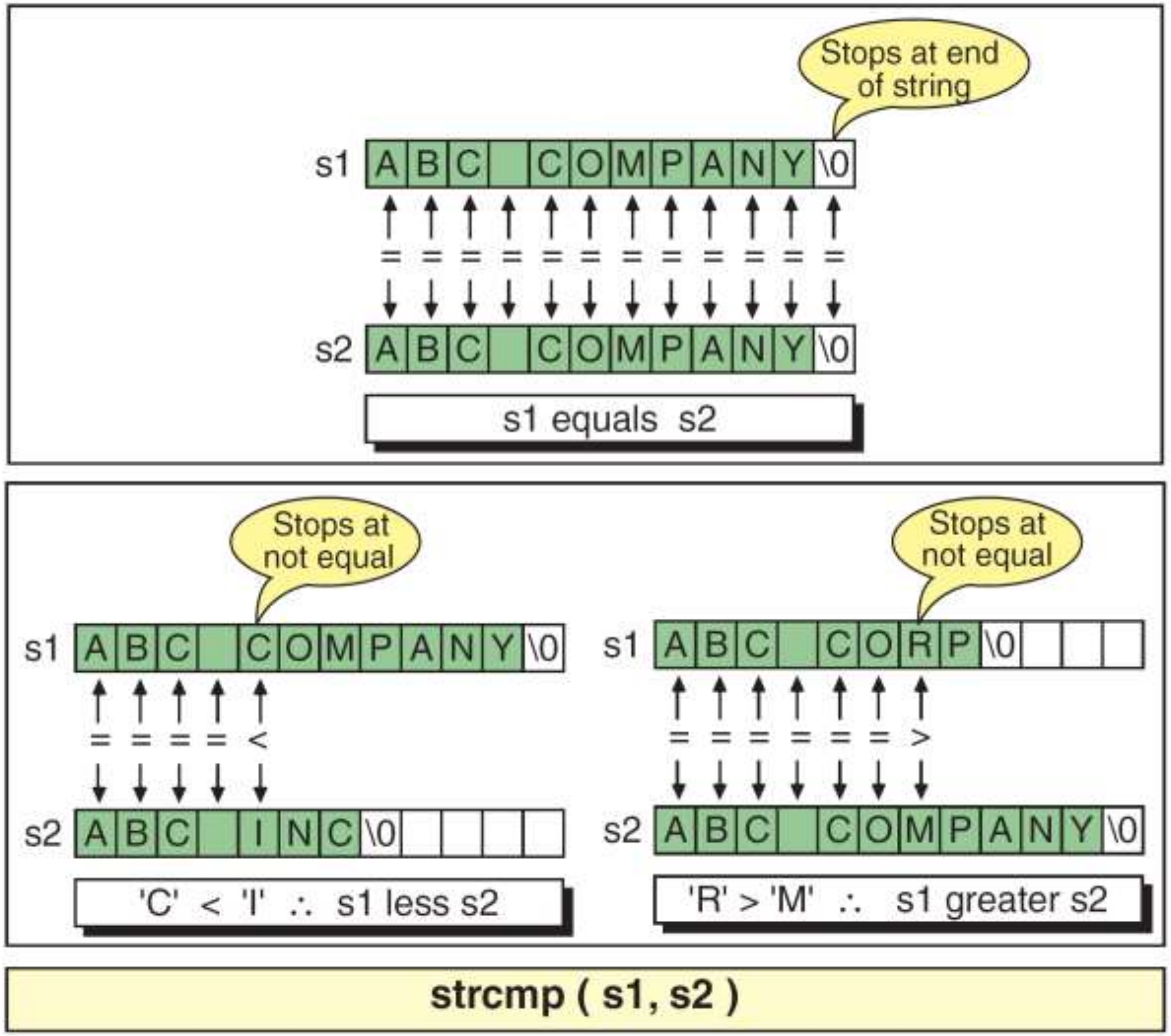
n=5

strcmp(str1, str2)

- compares two strings

Relationship	**Returned Value**	**Example**
`str1 < str2`	Negative	“Hello”< “Hi”
`str1 = str2`	0	“Hi” = “Hi”
`str1 > str2`	Positive	“Hi” > “Hello”

Comparing Strings



- strlwr(s1)

  char s1[] = “Hello”
  strlwr(s1)
  s1=hello

- strupr(s2)

  char s2[] = “hello”
  strupr(s2)
  s2=HELLO

- strrev(s3)

  char s3[] = “Hello”
  strrev(s3)
  s3=olleH

Character Manipulation in the String using functions in ctype.h

Function	Work Of Function
isalnum(c)	Tests whether a character c is alphanumeric or not
isalpha(c)	Tests whether a character is alphabetic or not
isdigit(c)	Tests whether a character is digit or not
islower(c)	Tests whether a character is lowercase or not
isspace(c)	Tests whether a character is white space or not
isupper(c)	Tests whether a character is uppercase or not
tolower(c)	Converts to lowercase if the character is in uppercase
toupper(c)	Converts to uppercase if the character is in lowercase

Write a program to count number of vowels,digits,spaces,consonants & alphabates in a given string

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char a[20];
int i=0,l,cons,alpha,space=0,vowel=0,digit=0;
clrscr();
printf("ENTER STRING WITHOUT
    SPECIAL SYMBOLS: ");
gets(a);
strlwr(a);
puts(a);
l=strlen(a);
```

```
while(a[i]!='\0')
{
    if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u'
    )
    {
            vowel++;  }
    if(a[i]<='9'&&a[i]>='0')
    {
            digit++;    }
    if(a[i]==' ')
    {                   space++;   }
    i++;
}
cons=l-digit-vowel-space;
alpha=cons+vowel;
```

```
printf("\nNUMBER OF VOWELS= %d",vowel);
printf("\nNUMBER OF ALPHABATES= %d“,alpha);
printf("\nNUMBER OF CONSONANTS= %d",cons);
printf("\nNUMBER OF DIGITS= %d",digit);
printf("\nNUMBER OF SPACES= %d",space);
getch();}
```

OUTPUT

```
ENTER STRING WITHOUT SPECIAL SYMBOL:
Hi 123 how r u
hi 123 how r u
NUMBER OF VOWELS=3
NUMBER OF ALPHABATES=7
NUMBER OF CONSONANTS=4
NUMBER OF DIGITS=3
NUMBER OF SPACES=4
```

Write a program to count number of occurences of a character in string

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch,str[20];
int i=0,count=0;
clrscr();
printf("ENTER STRING : ");
gets(str);
printf("ENTER CHARACTER TO COUNT OCCURENCES : ");
scanf("%c",&ch);

while(str[i]!='\0')
{
	if(str[i]==ch)
	{
	count++;
	}
	i++;
}
printf("%c occurs %d times",ch,count);

getch();
}
```

```
OUTPUT
ENTER STRING : Hello
ENTER CHARACTER TO
COUNT OCCURENCES : l


l occurs 2 times
```

TWO-DIMENSIONAL CHARACTER ARRAY

- A two-dimensional array of strings can be declared as follows:
  - data_type string_array_name [size1] [size2];
- Example

```
char s1[5][30];
char s2[5][10];
```

Initialization

- char s[5][10] ={“hello”,”hi”,”C”,”Good”,”Morning”};

s[0]	h	e	l	l	o	\0				
s[1]	h	i	\0							
s[2]	C	\0								
s[3]	G	o	o	d	\0					
s[4]	M	o	r	n	i	n	g	\0		

- Here every row is a string. That is, s[i] is a string.

PROGRAM FOR INPUT AND OUTPUT

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[5][10];
int i,j;
clrscr();
puts("Enter String :");
for(i=0;i<5;i++)
{
gets(s[i]);
}
for(i=0;i<5;i++)
{
printf(“s[%d]= “,i);
puts(s[i]);
}
getch();
}
```

```
OUPPUT
Enter Strings :
Hello
Good
Morning
Hi
SPA
```

```
s[0]=Hello
s[1]=Good
s[2]=Morning
s[3]=Hi
s[4]=SPA
```

Write a program to sort strings in ascending order

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char str[5][10]={"Mumbai","Delhi",“Calcutta","Chennai","Banglore"};
int i=0,j=0;
char temp[10];
clrscr();
for(i=0;i<5;i++)
{
 for(j=i+1;j<5;j++)
 {
  if((strcmp(str[i],str[j]))>0)
  {
     strcpy(temp,str[i]);
    strcpy(str[i],str[j]);
    strcpy(str[j],temp);
  }
 }
 }
```

```
 i=0;
 printf("Sorted Strings : “);
 while(i<5)
 {
printf("\n%s",str[i]);
i++;
}
getch();
}
```

OUTPUT

```
Sorted Strings


Banglore
Calcutta
Chenai
Delhi
Mumbai
```