Array

Definition of an array

An array is a collection of homogenous data elements of a specific data type (*i.e.* int or float or char) that have been given a single name, which are stored at contiguous memory locations (*i.e.* one after the other). Each individual element of an array is referenced by a subscripted variable. The size of the array is enclosed in brackets after array name.

$$\begin{tikzcd} \textit{e.g.}\ \text{int} \arrow[d] & s \arrow[d] & {[40];} \arrow[d] \\ \text{data type} & \text{name of the array} & \text{size} \end{tikzcd}$$

If one subscript is used to refer to an element of an array the array is known as one-dimensional or linear array. If two subscripts are used to refer to an element of an array the array is known as two-dimensional array or matrix and so on. The arrays, which has two or more subscripts are known as multidimensional array. For each subscript, we use a pair of brackets.

$$\begin{tikzcd} \textit{e.g.}\ \text{int}\ a & {[5]} \arrow[d] & {[6];\ \text{/* two dimensional array because of two subscripts */}} \arrow[d] \\ & \text{subscript 1} & \text{subscript2} \end{tikzcd}$$

Array Declaration

It includes
1. Data type of an array
2. The name of the array (rules are same as variable name)
3. The number of subscripts in an array
4. The maximum value of each subscript

Syntax :

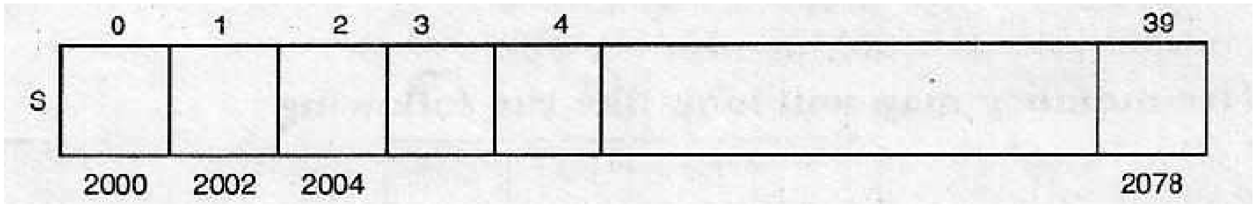
type name [subl] [sub2] ....;

where type is the data type (int, char or float)
name is the name of the array
subl, sub2, ... are number of subscripts available in an array which are used to specify the dimension of an array.

e.g. int s [40]; /* one subscript so one-dim array */

where int specifies the type of the variable, s is the name of the array variable, [40] is the subscript to specify that there are maximum 40 elements stored in this array.

The elements in an array start with 0 not one and ends with one less than value of subscript(n-1).
Since one integer value requires 2 bytes of memory, so for array compiler allocates a contiguous block of 80 bytes in memory.



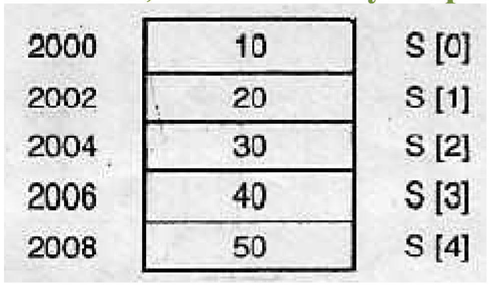
2000, 2002, ... are the addresses of memory location.

Memory Map of an array

Memory map of an array can be defined as a map by which an array can be represented inside the memory.

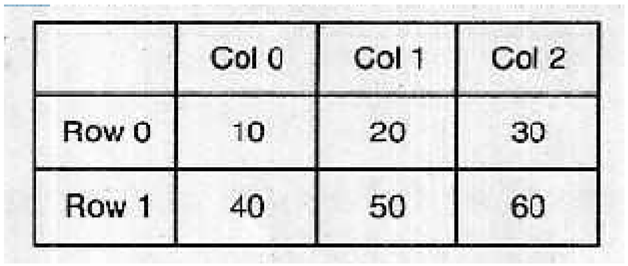
One dimensional array

int S[5] = {10, 20, 30, 40, 50};
Since we know that array elements are stored at contiguous memory locations, so its memory map is



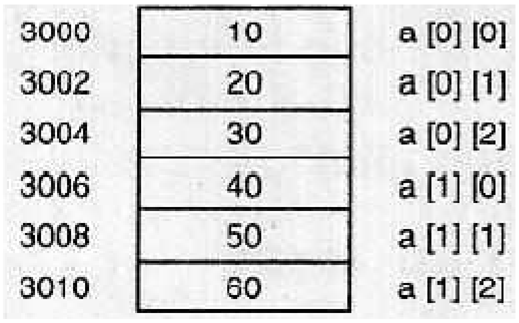
Two dimensional array

Let us suppose that we have two dimensional array declaration
int a[2][3] = {10, 20, 30, 40, 50, 60};
Its memory map is



	Col 0	Col 1	Col 2
Row 0	10	20	30
Row 1	40	50	60

memory map in actual memory will be :



Entering data

Array is a collection of values. So we can enter data into array by using loop statement. For data entry into one dimensional array, one loop is used, for two dimensional array two loops are used and so on..

One dimensional array

e.g. /* demonstration of data entry into an one dimensional array */

```
int s[10],i;
for (i = 0; i <= 9; i ++)
{
   printf ("\n Enter marks");
   scanf ("%d", &s[i]);
}
```

Two dimensional array

/* demonstration of data entry into two dimensional array */

```
int a [5][6], i, j;
for (i = 0; i < = 4; i ++ )
{
  for (j = 0; j < = 5;j + +)
  {
     printf ("\n Enter data");
     scanf ("%d", &a [i] [j]);
  }
}
```

Traversing or outputting an array

Data can be traversed by using loop statements. To traverse each element of an array, we use a variable in the subscript.

One dimensional array

e.g. /* Demonstration of outputting one dimensional array */

```
    printf ("\n Elements of an array are \n");
    for (i = 0; i < = 9; i + +)
         printf ("%d \t", s[i]);
```

Two dimensional array

```
/* demonstration of outputting two dimensions array */
printf ("\n Elements of an array are \n");
for (i = 0; i < = 4; i + +)
{
   for(j = 0; < = 5; j + +)
         print ("%d \t", a[i][j]);
  printf ("\n");
}
```

Initializing or assignment of an array

When we declare an array, the elements of the array has values according to the storage class of the array *i.e.* 0 value for static and extern and garbage value for auto and register.

***e.g.* int S [10]; /* An array S of 10 elements with initial value as garbage value */**

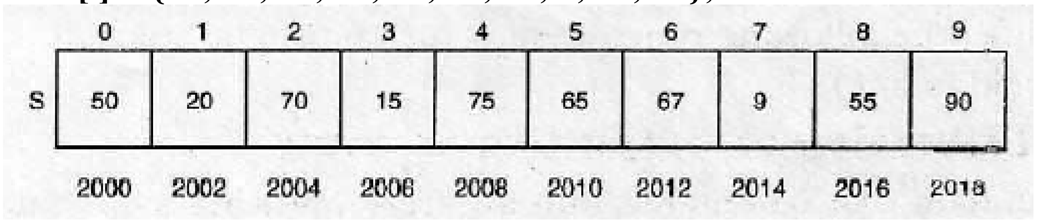
Initializing one dimensional array

int S[10] = {50, 20, 70, 15, 75, 65, 67, 9, 55, 90};

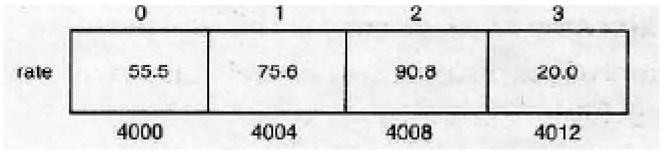
S [0] = 50, S [1] = 20, ... , S [9] = 90

Specifying subscript value is optional

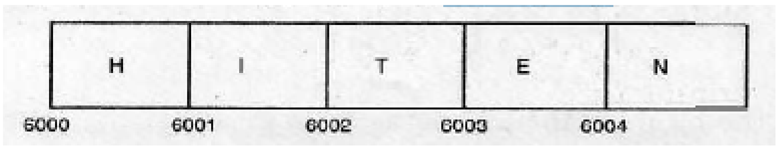
int S[ ] = {50, 20, 70, 15, 75, 65, 67, 9, 55, 90};



float rate [ ] = {55.5, 75.6, 90.8, 20.0};



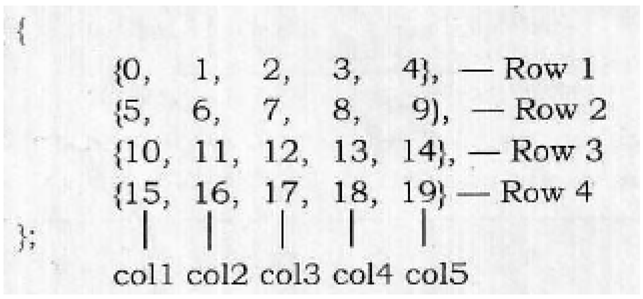
char name []= {'H', 'I', 'T', 'E', 'N'};



Initializing two dimensional array

It is initialized as

int a [4][5]



```
{
        {0,   1,   2,   3,   4}, — Row 1
        {5,   6,   7,   8,   9), — Row 2
        {10,  11,  12,  13,  14}, — Row 3
        {15,  16,  17,  18,  19} — Row 4
         |    |    |    |    |
};
        col1 col2 col3 col4 col5
```

where a is the name of the 2 dim. array of int type

4 is the first subscript *i.e.* number of rows

5 is the second subscript *i.e.* number of columns

It is treated as an array of one dimensional arrays.

int a [4] [5] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19};

Write a program which reads a list of ten numbers and prints the list in reverse order.

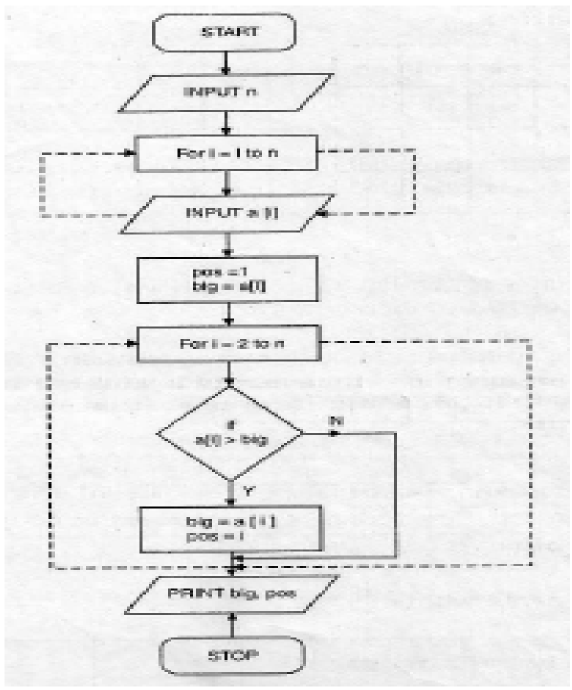
```
/* Pnnt an array in reverse order */
#include <stdio.h>
#include <conio.h>
void main( )
{
   int n[10], i;
   clrscr( );
   printf ("\n Enter the list of 10 numbers");
   for (i = 0; i < = 9; i + +)
     scanf ("%d", &n[i]);
    printf ("\n The list in reverse order is \n");
   for (i = 9; i > = 0; i - -)
     printf ("%d \t", n[i]);
 getch( );
}
```

Output.

```
Enter the list of 10 numbers
10 20 30 40 50 60 70 80 90 100
The list in reverse order is
100 90 80 70 .60 50 40 30 20 10
```

Write a program which finds the largest number and its position in a list of n numbers.

```
/* Find the largest number and its position in a list of n numbers */
#includc <stdio.h>
#include <conio.h>
void main( )
{
    int a[ 100], big, pos, i, n ;
    clrscr( );
    printf ("\n Enter the size of the list");
    scanf("%d", &n);
    printf ("\n Enter list of %d numbers", n);
    for (i = 0; i < n; i + + )
      scanf ("%d", &a[i]);
    big = 0; /* set the first value to big */
   pos = 0; /* set the position of big as 0 */
   for (i = 0; i < n; i + +)
   {
    if (a[i] > big)              10,2,60,46,20
    {
       big = a[i];
       pos = i;
    }
```



```
  }
  pos ++; /* Increase position by 1 as array is counted from 0 */
  printf ("\n The largest number = %d is stored at position = %d", big, pos);
getch( );
```

Output.

```
Enter the size of the list 10
Enter list of 10 numbers
2 8 1 9 50 40 30 7 20 10
The largest number = 50 is stored at position = 5
```

An array of 10 elements is entered from keyboard. Write a program to interchange its even position elements with odd position elements.

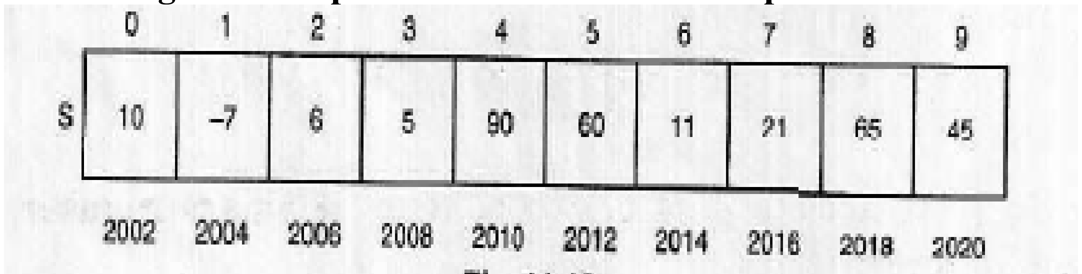
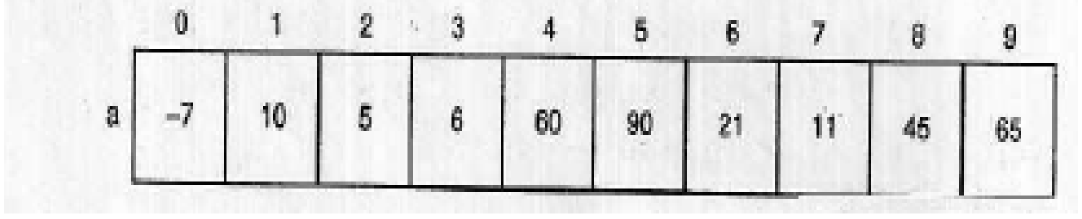


Fig. 11.12

Resulting array will be



```
#include < stdio. h>
#include <conio.h>
void main( )
{
   int a [10], i,t;
   clrscr( );
   printf ("\n Enter a list of 10 elements");
  for (i = 0; i < =9; i + +)
    scanf ("%d", &a[i]);
  for (i = 0;i < 10; i + = 2) /* Interchange even elements with odd elements
position wise */
  {
     t = a [i]; a[0]
    a[i] = a[i + 1];   a[0]=a[1]
    a[i + 1] = t;
  }
  printf ("\n The resultant list is ... \n");
  for (i = 0; i < 10; i ++)
     printf ("%d \t", a[i]);
  getch( );
}
Output.
Enter a list of 10 elements
10 -7 6 5 90 60 11 21 65 45
The Resultant list is ...
-7 10 5 6 60 90 21 11 45 65
```

Write a program to convert a binary number to a decimal number.

e.g $(110101)_2 = 1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1\ 1x2^0$
$= 32 + 16 + 0 + 4 + 1 =(53)10$

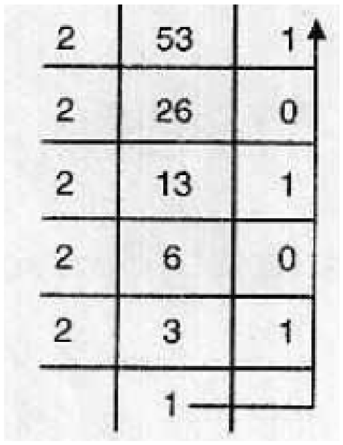
```
/* Convert binary to decimal numbers */
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main( )
{
  int i, r, count = 0;
  long b, n = 0,b1;
  clrscr( );
  printf ("\n Enter a binary number");
  scanf ("%ld", &b);
  bl = b;
  while (b > 0)       110101                r       n       b              n
  {                                                 1       0       110101
    r = b% 10;                              0       1       1101  1
    n = n + r * pow(2, count);  1           1
    count = count + 1
    b= b/10;
  }
  printf ("\n Decimal value of binary %ld = %ld", b1,, n);
  getch( );
}
Output.
Enter a binary number 110101
Decimal value of binary 110101 = 53
```

Write a program to convert a decimal number into binary number.
e.g. $(53)_{10} = (x)_2 = (110101)_2$

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

```
void main( )
{
  long d;
  int i = 0, a [25], j;
  clrscr( );
  printf ("\n Enter a decimal number");
  scanf ("%ld", &d);
  while (d > 0)
  {
    a[i] = d %2;
   d = d/2

   i = i + 1
  }
  printf ("\n The binary number is ... \”);
  for (j = 10; j > = 0; j - -)
     printf ("%d", a [j]);
  getch( );
}
```



Output.

```
Enter a decimal number 53
The binary number is ... 110101
```

Write a program to search a given number from a list of n numbers. Also print its location.

```
/* Search a number from n numbers */
#include <stdio.h>
#include <conio.h>
void main())
{
   int a [50], i, s, n, loc ;
   clrscr( );
   printf ("\n Enter number of elements in the list");
   scanf ("%d", &n);
   printf ("\n Enter %d elements from keyboard", n);
   for (i = 0; i < n; i ++)
       scanf ("%d", &a[i]);
   printf ("\n Enter element to search");
   scanf ("%d", &s);     40
   for (i = 0; i < n; i + +)
   {
     if (s == a [i])   40==40
     {
       loc = i + 1;
       printf ("\n Element %d is present at %d location", s, loc);
       break;
```

```
      }
   }
   if (i == n)
       printf (An Element %d is not present in the list", s);
   getch( );
}
```

Output.

```
Enter number of elements in the list 10
Enter 10 elements from keyboard
5   10   20   1   11   -15   6   2   100   7
Enter element to search 2
Element 2 is present at 8 location
```

Write a program to enter data into two dimensional array. Also print the data in matrix form.

```
/* Enter and print a matrix called .as traversal of two dim. array *1
#include <stdio.h>
#include <conio.h>
void main( )
{
   int a[10][10], i, j, r, c;
   clrscr( );
   printf ("\n Enter number of rows and column of a matrix");
   scanf ("%d %d", &r, &c);
   printf ("\n Enter elements of %d rows and %d column \n", r, c);
   for (i = 0; < r; ++)   //to read
   {
       for (j = 0; j < c; j ++)
         scanf (" %d", & a[i][j]);
   }
  for (i = 0; i < =r; i + +)   4   // to print
  {
    for (j = 0; j <= c; j + +)   3
       printf ("%d \t", a[i][j]);
    printf ("\n"); /* to print new line after row completion*/
  }
  getch( );
}
rc
00      01      02
10      11      12
20      21      22
30      31      32     4x3
```

Output.

```
Enter number of rows and columns of a matrix 3 4
Enter elements of 3 rows and 4 columns
2 4 6 8 10 12 14 16 18 20 22 24
Elements in matrix from are ...
        2       4       6       8
        10      12      14      16
        18      20      22      24
```

Write a program to find the transpose of a square matrix.

Transpose of a matrix can be found out if any only if matrix is square *i.e.* number of its rows and columns are same. Transpose of a matrix can be obtained, if we interchange the rows and columns of the matrix. *e.g.*

```
                if a=
                1       2       3
                4       5       6
                7       8       9
                Transpose of a =
                1       4       7
                2       5       8
                3       6       9
```

```
#include <stdio.h>
#include<conio.h>
void main()
{
   static int array[10][10];
   int i, j, m, n;
   printf("Enter the order of the matrix \n");
   scanf("%d %d", &m, &n);   3,3
   printf("Enter the coefficients of the matrix\n");
   for (i = 0; i < m; ++i)   //to read
   {
      for (j = 0; j < n; ++j)
        scanf("%d", &array[i][j]);
    }
   printf("The given matrix is \n");
   for (i = 0; i < m; ++i)
   {
      for (j = 0; j < n; ++j)
        printf(" %d", array[i][j]);
       printf("\n");
   }
   printf("Transpose of matrix is \n");
   for (j = 0; j < n; ++j)
   {
        for (i = 0; i < m; ++i)
            printf(" %d", array[i][j]);
         printf("\n");
    }
}
```

OUTPUT

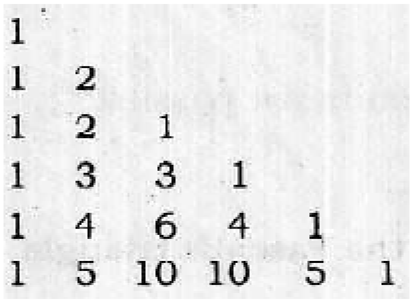
```
Enter the order of the matrix
3        3
Enter the coefficients of the matrix
3       7       9
2       7       5
6       3       4
The given matrix is
 3      7       9
 2      7       5
 6      3       4
Transpose of matrix is
 3      2       6
 7      7       3
 9      5       4
```

Write a program to print the Pascal's triangle

```
/* Pascal's triangle */
#include <stdio.h>
#include <conio.h>
void main( )
{
  int a[101[10], i, j, n;
  clrscr( );
  printf ("\n How many rows");
  scanf ("%d", &n);
  for (i = 1; i < ; + +)
  {
   for (j = 1; j < = i; + +)
   {
     if(j==1 ||  j==i)
       a [i][j] = 1;
    else
      a[i][j] = a [i - 1] [j] + a [i - 1] [j - 1);
    printf ("%d \t", .a [i][j]);
  }
  printf ("\n");
 }
 getch( );
}
```

Output.

How many rows 6



```
1
1  2
1  2   1
1  3   3   1
1  4   6   4   1
1  5  10  10   5   1
```

Pass individual elements of an array to a function one by one i.e. (call by value).

```
#include <stdio.h>
#include <conio.h>
void main( )
{
   int m[] = {50, 60, 70, 40, 90, 80}, i;
   clrscr( );
   for (i = 0; i < 6; i + +)
      show (m [i]);
   getch( );
}
show (int marks)
{
  printf ("\t %d", marks);

}
```

Output.

```
50      60      70      40      90      80
```

Pass individual elements of an array to a function one by one i.e. (call by reference)

```
void main( )
{
  int m[ ] = {50, 60, 70, 40, 90, 80},i;
  clrscr( );
  for (i = 0; i < 6; i + +)
     show (m + i);
 }
 show (int *marks)
 {
    printf ("\t %d", *marks);
 }
```