Arrays

Gagan Deep
Founder & Director
Rozy Computech Services
Kurukshetra, rozygag@yahoo.com
Visit us at : www.rozyph.com

Arrays

- Array is a linear list of homogenous elements, stored at successive memory locations in consecutive order.
- C programming language provides a data structure called **the array**, that can store a fixed size sequential collection of elements of same data type.
- An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- For example, if your class strength is 50 then your roll numbers are rollnumber1, rollnumber2, rollnumber3 …… rollnumber50.
- Instead of declaring individual variables, such as rollnumber1, …, and rollnumber50, you declare one array variable such as rollnumb.

- But in C, array always starts from 0 and end with size-1 and be written as

  rollnumb[0], rollnumb[1]……….rollnumb[49]

- and be declared as

  int rollnumb[50];

- means declaration syntax is

  datatype arrayname [size];

- Types of array

  One-Dimensional

  Multi-Dimensional

One Dimensional Arrays

- Array having only one subscript variable is called one-dimensional array and also called as single dimensional array or linear array.
- Example of one dimensional array definition

```
int marks[5] = {50, 85, 60, 55, 90};
```

- And represented as

```
marks[0] =50
marks[1] = 85
marks[2] = 60
marks[3] = 55
marks[4] = 90
```

- A specific element in an array is accessed by an index.

Accessing One Dimensional Arrays

Program 1 : Input and output of an Array

```
#include <stdio.h>
void main()
{ int marks[5],i;
printf(“Input of Array marks”);
for (i=0; i<5; i++)
scanf(“%d”, &marks[i]);
    printf(“Output of Array marks”);
    for (i=0; i<5; i++)
    printf(“marks[%d]  =  %d    ”,  i,
            marks[i]);
}
```

Input of Array marks

(if you entered)

```
50
85
60
55
90
```

(you will get)

Output of Array marks

```
marks[0] =50
marks[1] = 85
marks[2] = 60
marks[3] = 55
marks[4] = 90
```

Program 2 : Find the largest in an array.

```
#include<stdio.h>
void main() {  int a[20], i, n, largest;
  printf("\nEnter no of elements :");
  scanf("%d", &n);
  for (i = 0; i < n; i++)  // Input Array
    scanf("%d", &a[i]);

  largest = a[0]; //Consider first element as largest

  for (i = 1; i < n; i++)  //Process
    if (a[i] > largest)
        largest = a[i];
  printf("\nLargest Element : %d", largest);   }//output largest
```

Program 3 : Linear Search unsorted list

```
#include<stdio.h>
void main() {  int a[20], i, n, item, loc=-1;
  printf("\nEnter no of elements :");   scanf("%d", &n);
  for (i = 0; i < n; i++)                 // Input Array
    scanf("%d", &a[i]);
  printf("\nEnter the item to be search:");  scanf("%d", &item);

  for (i = 0; i < n; i++)                 //Process
    if (a[i] == item)
        {  loc= i;            break;  }
                 if (loc>=0)        //output largest
                   printf("\nItem Found: %d", loc+1);
                         else  printf("\nItem Not Found: %d"   }
```

Program 4 : Linear Search Sorted list

```
#include<stdio.h>
void main() {  int a[20], i, n, item, loc=-1;
   printf("\nEnter no of elements :");   scanf("%d", &n);
   for (i = 0; i < n; i++)        scanf("%d", &a[i]);      // Input Array
     printf("\nEnter the item to be search:");  scanf("%d", &item);

   for (i = 0; i < n; i++)                //Process
      if (a[i] == item)    {  loc= i;             break;  }
       else if (a[i]>item)    break;


                        if (loc>=0)        //output location if item found
                          printf("\nItem Found: %d", loc+1);
                                 else  printf("\nItem Not Found: )   }
```

Program 5 : Binary Search Sorted list

```
#include<stdio.h>
void main() {  int a[20], i, n, item, b=0, loc=-1, e, m;
    printf("Enter the size of an array: ");  scanf("%d",&n);
    printf("Enter the elements in ascending order: ");
    for(i=0;i<n;i++)        scanf("%d",&a[i]);   //Input Sorted Array
    printf("Enter the item to be search: ");    scanf("%d",&item);
     e=n-1;
    while(b<=e){ m =(b+e)/2;   //process
         if(item==a[m]) {   loc=m;      break;      }
       else if(item<a[m]){  e=m-1;  }
         else     b=m+1;    }
    if(loc>=0)  //Output
         printf("The item is found at location  %d“, loc+1);
    else          printf("The item is not found."); }
```

Program 6 : Insertion of Item in an Array

```
#include<stdio.h>
void main() {  int a[20], i, n, j, k, item, loc;
   printf("Enter the size of an array: ");  scanf("%d",&n);
   for(i=0;i<n;i++)       scanf("%d",&a[i]);   //Input Array
   printf("Enter the item to be insert: ");    scanf("%d",&item);
   printf(“At what location: ");    scanf("%d",&loc);
         j=n-1;   k=loc-1;
   while(j>=k){    //process
         a[j+1]=a[j] ;
         j=j-1;      }
         a[k]=item;   n=n+1;
   for(i=0;i<n;i++)
     printf("%d",a[i]);   //Output Array
```

Program 7 : Deletion of Item from an Array

```
#include<stdio.h>
void main() {  int a[20], i, n, j, item, loc;
    printf("Enter the size of an array: ");  scanf("%d",&n);
    for(i=0;i<n;i++)        scanf("%d",&a[i]);   //Input Array
    printf("Enter the location from item to be delete: ");
     scanf("%d",&loc);
      j=loc-1;  item=a[j];
    while(j < n-1){     //process
            a[j]=a[j+1] ;
            j=j+1;      }
            n=n-1;
    for(i=0;i<n;i++)
      printf("%d",a[i]);   //Output Array
```

Program 8 : Bubble Sort

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

     for (p=0; p<n-1;p++)   //Bubble Sort Process - Ascending
            for(c=0; c< n-1-p; c++)
                        if a[c] > a[c+1]
                        {t=a[c]; a[c]=a[c+1]; a[c+1]=t;} //swapping

    for(i=0;i<n;i++)
      printf("%d",a[i]);   //Output Array
```

Do Yourself

Understand Algorithms and do the programming of Following

1. Merging
2. Insertion Sort
3. Selection Sort
4. Merge Sort
5. Radix Sort
6. Second Largest Element in the List
7. Delete the item from the list
8. Prime Numbers using Sieve’s Algorithm

MultiDimensional Array

- Multidimensional arrays are defined in much the same manner as one dimensional arrays, except that a separate square brackets required in each subscript.
- Thus, a two dimensional array will require two pairs of square bracket,

  a_{ij} -> a[i][j] -> a[rows][coulmn]
- a three dimensional array will require three pairs of square brackets, and so on

  a_{ijk} -> a[i][j][k] -> a[page][rows][coulmn]

MultiDimensional Array

- In general terms, a multidimensional array definition can be written as

Storage –class data-type array[exp 1][exp 2]………[exp n]

- Where storage-class refers to the storage class of the array,
- Where data-type is the data type
- array is the array name
- exp 1, exp 2 and exp n are positive valued integer expression that indicate the number of array elements associated with each subscripts.

2 Dimensional Array

- 2 dimensional array can be expressed as

	col1	col2	col3				coln
Row 1	**X[0][0]**	**X[0][1]**					**X[0][n-1]**
Row 2	X[1][0]	X[1][1]					X[1][n-1]
Row 3							
Row n	X[n-1][0]	X[n-1][1]					X[n-1][n-1]

2 Dimensional Array

- Consider the following two dimensional array definition

```
int x[3][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

x[0][0]=1	x[0][1]=2	x[0][2]=3	x[0][3]=4
x[1][0]=5	x[1][1]=6	x[1][2]=7	x[1][3]=8
x[2][0]=9	x[2][1]=10	x[2][2]=11	x[2][3]=12

Another ways of the 2-dimensional array deinition is

```
int x[3][4] = {
                 {1, 2, 3, 4},
                 { 5, 6, 7, 8},
                 {9, 10, 11, 12}
                 };
```

2 Dimensional Array

Another way, we a define

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

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

The inputs are same as in previous example

Another way, we a define as below

```
    int x[3][4] = {   {1, 2, 3},
                      { 5, 6, 7},
                      {9, 10, 11}   };Fourth Column value is 0
```

2 Dimensional Array

Another way, if we define 2-dimensional array as

```
int x[3][4] = {
                {1, 2, 3, 4, 5},
                { 6, 7, 8, 9, 10},
                {11, 12, 13, 14, 15}
                };
```

This will result in a compilation error, since the number of values in each inner pair of braces are 5, which exceeds the defined array size i.e. 4.

If you declare like this

int x[3][4] ;   Now you can input the values by scanf.

3-Dimensional Array

- Consider the following 3-dimensional array definition

```
int x[3][3][4] = {
                        {
                        {1, 2, 3,4},
                        { 5, 6, 7,8},
                        {9, 10, 11,12}   },
                        {
                         {21, 22, 23, 24},
                        {25, 26, 27, 28},
                        {29, 30, 31, 32}}
                 };
```

3-Dimensional Array

- Think of this array is collection of 3 tables, each having 3 rows and 4 columns means 3x3x4 = 36 total values. The groups of initial values will result in the assignment of the following nonzero values in the first two tables.

x[0][0][0]=1	x[0][0][1]=2	x[0][0][2]=3	x[0][0][3]=4
x[0][1][0]=5	x[0][1][1]=6	x[0][1][2]=7	x[0][1][3]=8
x[0][2][0]=9	x[0][2][1]=10	x[0][2][2]=11	x[0][2][3]=12
x[1][0][0]=21	x[1][0][1]=22	x[1][0][2]=23	x[1][0][3]=24
x[1][1][0]=25	x[1][1][1]=26	x[1][1][2]=27	x[1][1][3]=28
x[1][2][0]=29	x[1][2][1]=30	x[1][2][2]=31	x[1][2][3]=32

All the remaining elements will be assigned to zero

Addition of two matrices Programs 9

```
#include <stdio.h>
void main()
{ float a[3][3], b[3][3], c[3][3];
int i,j;
printf("Enter the elements of 1st matrix\n");
for(i=0;i<3;++i)
for(j=0;j<3;++j)
scanf("%f",&a[i][j]);

printf("Enter the elements of 2nd matrix\n");
for(i=0;i<3;++i)
for(j=0;j<3;++j)
 scanf("%f",&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("\nSum Of Matrix:");

for(i=0;i<3;++i)
{
for(j=0;j<3;++j)
printf("%.1f\t",c[i][j]);
printf("\n"); }
}
```

Thanks !

Mail at : rozygag@yahoo.com

Visit us at : www.rozyph.com