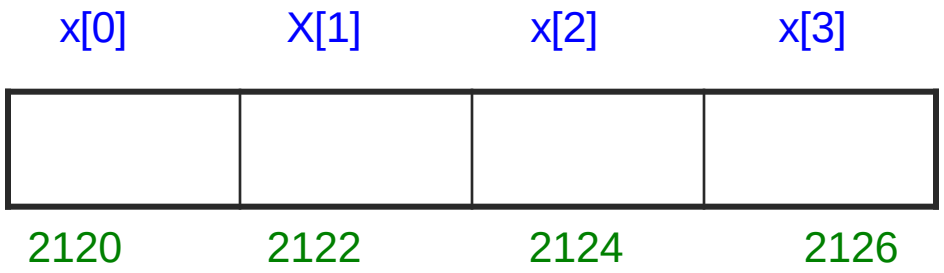
Paper 102: Programming & Problem solving through C

Lecture-13:Unit-II
Arrays

Arrays

- Most application programs require the processing of multiple data items that have a common characteristics like marks of students in a subject, a set of numerical data (x1,x2,x3….).
- In such situations, it would be more convenient if these data items are stored into an array.
- Accessing can be done by using the same name
- The individual data items can be characters, integers, floating point numbers, etc.
- However, the data items must be all of the same data type and same storage class.

- Each individual data item are referred as an array element
- They are stored consecutively in the memory
- Each array element can be accessed by specifying the name of the array followed by one or more subscripts, where each subscripts are enclosed in square brackets '[ ]'
- Each subscript must be a non-negative integer constant/integer variable/integer expression.



x is a 4-element, one dimensional array

Defining an Array

- Arrays are define using the following syntax:

  Storage-class data-type array-name[expression];

- Expression is a +ve valued integer expression

  e.g int x[4]; or auto int x[4];

  char c[10];

  static float n[20];

- Array size can make use of symbolic constant

```
#define SIZE 80
 void main()
{
      char  letter[SIZE];
}
```

Array initialization

- Array can also be initialized with values

  Storage-class data-type array-name[expression]={value 1, value 2, …..,value n};

  e.g

  int digits[10]={1,2,3,4,5,6,7,8,9,0};

  char color[3]={'R', 'E', 'D'};

color[0]	'R'
color[1]	'E'
color[2]	'D'

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

char color[3]="RED";

It defines a three element character array whose individual elements are

color[0]='R'

color[1]='E'

color[2]='D'

char color[]="RED";

It defines a four element character array whose individual elements are

color[0]='R'

color[1]='E'

coloe[2]='D'

color[3]='\0'

'\0' is added automatically at the end of the array, referred as 'end of string indicator'

Processing an Array

- Array is processed element by element, i.e., each individual item at a time
- Array are usually processed using a loop
- The number of iterations of the loop will often be equal to the number of elements of the array
- Each array element can be accessed by
  - List[0]

Values can be assigned directly

List[0]=10

List[1]=12

Or values can be input at execution time/ run time

In C, array elements start with index 0(zero) or subscript 0

The following program shows how to initialize all the elements of an integer based array to the value 10, using a for loop to cycle through each element in turn.

```
#include <stdio.h>
 void main()
{           int count;
            int  values[100];
            for( count = 0; count < 100; count++ )
                        values[count] = 10;
}
```

Or values can be entered by a user at run time

```
#include <stdio.h>
 void main()
{           int count;
            int  values[100];
            for( count = 0; count < 100; count++ )
                        scanf("%d",&values[count]);
}
```

```
/*To calculate the average of n numbers, then compute the deviation of
each number about the average*/
#include<stdio.h>
void main()
{
   int n, j;
   int list[100];
   float avg,d, sum=0;
   printf("\n enter how many values to input?");
   scanf("%d",&n);
   /*read the values from user*/
   for(j=0;j<n;j++)
   {
       scanf("%d",&list[j]);
       sum=sum+list[j];
    }
   avg=sum/n;
   printf("\n The average is %f",avg);
   /* calculate and print the deviations about the average*/
   for(j=0;j<n;j++)
   {
      d=list[j] – avg;
      printf("\n i=%d          value=%d          devation=%f",j+1,list[j],d);
   }
}
```

Passing an array to a function
(passed by reference)

- An array name can be used as an argument to a function (as an actual argument)
  - Permitting the entire array to be passed to the function
- On calling a function and passing the array, the array name must appear by itself, without any brackets or subscripts

  ```
  int a[10];
  Sum=sumarray(a,10);
  ```
- The corresponding formal arguments is written as

  ```
  int sumarray(int n[],int size) { }
  ```

Cont…

- When an array is passed to a function, the array name is interpreted as the address of the first array element (i.e., the memory location containing the first array element)
- This address is assigned to the corresponding formal argument when the function is called.
- The formal argument becomes a pointer to the first array element and all elements are accessed the same way as before
- When referring to an array element within the function, the value of the element's subscripts is added to the value of the pointer to indicate the address of the specified element

```
#include<stdio.h>
int modify(int n[],int s);      /* square up each element and sum the elements*/
void main()
{
        int a[10],i,sum,size;
        printf("\nenter how many elements to enter[max 10]:");
        scanf("%d",&size);
        printf("\n enter data into the array:");
        for(i=0;i<size;i++)
        {
                printf("\n enter element %d :",i);
                scanf("%d",&a[i]);
        }
        printf("\n calling the function\n");
        sum=modify(a,size);
        printf("\n Back to main\n");
        printf("\n displaying the array\n");
        for(i=0;i<size;i++)
                printf("\n element %d=%d",i,a[i]);
        printf("\n The sum of all the array elements is=%d",sum);
}
int modify(int n[],int s)
{
        int i,sum=0;
        for(i=0;i<s;i++)
        {
          n[i]=n[i] * n[i];
          sum=sum+n[i];
        }
        return(sum);
}
```

```
enter how many elements to enter[max 10]:5

 enter data into the array:
 enter element 0 :1

 enter element 1 :2

 enter element 2 :3

 enter element 3 :4

 enter element 4 :5

 calling the function

 Back to main

 displaying the array

 element 0=1
 element 1=4
 element 2=9
 element 3=16
 element 4=25
 The sum of all the array elements is=55
```

Class assignment

- Wap to input a list of numbers and find the biggest of these numbers in the list.
- Wap to input a list of numbers and arrange the numbers in ascending order.
- Wap to input a list of alphabets into an array of character and count the number of vowels in the list.