



I/O and Processing of Arrays

Processing An Array

- Single operations which involve entire arrays are not permitted in C. Thus, if a and b are similar arrays (**i.e. same data type, same dimensionality and same size**), assignment operations, comparison operations, etc. must be carried out on an element-by-element basis.
- This is usually accomplished within a loop, where each pass through the loop is used to **process one array element**.
- The **number of passes through the loop** will therefore **equal the number of array elements to be processed**.

Processing An Array - Example

- In numerical array, each array element represents a single numerical quantity, as illustrated in the next example:
 - *Suppose we want to read a list of n floating-point quantities and then calculate their average.*
 - *After the average calculation, we will compute the deviation of each numerical quantity about the average, using the formula*

$$d = x_i - avg$$

Where x_i represents each of the given quantities, $i = 1, 2, \dots, n$, and avg represents the calculated average.

Processing An Array - Example

- In order to solve this problem we must store each of the given quantities in a one-dimensional, floating-point array. This is an essential part of the program.
- *Let us define list to be a 100-element, floating-point array. However, we need not make use of all 100 elements. Rather, we shall specify the actual number of elements by entering a positive integer quantity (*not exceeding 100*) for the integer variable n .*



Processing An Array - Example

- Here is a complete C program:

/ calculate the average of n numbers, then computer the deviation of each number about the average */*

```
#include<stdio.h>
void main(void) {
int n, count;
float avg, d, sum = 0;
float list[100];
/* read a value for n */
printf("\n How many numbers will be averaged? " );
scanf("%d",&n);
```



Processing An Array - Example

- Here is a complete C program:

/ calculate the average of n numbers, then computer the deviation of each number about the average */*

```
printf("\n");  
/* read the numbers and calculate their sum */  
for ( count = 0 ; count < n; ++count)  
{  
    printf("i = %d    X = ", count + 1 );  
    scanf("%f", &list[count]);  
    sum += list[count];  
}
```



Processing An Array - Example

- Here is a complete C program:

/ calculate the average of n numbers, then computer the deviation of each number about the average */*

```
/* calculate and display the average */
```

```
avg = sum / n;
```

```
printf("\n The average is %5.2f\n\n", avg);
```

```
/* calculate and display the deviations about the average */
```

```
for ( count = 0; count < n; ++count) {
```

```
    d = list[count] - avg;
```

```
    printf("i = %d   X = %5.2f   d = %5.2f\n", count+1, list[count], d); }
```

```
}
```

Processing An Array - Example

- Here is a complete C program:

*/ * calculate the average of n numbers, then computer the deviation of each number about the average */*

Now suppose the program is executed using the following five numerical quantities:

$$x_1 = 3$$

$$x_3 = 12$$

$$x_5 = 3.5$$

$$x_2 = -2$$

$$x_4 = 4.4$$

Processing An Array - Example

- Here is a complete C program:

*/ * calculate the average of n numbers, then computer the deviation of each number about the average */*

How many numbers will be averaged? 5

i = 1 X = 3

i = 2 X = -2

i = 3 X = 12

i = 4 X = 4.4

i = 5 X = 3.5

The average is 4.18



Processing An Array - Example

- Here is a complete C program:

*/ * calculate the average of n numbers, then computer the deviation of each number about the average */*

i = 1	X = 3.00	d = -1.18
i = 2	X = -2.00	d = -6.18
i = 3	X = 12.00	d = 7.82
i = 4	X = 4.40	d = 0.22
i = 5	X = 3.50	d = -0.68



Practice Questions – Exclusively designed by Mr. Sumit Mittu, LIE.

- How does the computer react to the following code:

```
int main()
{
    int x[3], n=3;
    while(n--)
        scanf("%d", &x[n]);
    printf("Values are:");
    for(n=0;n<=3;n++)
        printf("\t %d", x[n]);
    return x[0];
}
```

Practice Questions – Exclusively designed by Mr. Sumit Mittu, LIE.

- Write a program to print first N terms of the Fibonacci series in reverse order (i) using arrays and (ii) **without using arrays**.

- Write a program to print the Kth row of Pascal's triangle. The Pascals' Triangle is something like:**

