



Jiangxi University of Science and Technology

Chapter 8 Arrays

lecture0802 Arrays as Function Arguments



8.3 Arrays as Function Arguments

➤ Pass by value 传值

- **Individual array elements** are passed to a function by including them as subscripted variables in the function call argument list
- `findMin(grades[2], grades[6]);`

➤ Pass by reference 传址

- When passing **a complete array** to a function, the called function receives access to **the actual array**, rather than a copy of the values in the array
- `findMax(grades);`

8.3 Arrays as Function Arguments

➤ Program 8.4 Find the maximum value

```
1.  #include <stdio.h>
2.  #define MAXELS 5
3.  void findMax(int [MAXELS]);
4.  int main()
5.  {
6.      int nums[MAXELS] = {2, 18, 1, 27, 16};
7.      findMax(nums);
8.      return 0;
9.  }
10. void findMax(int vals[MAXELS])
11. {
12.     int max = vals[0];
13.     for (int i = 1; i < MAXELS; i++)
14.         if (max < vals[i])
15.             max = vals[i];
16.     printf("The maximum value is %d\n", max);
17. }
```

数组名代表数组首元素地址

Size can be omitted

8.3 Arrays as Function Arguments

➤ Program 8.4 The starting address of the array is passed

```
1  #include <stdio.h>
2  #define MAXELS 5
3  void findMax(int [MAXELS]);
4  int main()
5  {
6      int nums[MAXELS] = {2, 18, 1, 27, 16};
7      findMax(nums);
8      return 0;
9  }
```

8.3 Arrays as Function Arguments

Program 8.5 Find the maximum value

```
1.  #include <stdio.h>
2.  int findMax(int [], int);
3.  int main(){
4.      #define MAXELS 5
5.      int nums[MAXELS] = {2, 18, 1, 27, 16};
6.      int max=findMax(nums, MAXELS);
7.      printf("The maximum value is %d\n", max);
8.      return 0;
9.  }
11. int findMax(int vals[], int numels){
12.     int max = vals[0];
13.     for (int i = 1; i < numels; i++)
14.         if (max < vals[i])
15.             max = vals[i];
16.     return (max);
17. }
```

8.4 Case Study: Computing Averages and Standard Deviations

➤ Requirements Specification

- Two statistical functions are created to determine the **average** and **standard deviation**, respectively, of an array of numbers
 - Need two functions
 - Determine the **average**...
 - Determine the **standard deviation**...
- ...of **a list of integer numbers**

8.4 Case Study: Computing Averages and Standard Deviations

➤ Analyze the Problem

- Determine the **input** items: list of integer numbers
- Determine the desired **outputs**: (1) average, and (2) standard deviation
- List the **algorithms** relating the inputs and outputs:
- **Average Function:**
 - Calculate the average by adding the grades and dividing by the # of added grades

8.4 Case Study: Computing Averages and Standard Deviations

➤ Analyze the Problem

— Standard Deviation Function:

- Subtract the average from each individual grade. Each number in the new set is called a deviation.
- Square each deviation found in Step 1.
- Add the squared deviations and divide the sum by the number of deviations.
- The square root of the number found in Step 3 is the standard deviation.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

8.4 Case Study: Computing Averages and Standard Deviations

➤ **Problem-Solver Algorithm is adapted:**

1. *Initialize an array of integers*
2. *Call the average function*
3. *Call the standard deviation function*
4. *Display the returned value of the average function*
5. *Display the returned value of the standard deviation function*

8.4 Case Study: Computing Averages and Standard Deviations

Program 8.6 Computing Averages and Standard Deviations

```
1.  #include <stdio.h>
2.  #include <math.h>
3.  double findAvg(int [], int);
4.  double stdDev(int [], int, double);
5.  int main() {
6.      #define N 10
7.      //Initialize an array of integers
8.      int values[N]={98,82,67,54,78,83,95,76,68,63};
9.      double average, stddev;
10.     //Call the average function
11.     average = findAvg(values, NUMELS);
12.     //Call the standard deviation function
13.     stddev= stdDev(values, NUMELS, average);
14.     //Display the average
15.     printf("The average of the numbers is %5.2f\n", average);
16.     //Display the deviation
17.     printf("The standard deviation of the numbers is %5.2f\n", stddev);
18.     return 0;
19. }
```

```
21. double findAvg(int nums[], int numel)
22. {
23.     double sumnums = 0.0;
24.     /* calculate the sum of the grades */
25.     for (int i = 0; i < numel; i++)
26.         sumnums = sumnums + nums[i];
27.     /* calculate and return the average */
28.     return (sumnums / numel);
29. }
30. {double stdDev(int nums[], int numel, double av)
31.     double sumdevs = 0.0;
32.     for (int i = 0; i < numel; i++)
33.         sumdevs=sumdevs+pow((nums[i]-av),2);
34.     return(sqrt(sumdevs/numel));
35. }
```

8.4 Case Study: Computing Averages and Standard Deviations

➤ Test and Debug the Functions of Program 8.6

```
5 int main()
6 {
7     #define NUMELS 10
8     //Initialize an array of integers
9     int values[NUMELS] = {98, 82, 67, 54, 78, 83, 95, 76, 68, 63};
10    double average, stddev;
11    //Call the average function
12    average = findAvg(values, NUMELS);
13    //Call the standard deviation function
14    stddev = stdDev(values, NUMELS, average);
15    //Display the returned value of the average function
16    printf("The average of the numbers is %5.2f\n", average);
17    //Display the returned value of the standard deviation function
18    printf("The standard deviation of the numbers is %5.2f\n", stddev);
19    return 0;
20 }
```

Reference



- <https://www.codesdope.com/blog/article/int-main-vs-void-main-vs-int-mainvoid-in-c-c/>

