一维数组

2022 年 8 月 16 日

1. 将数组的第一个元素和第二个元素互换位置，第三个和第四个元素互换位置，第五个和第六个元素互换位置……然后输出。要求数组的长度为 12，元素由用户输入，并且都是整数。

```c
#include <stdio.h>
int main()
{
    int a[12] = {}; // 声明长度为12的数组a，不进行初始化
    for(int i = 1; i <= 12; i++)
    {
        printf("Input the %d-th element:\n", i);
        scanf("%d", &a[i-1]); // 用户通过键盘输入a[i-1]的值
    }

    printf("The input array is below:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]); // 输出a[i]的值
    }
    printf("\n");

    for(int i = 0; i <= 10; i += 2)
    {
        // 交换a[i]和a[i+1]的值
        int temp = a[i];
        a[i] = a[i+1];
        a[i+1] = temp;
    }

```

```
    printf("The output array is below:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]);
    }
    printf("\n");
    return 0;
}
```

2. 设有一个数组，长度为 12，并且数组元素由用户输入。现在要求，把 7 号位置的元素取出来，放到 0 号位置，并且把原来的 1 号到 6 号位置的元素相应后移，然后输出调整后的数组。例如：当输入数组为 1,2,3,-4,5, 6, 7, 8, 9, 10,-11,12 的时候，输出数组为 8,1,2,3,-4,5,6,7,9,10,-11,12。

```
#include <stdio.h>
int main()
{
    int a[12];
    for(int i = 0; i < 12; i++)
    {
        printf("Input the %d-th integer:\n", i + 1);
        scanf("%d", &a[i]); // 输入a[i]的值
    }

    printf("The input array is:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]);
    }
    printf("\n");

    // 把a[7]的值保存在temp中
    int temp = a[7];
    for(int i = 7; i > 0; i--)
    {
        a[i] = a[i-1]; // 把a[i-1]的值代入到a[i]，因此更新了a[i]的值
        // 意思就是把a[i]的值后移
    }
    a[0] = temp; // 把temp的值装到a[0]中
```

```
26
27      printf("The output array is:\n");
28      for(int i = 0; i < 12; i++)
29      {
30          printf("%d\t", a[i]);
31      }
32      printf("\n");
33
34      return 0;
35  }
```

3. 设有一个数组，长度为 12，并且数组元素由用户输入。现在要求，把 7 号位置的元素取出来，放到 11 号位置，并且把原来的 8 号到 11 号位置的元素相应前移，然后输出调整后的数组。例如：当输入数组为 1,2,3,-4,5，6，7，8，9，10,-11,12 的时候，输出数组为 1,2,3,-4,5，6，7，9，10,-11,12，8。

```
1   #include <stdio.h>
2   int main()
3   {
4       int a[12];
5       for(int i = 0; i < 12; i++)
6       {
7           printf("Input the %d-th integer:\n", i + 1);
8           scanf("%d", &a[i]);
9       }
10
11      printf("The input array is:\n");
12      for(int i = 0; i < 12; i++)
13      {
14          printf("%d\t", a[i]);
15      }
16      printf("\n");
17
18      int temp = a[7]; // 把a[7]的值保存在temp中
19      for(int i = 7; i < 11; i++)
20      {
21          a[i] = a[i+1]; // 把a[i+1]的值代入a[i],
22          // 把a[i+1]的值前移
23      }
```

```
    a[11] = temp; // 把temp中的值存入a[11]

    printf("The output array is:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]);
    }
    printf("\n");

    return 0;
}
```

4. 输入一个正整数，求它用到的不同数字（0-9）出现的次数。例如给定 n=100311，那么它有 2 个 0，3 个 1 和 1 个 3。

```
#include <stdio.h>
int main()
{
    printf("Input an integer:\n");
    int n = 0;
    scanf("%d", &n);

    // 声明数组，并且把每个单元都初始化为0
    int occur_times[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}; //
        该数组用于存储每个数字出现的次数，一开始还没有检查数字，因此出现次数都为0
    while(n != 0) // 仍有数位需要被提取
    {
        int r = n % 10; // 取最低位的数字
        occur_times[r]++; // 它对应的出现次数加一
        n /= 10; // 截去最低位
    }
    printf("statistics info:\n");
    for(int i = 0; i < 10; i++)
    {
        if(occur_times[i] != 0) // 出现次数不为0
        {
            printf("%d: %d\n", i, occur_times[i]);
        }
    }
```

```
    return 0;
}
```

5. 输入 5 个整数，求其中的最大值和最小值，并把这些整数按照与输入相反的顺序输出。

```
#include <stdio.h>
int main()
{
    int a[5] = {0};
    for(int i = 1; i <= 5; i++)
    {
        printf("Input %d-th integer:\n", i);
        scanf("%d", &a[i-1]);
    }

    int max = a[0], min = a[0]; // 还没开始扫描数组，先假定第一个是最小和最大值
    for(int i = 1; i < 5; i++)
    {
        if(a[i] > max) // 找到更大的
        {
            max = a[i]; // 更新
        }
        if(a[i] < min) // 找到更小的
        {
            min = a[i]; // 更新
        }
    }
    printf("The maximum is: %d.\n", max);
    printf("The minimum is: %d.\n", min);

    int b[5] = {0}; // 创建另一个数组
    for(int i = 0; i < 5; i++)
    {
        b[4-i] = a[i]; // 复制到适当位置，形成逆序
    }

    for(int i = 0; i < 5; i++)
    {
        printf("%d\t", b[i]);
```

```
    }
    printf("\n");
    return 0;
}
```

6. 设一个数组包含 12 个元素，每一个元素都由用户输入。返回并输出最大元素所在的位置的下标和最小元素所在的位置的下标。如果出现多个并列最大元素或者并列最小元素，只取它们第一次出现的位置的下标。

```c
#include <stdio.h>
int main()
{
    int a[12];
    for(int i = 0; i < 12; i++)
    {
        printf("Input the %d-th element:\n", i + 1);
        scanf("%d", &a[i]);
    }

    printf("The input array is:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]);
    }
    printf("\n");

    // 先假定最大和最小值出现在第一个位置，随后扫描数组，再进行更新
    int maxi_at_loc = 0, mini_at_loc = 0;
    for(int i = 1; i < 12; i++)
    {
        if(a[i] > a[maxi_at_loc]) // 找到比当前发现的最大值还要更大的
        {
            maxi_at_loc = i; // 更新最大值的位置
        }
        if(a[i] < a[mini_at_loc]) // 找到比当前发现的最小值还要更小的
        {
            mini_at_loc = i; // 更新最小值的位置
        }
    }
```

```
31
32      printf("The first maximum element is at Location %d and its value is %d\n",
            maxi_at_loc, a[maxi_at_loc]);
33      printf("The first minimum element is at Location %d and its value is %d\n",
            mini_at_loc, a[mini_at_loc]);
34
35      return 0;
36  }
```

7. 设一个数组包含 12 个元素，每一个元素都由用户输入。先把数组的最小元素与第一个元素交换，然后把数组的最大元素与最后一个元素交换。如果出现多个并列最大元素或者并列最小元素，只取第一个最大或者最小元素来交换。

```
1   #include <stdio.h>
2   int main()
3   {
4       int a[12];
5       for(int i = 0; i < 12; i++)
6       {
7           printf("Input the %d-th element:\n", i + 1);
8           scanf("%d", &a[i]);
9       }
10
11      printf("The input array is:\n");
12      for(int i = 0; i < 12; i++)
13      {
14          printf("%d\t", a[i]);
15      }
16      printf("\n");
17
18      int mini_at_loc = 0; // 假定最小值在0号单元
19      for(int i = 1; i < 12; i++)
20      {
21          if(a[i] < a[mini_at_loc]) // 找到更小的
22          {
23              mini_at_loc = i; // 更新位置
24          }
25      }
26
```

```
    // 交换
    int temp = a[0];
    a[0] = a[mini_at_loc];
    a[mini_at_loc] = temp;

    int maxi_at_loc = 0; // 假定最大值在第一个单元
    for(int i = 1; i < 12; i++)
    {
        if(a[i] > a[maxi_at_loc]) // 找到更大的
        {
            maxi_at_loc = i; // 更新位置
        }
    }

    // 交换
    temp = a[11];
    a[11] = a[maxi_at_loc];
    a[maxi_at_loc] = temp;

    printf("The adjusted array is:\n");
    for(int i = 0; i < 12; i++)
    {
        printf("%d\t", a[i]);
    }
    printf("\n");

    return 0;
}
```