



数组元素移动

【问题描述】

将 a 数组（最多 10 个元素）中第一个元素移到数组末尾，其余数据依次往前平移一个位置。（可以借助一个临时变量）

【输入样例】

```
5
1 2 3 4 5
```

【输出样例】

```
2 3 4 5 1
```

【分析】

算法包括以下主要步骤：

- 把第一个元素的值取出放在临时变量 temp 中；
- 通过 $a[1] \rightarrow a[0]$, $a[2] \rightarrow a[1]$, $a[3] \rightarrow a[2]$, ..., $a[n-1] \rightarrow a[n-2]$, 实现其余元素前移；
- 将 temp 值送入 $a[n-1]$ 。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. const int N = 10;
4. int main()
5. {
6.     int a[N];
7.     int n;
8.     cin >> n;
9.     for (int i=0; i<n; i++)
10.    {
11.        cin >> a[i];
12.    }
13.     int temp = a[0];
14.     for (int i=0; i<n-1; i++)
```



```
15.  {
16.     a[i] = a[i+1];
17. }
18. a[n-1] = temp;
19. for (int i=0; i<n; i++)
20. {
21.     cout << a[i] << " ";
22. }
23. return 0;
24. }
```



最大数的位置

【问题描述】

输入 n 个不同的整数，存放在数组 $a[1]$ 至 $a[n]$ 中，输出最大数所在位置 ($n \leq 10000$)。

【输入样例】

```
5
67 43 90 78 32
```

【输出样例】

```
3
```

【分析】

设 max_value 存放最大值， max_pos 存放对应最大值所在的数组位置。

max_value 的初值为 $a[1]$ ， max_pos 的初值对应为 1。

枚举数组元素，找到比当前 max_value 大的数成为 max_value 的新值， max_pos 值为对应位置，输出最后的 max_pos 值。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. const int N = 10000;
4. int a[N+1];
5. int main()
6. {
7.     int n;
8.     cin >> n;
9.     for (int i=1; i<=n; i++)
10.    {
11.        cin >> a[i];
12.    }
13.    //赋最大值初值和初始位置
14.    int max_value = a[1];
15.    int max_pos = 1;
16.    //枚举数组，找到最大数和位置
17.    for(int i=2; i<=n; i++)
```



```
18.     {
19.         if (a[i] > max_value)
20.         {
21.             max_value = a[i];
22.             max_pos = i;
23.         }
24.     }
25.     //输出最大数所在数组中的位置
26.     cout << max_pos;
27.     return 0;
28. }
```



数据插入

【问题描述】

输入 n 个数，往指定位置之前插入一个数。

第 1 个数字位置为 1。

【输入格式】

第 1 行为 3 个正整数 n, p, x 。 n 表示输入数字个数， p 表示指定位置， x 表示要插入的数字，第 2 行为 n 个小于 100 的正整数。

【输出格式】

1 行插入之后的数字序列，中间以 1 个空格隔开。

【数据范围】

$$1 \leq n, p, x \leq 100$$

【输入样例】

```
5 2 9
3 4 5 6 7
```

【输出样例】

```
3 9 4 5 6 7
```

【参考程序】

```
#include<iostream>
using namespace std;
const int N = 100;
int a[N + 2];
int main()
{
    int n, p, x;
    cin >> n >> p >> x;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
```



```
    }  
    for (int i = n + 1; i >= p + 1; i--)  
    {  
        a[i] = a[i-1];  
    }  
    a[p] = x;  
    for (int i = 1; i <= n + 1; i++)  
    {  
        cout << a[i] << " ";  
    }  
    return 0;  
}
```