

这次课程主要巩固了一维数组的相关知识,发现问题比较大,课程上我也针对性的对同学们的程序进行了讲解,下面我再把这次课程上的题目进行一个讲解,配合内存示意图方便同学们能够更好的理解,后面布置了这周的作业

**再次提醒:**必须完成作业,即便没有完成也要把思路过程以注释的形式写出来,将作业写好打包为一个压缩文件,在微信上发给我!根据完成情况会有一定的奖赏和惩罚,我们的目的是竞赛,当然会有更高的要求,三天打鱼两天晒网什么事都做不好

## 1.习题回顾

### 题目描述

### 数据插入

版本号: 2021-07-31 有效期至: 2023-10-01

<b>【问题描述】</b> <ul style="list-style-type: none"><li>● 输入<math>n</math>个数, 往指定位置之前插入一个数。</li><li>● 第1个数字位置为1。</li></ul>	<b>【数据范围】</b> $1 \leq n, p, x \leq 100$
<b>【输入格式】</b> <ul style="list-style-type: none"><li>● 第1行为3个正整数<math>n, p, x</math>。<math>n</math>表示输入数字个数, <math>p</math>表示指定位置, <math>x</math>表示要插入的数字, 第2行为<math>n</math>个小于100的正整数。</li></ul>	<b>【输入样例】</b> <pre>5 2 9 3 4 5 6 7</pre>
<b>【输出格式】</b> <ul style="list-style-type: none"><li>● 1行插入之后的数字序列, 中间以1个空格隔开。</li></ul>	<b>【输出样例】</b> <pre>3 9 4 5 6 7</pre>

### 题目分析:

1. 先读问题描述, 需要我们输入三个变量以及一个数组,,
2. 观看输入样例, 5 2 9 分别是这三个变量, 5 表示的是原始数组的长度, 2 表示的是插入元素的位置, 9 表示的是插入元素的值, 下面一行表示的是原始数组的元素分别是多少, 他的长度是 5, 与第一个变量是对应的,
3. 注意:上面提到的这些都是我们要输入的东西,输入的东西是我们提供给程序的内容, 而这

个程序的输出是插入元素后的数组,

4. (上课的时候有的同学题目没有读清楚,这个要引起注意, 读题策略: 1. 首先读问题描述, 大概理解题的意思,思考需要几个变量 2. 其次看输入格式的同时看输入样例一一对应 3. 观看输出格式了解什么意思)
5. 注意:我发现有的同学只在乎程序能不能跑通, 而不在于程序哪里写的不对,为什么跑不通, 哪里出问题了,这样是不对的,

## 思路分析:

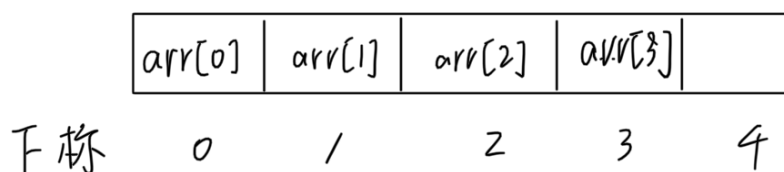
题目实现的是在数组中插入一个元素,但是在计算机中,分配给一个数组的内存是固定的,不能平白无故插一个内存块进去,所以我们要做的只能是先将其他部分的元素进行移动,然后空出一个位置给这个元素,再把这个元素放进去,

1. 创建初始变量以及数组,用来存储我们的输入(输入必须存到变量或者数组里,这样计算机才能够记住他们)
2. 从输入端接受输入, 变量可以直接连续输入, 数组元素必须从循环内输入
3. 先移动插入位置  $p$  之后的元素,让他们整体向后移动一个位置(具体实现是,从最后一个位置的后一个位置开始,向前遍历, 每一个位置的值等于前一个位置的值)
4. 位置  $p$  插入元素  $x$ (所有的元素已经向后移动一个位置,所以这个位置就空出来了)
5. 用循环输出这个数组,c++不可以直接 `cout<< arr;`输出这个数组,只能利用循环,也就是第一步的逆过程

## 内存图示

```
const int MAX = 1000;
```

```
int arr[MAX];
```



有的同学不理解 下标与内存中存的值的关系,

有的同学不理解下标与内存中值的关系,假如i是当前位置的下标,arr[i]就是当前位置的值,

可以将arr[i]当作一个变量来使用, 例如

```
arr[i]++; // 将该位置的值加1
```

```
arr[i] = x; // 将该位置的值置为x
```

所以数组整体向后移动一个位置就是从位置num向前遍历到位置p,每次循环内让当前位置的值等于前一个位置的值

```
for(int i = num; i >= p; -i){  
    arr[i] = arr[i-1];  
}
```

然后让p位置的值等于x(注意有个易错点,题目中第一个位置是数字1,这与计算机中的规则不同,所以我们不能写arr[p] = x;)应该写

```
arr[p-1] = x;
```

具体问题应该具体分析,这里教大家一个方法,带一个数字进去就可以,

假如题目要求插入的位置是1,也就是 $p=1$ ,但是我们实际插入的应该是位置0,也就是 $p-1$ ,

所以应该写

```
arr[p-1] = x;
```

最后我们要将数组输出,但是目前数组的大小变成了 $num+1$ ,所以循环满足条件要写成  $i < num+1$ ;

## 程序示意图

```
1  #include<iostream>
2  using namespace std;
3  const int MAX = 101;    // 定义常量
4  int arr[MAX];  // 定义一个大容量数组
5  int main()
6  {
7      int num,p,x;    // 定义三个变量
8      cin >> num >> p >> x;  //输入三个变量
9      for (int i = 0; i < num; i++)  //输入数组元素
10     {
11         cin >> arr[i];
12     }
13     for (int i = num; i >=p; i--)  //移动数组元素
14     {
15         arr[i] = arr[i-1];
16     }
17     arr[p-1] = x;    //插入数组元素
18     for (int i = 0; i <num+1 ;i++) // 输出数组元素
19     {
20         cout << arr[i];
21         /* code */
22     }
23
24
25     return 0;
26 }
```

记得与上面的图示一一对应

## 2.本周作业

### 1. 移动元素

要求输入一个大小为 n 的数组,将最后一个元素移动到数组的开头,然后其余元素向后移动一格

输入格式:1. 数组大小 2. 数组元素

输入样例:

5  
1 2 3 4 5

输出样例:

5 1 2 3 4

提示: 是否可以直接修改上面的程序,然后使用它

### 2. 房间的门

房间的门

版本号: 2021-07-31 有效期至: 2023-

【问题描述】

- 宾馆里有N（最大1000）个房间，从1~N编号。
- 第一个服务员把所有的房间门都打开了，第二个服务员把所有编号是2的倍数的房间“相反处理”，第三个服务员把所有编号是 3的倍数的房间作“相反处理” …，以后每个服务员都是如此。
- 当第N个服务员来过，哪几扇门是打开的。
- “相反处理”指的是：原来开着的门关上，原来关上的门打开。

【输入样例】

100

【输出样例】

1 4 9 16 25 36 49 64 81 100

提示:使用 bool 数组,其中 1 表示门开着,0 表示门关着, 使用嵌套循环处理