



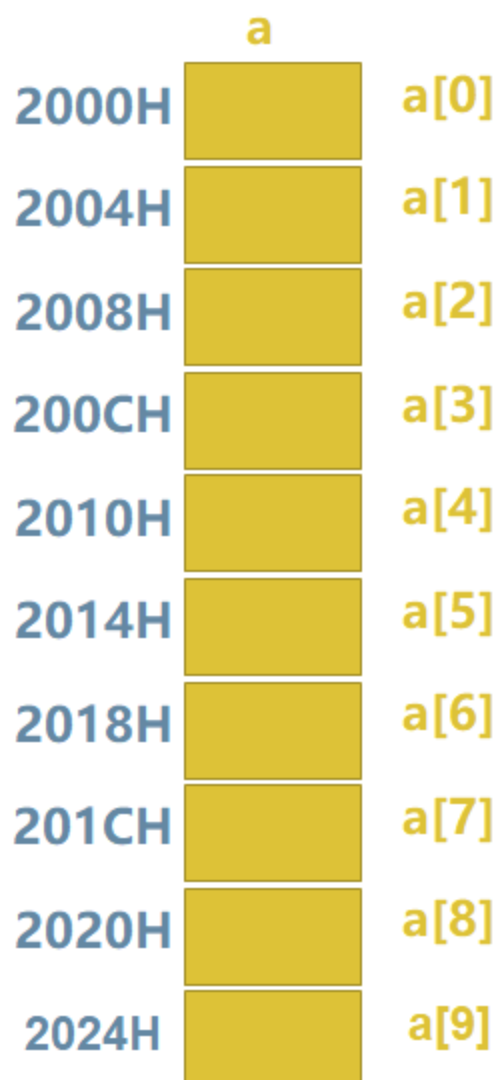
数组名和指针

数组与变量一样，在内存中占据单元。数组是由一块连续的内存单元组成的。

数组名是这块连续内存单元的首地址。

数组元素的首地址是指它所占有的几个内存单元的首地址。

因此数组名就是指针。





指针变量的+、-运算

指针变量的内容是内存地址，也可以进行加、减运算。

“p++”的意思是“广义的加 1”，不是 p 的值（地址）加 1，而是根据 p 指向的数据类型所占空间的大小增加，即刚好“跳过”一个元素所占的空间，达到下一个元素的位置。

例如，如果 p 指向的是 int 型的数据，那么

- p++实际上表示地址增加了 sizeof(int)，即刚好“跳过”一个整数的空间，达到下一个整数。
- p--就是向前“跳过”一个整数的空间，达到前一个整数。
- p+3 就是指向后面第 3 个整数的地址。



通过指针引用数组元素

设有数组 a ，则 $a+1$ 指向数组的下一个元素，而不是下一个字节，所以：

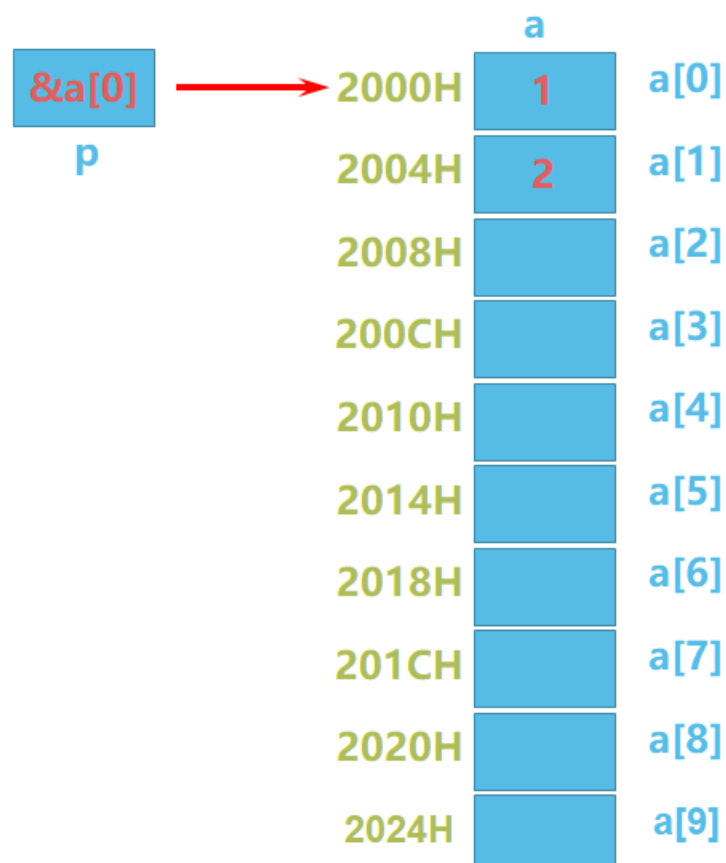
- $a+1$ 、 $\&a[1]$ 均表示 1 号元素 $a[1]$ 的地址。
- 类推可知， $a+i$ 、 $\&a[i]$ 均表示 i 号元素 $a[i]$ 的地址。

这样就可以用两种方法访问数组元素：

- 第一种方法为下标法，即用 $a[i]$ 形式访问数组元素 $a[i]$ 。
- 第二种方法为指针法，即用 $*(a+i)$ 形式访问数组元素 $a[i]$ 。

因此，可以使用下面的代码，通过指针引用数组元素。

```
1. int a[10];  
2. *a = 1;  
3. a[0] = 1;  
4. *(a+1) = 2;  
5. a[1] = 2;
```





例：数组元素访问

【问题描述】

输入 10 个整数，用指向数组的指针变量输出数组的全部元素。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int a[10];
6.     for (int i=0; i<10; i++)
7.     {
8.         cin >> *(a+i);
9.     }
10.    for (int i=0; i<10; i++)
11.    {
12.        cout << *(a+i) << "\t";
13.    }
14.    return 0;
15. }
```

直接拿 a 当指针用，a 指向数组的开始元素，a+i 是指向数组的第 i 个元素的指针。

【参考结果】

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10