2020春季学期C++整理3.0

1.关于指针的几种定义：

1. int a;表示一个内存空间，这个空间用来存放一个整数（int）；
2. int* a;表示一个**内存空间**，这个空间**用来存放一个指针**，这个指针**指向一个存放整数的空间**，即1中提到的空间；
3. int * a;表示一个内存空间，这个空间用来存放一个指针，这个指针指向一个存放指针的空间，并且指向的这个空间中的指针，指向一个整数。也简单的说，指向了一个b)中提到的空间；*
4. int (* a)[4];表示一个内存空间，这个空间用来存放一个指针，这个指针指向一个长度为4、类型为int的数组；和int * a的区别在于，++、+=1之后的结果不一样，其他用法基本相同。
5. int (*a)(int);表示一个内存空间，这个空间用来存放一个指针，这个指针指向一个函数，这个函数有一个类型为int的参数，并且函数的返回类型也是int。

注意：

int p[]和int (*p)[]

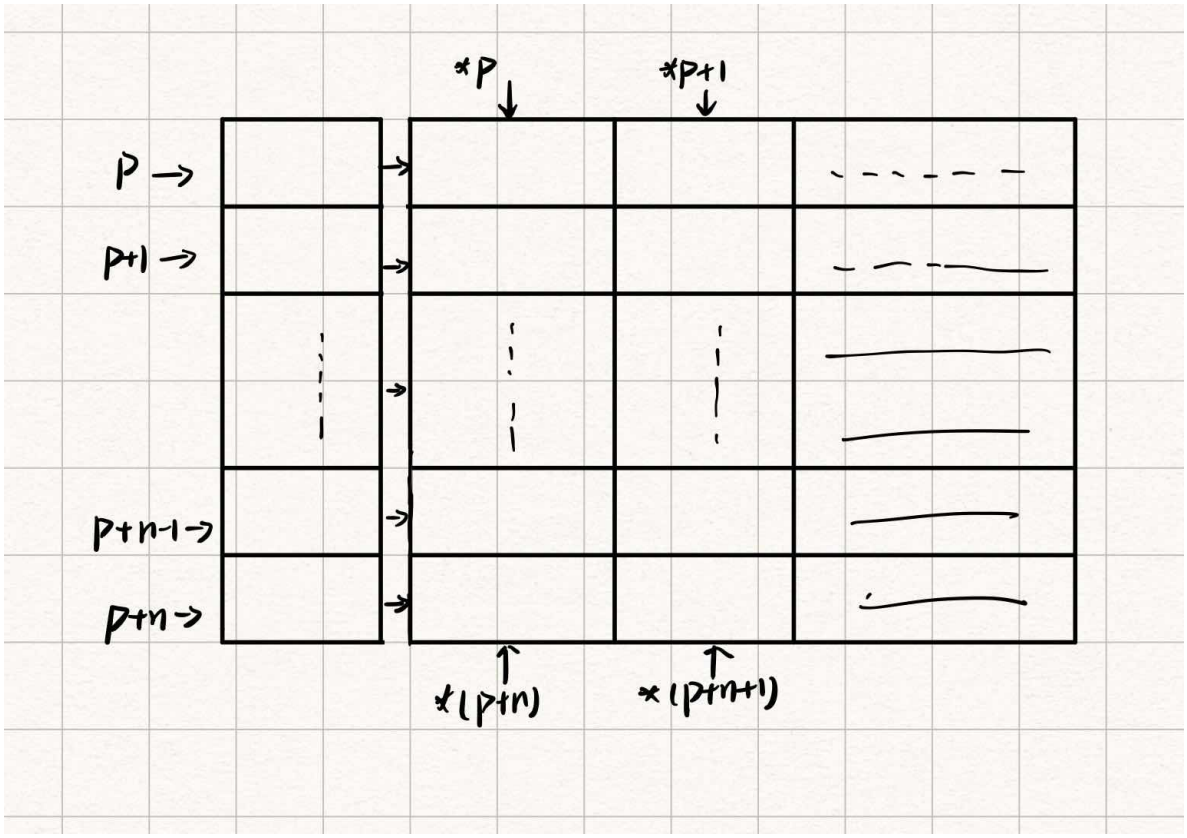
前者是**指针数组**，后者是**指向数组的指针。**

前: 指针数组;是一个**元素全为指针的数组**.
后: 数组指针;可以直接理解是指针,只是这个指针类型不是int也不是char而是 int [4]类型的数组

int * p[4]------**p是一个指针数组**，每一个指向一个int型的
int (* q)[4]---------**q是一个指针**，指向int[4]的数组。

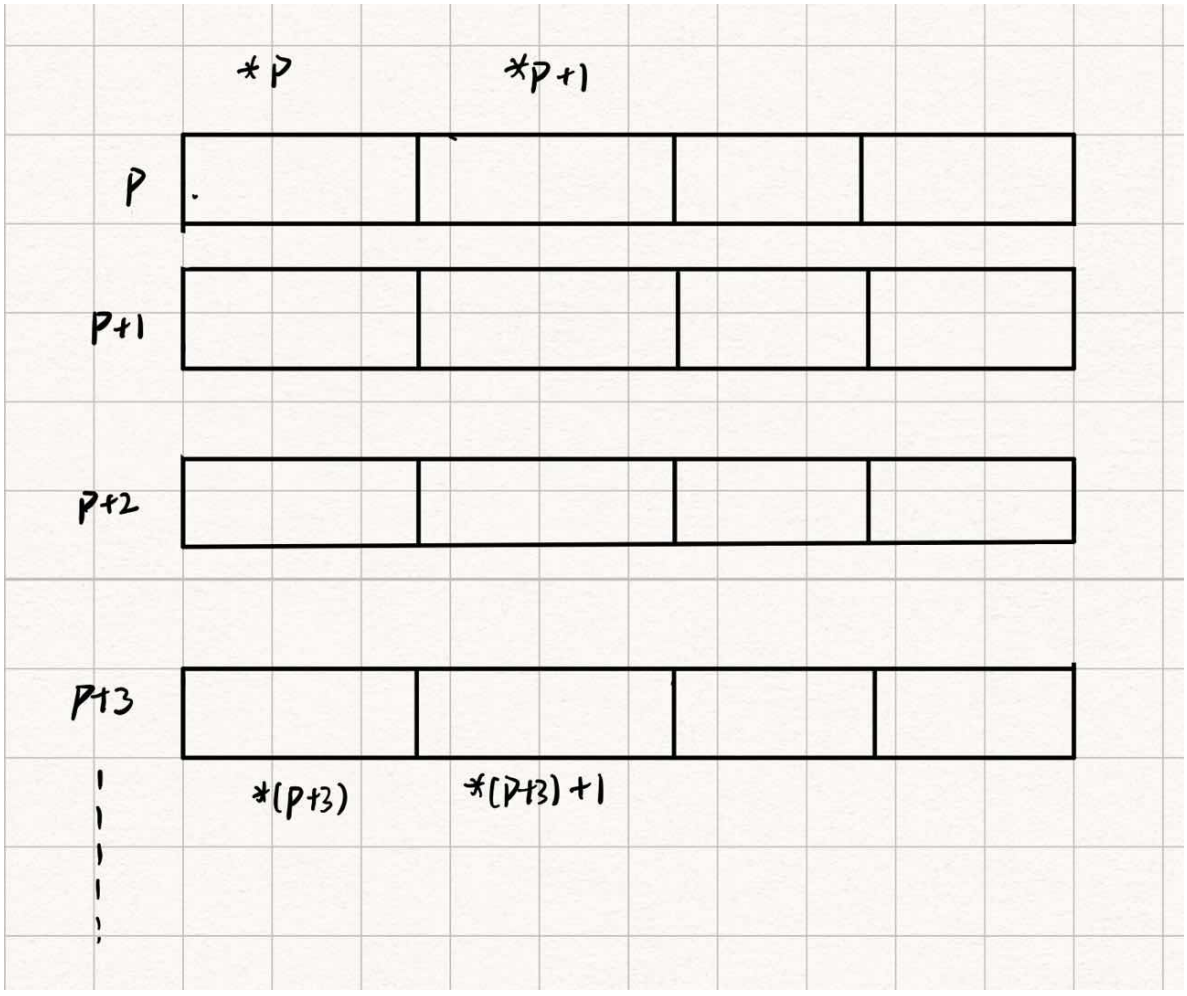
定义涉及两个运算符：“ * ”（间接引用）、“[]”（下标），**“[]”的优先级别大于“*”的优先级别。**

下面用几幅图来表示下：
对于int *p[4]:它**首先是个大小为4的数组**，即p[4]；剩下的“int *”作为补充说明，即**说明该数组的每一个元素为指向一个整型类型的指针**



(PS:最后应该是*(p + n) + 1)

再看一下**int (*p)[4]**:它**首先是个指针**，即*q，剩下的"int [4]"作为补充说明，即说明**指针q指向一个长度为4的数组。**

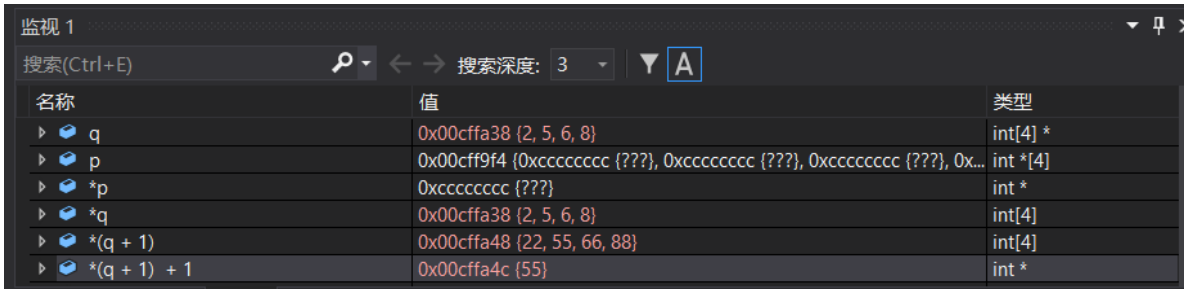


综上所述，我们可以将这两种指针定义出的p**看作是**一个二维数组来理解。

可以通过一些样例来说明:

```cpp
#include<iostream>
using namespace std;
int main()
{
    int a[2][4]={{2,5,6,8},{22,55,66,88}};

    int c[4]={5,8,9,4};

    int d[3]={23,12,443};

    int *p[4],(*q)[4];

    q=a;

    *p=c;

    *(p+1)=d;
     return 0;
}
```

通过调试追踪可看出:

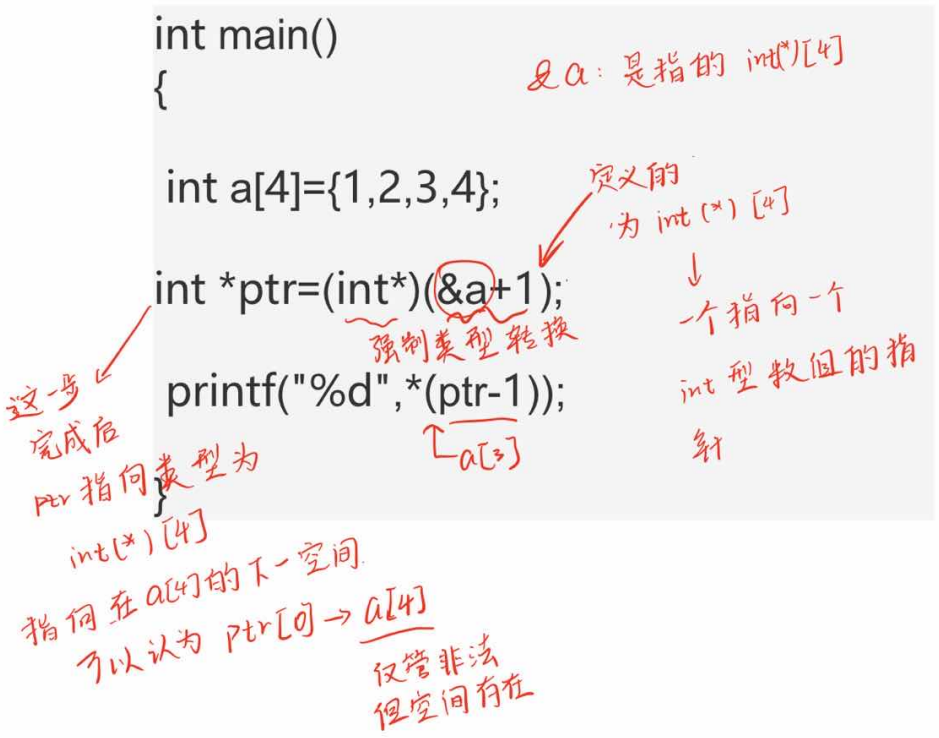


确实可以理解为一个二维数组的形式。

2.标注注释:

注释一:

注释1



注释二：

注释2

```
1  int main()
2  {
3          int a[5][5];
4          int(*p)[4];
5          p = (int(*)[4])a;
6          printf("%p,%d\n", &p[4][2] - &a[4][2],&p[4][2] - &a[4][2]);
7
8          system("pause");
9          return 0;
10 }
```


注释三：

注释3

```c
int main()
{
        char *a[] = { "work", "at", "alibaba" };
        char **pa = a;
        pa++;
        printf("%s\n", *pa);
        system("pause");
        return 0;
}
```