const char* p与char const *p的辨析

H1

> const char * arr = "123";

//字符串123保存在常量区，const本来是修饰arr指向的值不能通过arr去修改，但是字符串“123”在常量区，本来就不能改变，所以加不加const效果都一样

> char * brr = "123";

//字符串123保存在常量区，这个arr指针指向的是同一个位置，同样不能通过brr去修改"123"的值

> const char crr[] = "123";

//这里123本来是在栈上的，但是编译器可能会做某些优化，将其放到常量区

> char drr[] = "123";

//字符串123保存在栈区，可以通过drr去修改

```
//
// Created by wbzhang on 2020/9/15.
//
#include <iostream>
#include <vector>

using namespace std;

int main() {

//  int a = 2;
//  printf("%d",a++);
//  printf("%d",++a);

    // for循环判断 中间表达式 是否为真
    for(int i = 0; i + 20; i--)
```

```
17            cout << "hello "<< i<< endl;
18
19        const char *p = "123";
20        // const修饰 char *p；表明以p所指向的字符串数组不可变
21        // p为(字符串常量)的指针，p指向该字符串的首字母。并且所指向内容不可变(即*p代表的字符串为常量，*p不可变，但是p可变)
22        char const *p1 = "456"; // 常量指针，p1所指向内容不可变;
23        // const 修饰 *p1，其类型为char;
24
25        char *const p2 = "369"; // 指针常量，p2本身不可变
26        // const修饰 p2，p2为(指针常量)，指向一个字符串
27        // 该指针的指向p2不可变；但所指向的内容*p2可变
28
29        // before
30        cout<<"-- before --"<<endl;
31        cout<<"*p: "<<*p<<endl; // p指向该字符串的首字母，即*p=1
32        cout<<"p: "<<p<<endl; // p为该字符串常量的指针，将输出 123
33
34        cout<<"*p1: "<<*p1<<endl; // p1指向该字符串的首字母,即*p1=4
35        cout<<"p1: "<<p1<<endl; //  p1为该字符串常量的指针，将输出 456
36
37        cout<<"*p2: "<<*p2<<endl; // p2指向字符串常量的首字母，因此为3
38        cout<<"p2: "<<p2<<endl; // p2为指针常量，将输出字符串常量 369
39
40        // after
41        cout<<"-- after --"<<endl;
42        p = "5"; // p本身可变
43    //  *p = 'a'; //p所指向的内容不可变
44    //  *p = "abd";
45
46        p1 = "678"; // p1本身可变
47    //  *p1 = 'p'; //p1所指向内容不可变
48    //  *p1 = "bcd"; //p1所指向内容不可变
49
50        char str[100]="Hello World";
51    //  *p2 = str[1];
52    //  p2 = "a";  //p2不可变,
53    //  p2[0] = 'a'; //所指向内容可变,且 p2指向字符串中的第一个字符，将报错
54    //  (*p2) = 'a'; //所指向内容可变,且 p2指向字符串中的第一个字符,将报错
55    //  *p2 = "amd";  //所指向内容可变
56
57
58        cout<<"*p: "<<*p<<endl;
59        cout<<"p: "<<p<<endl;
60
61        cout<<"*p1: "<<*p1<<endl;
62        cout<<"p1: "<<p1<<endl;
63
64        cout<<"*p2: "<<*p2<<endl;
65        cout<<"p2: "<<p2<<endl;
66
67        cout << "done" << endl;
```

```
68
69    }
```