

注：本文中，“指向xxx的函数指针”意思是一个函数指针，指向的函数返回类型为 xxx

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个void

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个长度为3的void数组

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个长度为3的指向void的指针数组

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个长度为3的指向void的函数指针数组

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个长度为3的指向指向void的函数指针的指针数组

```
void(*(*f[3])(const char *p))(int x);
```

- f 是一个长度为3的指向指向void的函数指针的函数指针数组

```
void(*(*f[3])(const char *p))(int x);
```

- 也就是说，f 数组中有三个元素，都是指向函数的指针，这个函数指针指向的函数接受一串字符返回一个函数指针，这个函数指针指向的函数接受一个整型，返回void。
- 看看要输出的内容，就是输出“str=那个字符串”，再输出“x=那个整型”。

```
f[0] ("Hello") (2);  
f[0] ("Tongji") (7);
```

C# Microsoft Visual Studio 调试

```
str = Hello  
x = 2  
str = Tongji  
x = 7
```


f[0]("Hello")(2);

- f[0]指向一个函数 f_str，传参("Hello")，返回一个函数指针，对应的函数f_sub传参(2)，返回void。

```
5  /* ... */
12 void f_sub(int x)
13 {
14     cout << "x = " << x << endl;
15 }
16
17 /* ... */
24 void (*f_str(const char*str))(int)
25 {
26     cout << "Str = " << str << endl;
27     return f_sub;
28 }
29
30 /* ... */
37 int main()
38 {
39     void(*(*f[3])(const char *p))(int x);
40     f[0] = f_str;
41     f[0]("Hello")(2);
42     f[0]("Tongji")(7);
43
44     return 0;
45 }
```

Microsoft Visual St

Str = Hello
x = 2
Str = Tongji
x = 7

```
#include <iostream>
using namespace std;

void f_sub(int x)
{
    cout << "x = " << x << endl;
}

void (*f_str(const char *str))(int)
{
    cout << "Str = " << str << endl;
    return f_sub;
}

int main()
{
    void(*(*f[3])(const char *p))(int x);
    f[0] = f_str;
    f[0]("Hello")(2);
    f[0]("Tongji")(7);

    return 0;
}
```