

1.

```
1 int main()
2 {
3     char arr[] = "abcdef"; //abcdef\0
4
5     printf("%d\n", sizeof(arr)); //7
6     printf("%d\n", sizeof(arr + 0)); //4/8 计算的是地址的大小 arr+0是首元素的地址
7     printf("%d\n", sizeof(*arr)); //*arr是首元素 首元素是a a的大小是1
8     printf("%d\n", sizeof(arr[1])); //arr[1] = b 为1
9     printf("%d\n", sizeof(&arr)); //4/8 虽然是数组的地址但也是地址 所以是4/8
10    printf("%d\n", sizeof(&arr+1)); //4/8 跳过整个数组后的地址 但也是地址
11    printf("%d\n", sizeof(&arr[0]+1)); //4/8 第二个元素的地址
12
13
14    printf("%d", strlen(arr)); //6
15    printf("%d", strlen(arr + 0)); //6
16    printf("%d", strlen(*arr)); //error
17    printf("%d", strlen(arr[1])); //error
18    printf("%d", strlen(&arr)); //数组的地址
19    printf("%d", strlen(&arr + 1)); //6
20    printf("%d", strlen(&arr[0] + 1)); //随机值
21
22    char* p = "abcdef";
23
24    printf("%d\n", sizeof(p)); //4/8 指针变量p的大小
25    printf("%d\n", sizeof(p+1)); //4/8 p存放a的地址 p+1是b的地址
26    printf("%d\n", sizeof(*p)); //1
27    printf("%d\n", sizeof(p[0])); //1 int arr[10]; arr[0] == *(arr + 0) p
    [0] == *(p+0) == 'a'
28    printf("%d\n", sizeof(&p)); //4/8
29    printf("%d\n", sizeof(&p+1)); //4/8
30    printf("%d\n", sizeof(&p[0]+1)); //4/8
31
32    printf("%d", strlen(p)); //6
33    printf("%d", strlen(p+ 1)); //5
34    printf("%d", strlen(*p)); //error
35    printf("%d", strlen(p[1])); //error
36    printf("%d", strlen(&p)); //随机值
37    printf("%d", strlen(&p+ 1)); //随机值
38    printf("%d", strlen(&p[0] + 1)); //5
```

```
39 return 0;
40 }
```

## 2.二维数组

```
1 int main()
2 {
3     int a[3][4] = {0};
4     printf("%d\n", sizeof(a)); //3x4x4=48
5     printf("%d\n", sizeof(a[0][0])); //4
6     printf("%d\n", sizeof(a[0])); //16
7     printf("%d\n", sizeof(a[0] + 1)); //4 第一行第二个元素的地址
8     printf("%d\n", sizeof(*(a[0] + 1))); //4
9     printf("%d\n", sizeof(a+1)); //a是二维数组的数组名 没有sizeof(arr) , 也没有
    &(数组名
10    //所以a是首元素地址 而二维数组看成一维数组时 二维数组的首元素就是它的第一行 a
    就是第一行的(首元素)的地址
11    //a+1就是第二行的地址
12    printf("%d\n", sizeof(*(a+1))); //16 计算第二行的大小
13    printf("%d\n", sizeof(&a[0] + 1)); //4 第二行的地址
14    printf("%d\n", sizeof(&a[0] + 1)); //计算的是第二行的大小 单位是字节
15    printf("%d\n", sizeof(*a)); // *a表示第一行 a是首元素地址
16    printf("%d\n", sizeof(a[3])); //16 sizeof括号中的不会计算
17 }
```