day3: C语言

*
1 引用
2 类型
3 地址降级

[] 等价于 * （指针+单位）
[]
* & ++ —

一、指针

【1】getchar
1--- 获取一个字符
2--- 清除一个垃圾字符

【2】ASCII
0 '\0'
10 '\n'
32 '空格'
65-90 'A'- 'Z'
97-122 'a' - 'z'

【3】数组
1--- 元素类型相同
2--- 存储是连续的
3--- 数组名：代表首地址

【4】指针
1--- 指针变量
2--- 地址常量

【5】指针运算
1--- &：取地址符
2--- *
1>数据类型的一部分
2>指针解引用
3>地址的降级

3--- 加减运算
加法：加的是单位长度

4--- []
指针加单位长度后，取*运算

5--- &和*互为逆运算

【6】指针数组

【7】二维数组

int a[3][4] = {0,1,2,3};

1--- a：首元素的地址

a[0]、a[1]、a[2]：每一行的首地址

a：加一移动一行,行指针

a[0]：加一移动一列，列指针

2---

a[1][2]

*(a[1]+2)

*(*(a+1)+2)

【8】数组指针

int a[3][4] = {0,1,2,3};

int (*p)[4] = a;

相同：p与a的用法一样。

不同：a是常量，p变量

【9】const

const int * p = &a; //常量化*p

int * const p = &a; //常量化p

二、函数

【1】主函数

argc //命令行参数的个数

argv //本质是数组，元素类型char*

【2】库函数

<string.h>

char * strcpy (char *, char *) ;

【3】自定义函数

1--- 参数

复制形式，地址形式

2--- 函数指针

指向函数的入口地址

3--- 指针函数

返回值是一个地址量的函数

4--- 函数指针数组
本质是数组，元素是函数指针类型

5--- 递归函数
函数调用自己本身
控制结束条件

6--- 回调函数

```
char *  fun (    int (*b)(char c, char * d),    float * e,    int f);
```

【4】

```
int atoi  (char *)   ;
```

头文：<stdlib.h>
功能：将数值形式的字符转换成整形

【5】 str

```
char * strcpy(char *, char *);
```
功能：字符串拷贝

```
char * strcat(char *, char *);
```
功能：字符串拼接

```
int strcmp(char *, char *);
```
功能：字符串比较

```
int strlen(char *);
```
实际字符串长度，不包括'\0'

day3： homework

1 实现库函数 atoi

2 实现库函数strcmp

3 打印一个9*9乘法表