# 指针变量使用小结

- 有关指针变量的类型

| 定义 | 含义 |
| --- | --- |
| int i; | 定义整型变量i |
| int *p; | p是指向整型数据的指针变量 |
| int a[n]; | 定义数组a，元素类型为int，元素个数是n |
| int *p[n]; | p是指针数组，包含n个整型指针变量 |
| int (*p)[n]; | p是指向数组的行指针，数组每行有n个整型数 |
| int f(); | f是函数，返回值是int |
| int *p(); | p是函数，返回值是指针，该指针指向整型数据 |
| int (*p)(); | p是函数指针变量，指向int函数的入口地址。 |
| int **p; | p是指针变量，指向一个指向整型数据的指针 |

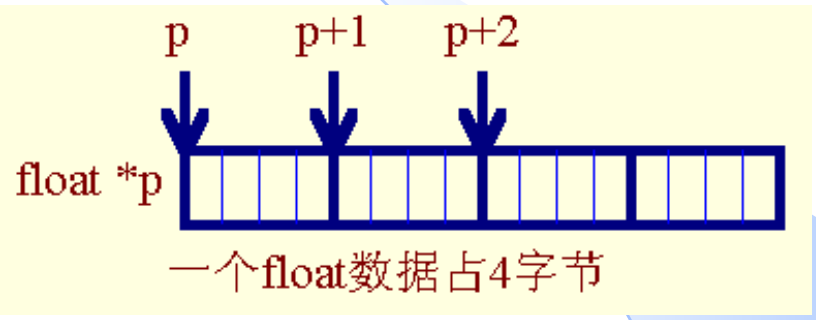
# 指针变量使用小结

指针变量运算小结

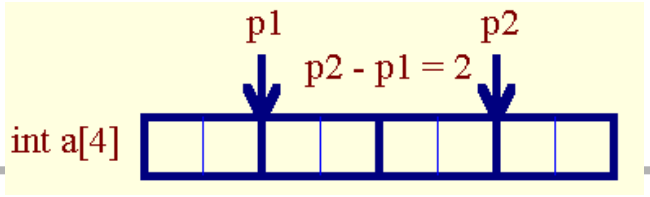
1、指针变量加/减运算

p++、p--、p+i、p-i、p+=i、p-=i

加1表示指向下一个数据。




2、指针变量相减。当p1、p2指向同一个数组的元素，指针相假 p2-p1等于p1、p2间的元素个数。　注意：指针相加无意义。



# 指针变量使用小结

3、指针变量赋值

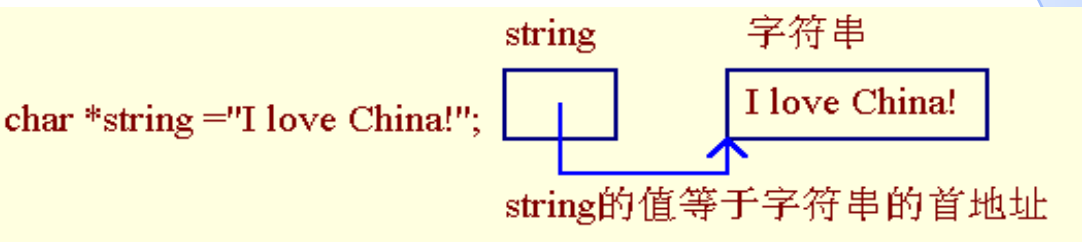
p = &a; 变量a的地址赋给p，即指针p指向a

p = array; 数组array首地址赋给p

p = &array[i]; 数组元素array[i]的地址赋给p

p = max; 函数max的入口地址赋给p

p1 = p2; 指针p2的值赋给指针p1，即p1、p2所指的数据相同




注意：“野指针”是指向“垃圾”内存（不可用内存）的指针

# 指针变量使用小结

4、空指针(null pointer ):不指向任何变量（对象）和函数

```
#define NULL   0
 char   *p=NULL;
```


5、空类型（通用）指针（void *）：基类型未确定的指针

- void *p，表示p是空类型指针，它可以指向任何数据类型。例如：

```
void *malloc( size_t size );
  int *p; p=(int*)malloc(sizeof(int)*10);
```

- 空类型指针与其他类型指针之间赋值时，应进行强制类型转换
- 例、

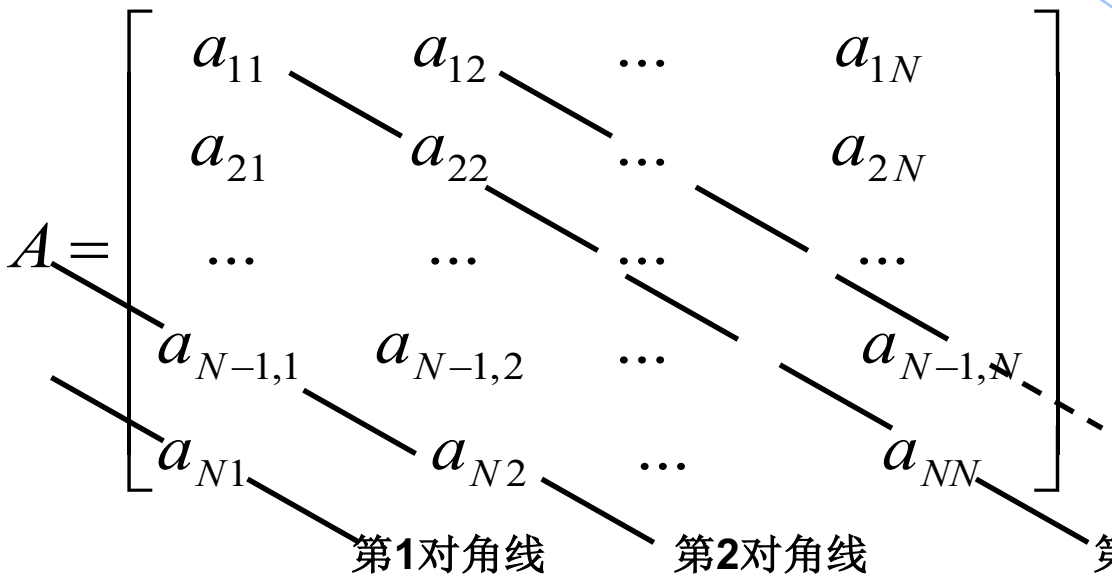
```
char *p1;
void *p2;
p1 = (char *)p2;
p2 = (void *)p1;
```

# 第12次课后练习

- 本次练习必做3道题，第14周末提交源程序和运行结果

1. 定义一个指向字符串的指针数组，用一个函数完成N个不等长字符串的输入，使得指针数组元素依次指向每一个输入的字符串。设计一个完成N个字符串按升序的排序函数（在排序过程中，要求只交换指向字符串的指针，不交换字符串）。在主函数中实现对排序后的字符串的输出。假设已知字符串的最大为80字节；根据实际输入的字符串长度来分配存储空间。

2.编写程序，求一个N×N方阵的第i对角线的元素之和。其中，方阵的第i对角线定义如下：




$N\times N$方阵$A$的第$n$对角线之和为：

$$S(n)=\sum_{i=1}^{N}a_{i,i\oplus n}$$

其中，

$$i\oplus n=\begin{cases}i+n, & \text{当}i+n\le N\\ (i+n)\%N, & \text{当}i+n>N\end{cases}$$

**要求：**

**方阵大小固定为10×10；**

**方阵元素如下：**

**使用指针编程。**

$$A=\begin{bmatrix}0 & 1 & \dots & 9\\ 10 & 11 & \dots & 19\\ \dots & \dots & \dots & \dots\\ 80 & 81 & \dots & 89\\ 90 & 91 & \dots & 99\end{bmatrix}_{10\times 10}$$

3.【emoji 表情转义符识别】通常一个基于转义符的 emoji 表情输入由以下三部分构成：

〈转义符＋表情名称＋终止符〉

以新浪微博为例，当微博正文读取到一个转义符“[”时，它与终止符“]”之间的文字将作为表情名称在表情库中进行搜索，如果存在匹配表情，则输出显示。

注意，如果在一段语句中存在多个转义符和一个终止符，那么以离终止符最近的一个转义符作为表情的起始标志。

【要求】编写程序，在输入的一句文字中，输出被转义符括起来的表情名称文字。

【输入】首先输入转义符，然后输入终止符，二者均为临时指定的任意半角标点符号，随后输入一行任意的文字，由英文、数字和符号组成。文字最长不超过 140 个字符。

【输出】被转义符括起来的表情名称文字，如有多个表情名称，则分行输出。输入输出样例如下：

转义符：*

终止符：#

输入文字：Time for lunch. *greedy# Hope a big meal.

输出：greedy

多个转义符以距离终止符最近的一个为准：

输入文字：*happy*smile#

输出：smile

# 课后练习选做题2道

【Zig-Zag 扫描】在图像的基于 DCT 变换的压缩中，通常对 DCT 变换后的系数矩阵进行 Zig-Zag 扫描。所谓 Zig-Zag 扫描，又名“之”字型扫描，即从矩阵的第一行第一列系数开始，按照“之”字形方向进行系数读取。以下方所示 3 阶矩阵为例，经过 Zig-Zag 扫描后，输出数据顺序为{1,2,4,7,5,3,6,8,9}。

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \{1,2,4,7,5,3,6,8,9\}$$

【要求】编写程序，输入任意阶矩阵，按照 Zig-Zag 方式输出数据。

【输入】首先输入一个整数 N，代表矩阵阶数，随后按照从左到右，从上到下的顺序输入矩阵数据。矩阵数据类型不做要求，可以使用 int 类型。

【输出】可以按行输出数据，数字之间用空格隔开。也可以按列输出，每行一个数字。

假设将下列程序生成可执行文件 test.exe,使用命令行： test FINAL EXAM，则程序的输出结果是 _____。

```
#include   <stdio.h>
int    main(int argc, char *argv[])
{      char **p;
       for(p=argv; argc--; p++)
            printf("%c%s",**p,*p);
       return 0;
}
```