

# 1.初识指针之遍历数组

定义一维整型数组x，并输入数组元素个数N。要求使用指针运算的方法遍历数组并对数组各个元素进行赋值及输出操作。要求使用如下公式给数组各个元素赋值： $x[i] = 600 * \cos(i * 0.35)$

输入

输入一个整数N ( $0 < N < 1000$ )。

输出

逐个输出数组中的N个元素，邻近两数之间用一个空格隔开。

```
#include <stdio.h>
#include <math.h>
int main() {
    int N, x[1000], *p;
    scanf("%d", &N);
    for (p = x; p < x + N; p++) {
        *p = 600 * cos( (p - x) * 0.35 );
    }
    for (p = x; p < x + N; p++) {
        printf("%d ", *p);
    }
}
```

## 2.初识指针之字符串小写判断

输入一个不含空格的字符串，要求使用指针的方法遍历该字符串。并判断字符串是否全部为小写英文字母，是则输出“TRUE”，否则输出“FALSE”（不输出引号）。

输入

输入一个不含空格的字符串。输入不多于999个字符，以回车结束。

输出

根据题意输出“TRUE”或“FALSE”（不输出引号）。

```
#include <stdio.h>
int main() {
    char str[1000], *p = str;
    int isAllLower = 1;
    scanf("%999s", str);
    while (*p != '\0') {
        if (!(*p >= 'a' && *p <= 'z')) {
            isAllLower = 0;
            break;
        }
        p++;
    }
    printf("%s", isAllLower ? "TRUE" : "FALSE");
}
```

### 3初识指针之字符串统计空格

输入一个字符串。要求使用指针的方法遍历该字符串，并统计字符串中空格的个数。

输入

输入一个字符串，不多于999个字符，以回车结束。

输出

输出字符串中空格的个数。

```
#include <stdio.h>
int main() {
    char str[1000], *p;
    int space_count = 0;
    scanf("%999[^\n]", str);
    p = str;
    while (*p != '\0') {
        if (*p == ' ') {
            space_count++;
        }
        p++;
    }
    printf("%d", space_count);
}
```

## 4.初识指针之字符串统计字母

输入一个字符串。要求使用指针的方法遍历该字符串，并统计字符串中英文字母的个数。

输入

输入一个字符串，不多于999个字符，以回车结束。

输出

输出字符串中英文字母的个数。

```
#include <stdio.h>
int main() {
    char str[1000], *p;
    int count = 0;
    if (scanf("%999[^\n]", str) != 1) {
        str[0] = '\0';
    }
    for (p = str; *p != '\0'; p++) {
        if ((*p >= 'A' && *p <= 'Z') || (*p >= 'a' && *p <=
'z')) {
            count++;
        }
    }
    printf("%d", count);
}
```

其实有个内置函数可以直接解决

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char str[1000], *p;
    int count = 0;
    if (scanf("%999[^\n]", str) != 1) {
        str[0] = '\0';
    }
    for (p = str; *p != '\0'; p++) {
        if (isalpha(*p)) {
            count++;
        }
    }
}
```

```
    }  
    printf("%d", count);  
}
```

## 5.使用指针判断字符串长度

输入一个字符串。要求使用指针的方法遍历该字符串，并统计该字符串的长度。不允许使用 `strlen()` 函数。

输入

输入一个字符串，不多于999个字符，以回车结束。

输出

输出字符串的长度。

```
#include <stdio.h>  
int main() {  
    char str[1000], *p;  
    int count = 0;  
    if (scanf("%999[^\n]", str) != 1) {  
        str[0] = '\0';  
    }  
    for (p = str; *p != '\0'; p++) {  
        if(p) count++;  
    }  
    printf("%d", count);  
}
```

## 6.使用指针保持队形

使用指针的方法，将一个整数插入到由小到大排列的数列中，使得插入后仍然保持由小到大的顺序。

### 输入

输入包含两行：

第一行是10个由小到大排好序的整数，邻近两数之间用一个空格隔开。

第二行是需要插入的新整数m。

### 输出

输出插入数字后，由小到大排好序的11个整数，邻近两数之间用一个空格隔开。

```
#include <stdio.h>
int main() {
    int arr[11], m, *p, *current, *tail;
    for (p = arr; p < arr + 10; p++) {
        scanf("%d", p);
    }
    scanf("%d", &m);
    tail = arr + 10;
    current = arr + 9;
    while (current >= arr && *current > m) {
        *tail-- = *current--;
    }
    *tail = m;
    for (p = arr; p < arr + 11; p++) {
        printf("%d ", *p);
    }
}
```

```
}  
}
```

## 7.指针与一维动态数组B

定义指向float类型的指针变量 $ptr$ ，并根据输入的数组元素个数 $N$ 从堆中申请内存，使得该动态分配的内存可以存储 $N$ 个float类型的元素；然后使用如下公式给数组各个元素赋值：

$$x[i] = 100 * \cos(i * 0.01)$$

数组元素赋值后，计算并输出数组元素之和。小数点后必须保留2位有效数字（四舍五入），不足补零。另外，请不要忘记释放该内存。

输入

输入一个整数 $N$  ( $1 \leq N \leq 10000$ )。

输出

输出数组元素之和，小数点后必须保留2位有效数字（四舍五入），不足补零。

```
#include <stdio.h>  
#include <math.h>  
int main() {  
    float x[10000], *ptr, sum=0;  
    int i, N;  
    scanf("%d", &N);  
    for (i = 0; i < N; i++) {  
        x[i] = 100*cos(i*0.01);  
    }  
    for(ptr = x; ptr < x + N; ptr++) {
```

```
        sum += *ptr;
    }
    printf("%.2f", sum);
}
```

## 8.指针与二维数组扒洋葱皮问题A

定义二维整型数组 $a[4][4]$ ，要求使用指针的方法对数组元素从M开始逐个赋值（每次递增1），然后将二维数组 $a[4][4]$ 的外层元素扒掉，形成新的二维数组 $b[2][2]$ ，最后输出二维数组 $b[2][2]$ 上的每个元素。要求输出为2行，且每行的两个元素之间用1个空格隔开。

例如，M为10，则数组 $a[4][4]$ 的各个元素为：

$$\begin{pmatrix} 10 & 11 & 12 & 13 \\ 14 & 15 & 16 & 17 \\ 18 & 19 & 20 & 21 \\ 22 & 23 & 24 & 25 \end{pmatrix}$$

扒掉最外层元素后的二维数组 $b[2][2]$ 的各个元素为：

$$\begin{pmatrix} 15 & 16 \\ 19 & 20 \end{pmatrix}$$

输入

输入一个整数M，且 $9 < M < 84$ 。

输出

输出二行，分别是二维数组 $b[2][2]$ 上的每个元素。每行的两个元素之间用1个空格隔开。



```

#include <stdio.h>
int main() {
    int M,a[4][4],*p,b[2][2],*pb,*row,i;
    scanf("%d", &M);
    p = (int *)a;
    for (i = 0; i < 16; i++) {
        *p++ = M + i;
    }
    pb = (int *)b;
    for (row = *(a + 1) + 1; row <= *(a + 2) + 1; row += 4)
    {
        *pb++ = *row;
        *pb++ = *(row + 1);
    }
    printf("%d %d\n", b[0][0], b[0][1]);
    printf("%d %d\n", b[1][0], b[1][1]);
}

```

## 9.初识指针数组之月份问题

定义指针数组char \*ptr[12]，分别指向12个月份的英文单词缩写。输入月份值n，输出该月份的对应该英文单词缩写；若n不在1~12之间，则输出“Error”(不输出引号)。

提示：12个月份的英文单词缩写如下，注意首字母大写。

[Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sept,Oct,Nov,Dec]

输入

输入一个整数n。

输出

输出n对应的月份英文单词缩写，或输出字符串“Error”(不输出引号)。

```
#include <stdio.h>
int main() {
    char *months[12] = {
        "Jan", "Feb", "Mar", "Apr", "May", "Jun",
        "Jul", "Aug", "Sept", "Oct", "Nov", "Dec"
    };
    int n;
    scanf("%d", &n);
    if (n >= 1 && n <= 12) {
        printf("%s\n", months[n-1]);
    } else {
        printf("Error\n");
    }
}
```

## 10. 指针变量之函数参数作返回值A

大家都知道函数返回值可以返回数据给调用者，也有必要掌握使用指针变量作为函数参数充当函数返回值的方法。请你帮助叮当定义形如下面的函数：

```
void fun(int a, int b, int sum, int *diff, int *product,
float *divide)
```

在主函数中，输入两个整数a和b，然后调用此函数计算数学上的a + b、a - b、a \* b和a / b四种运算的值，四种运算的结果通过函数的后四个参数返回主函数；最后在主函数中，输出这四种运算的值。除法运算要求小数点后必须保留2位有效数字（四舍五入），不足补零。

输入

输入两个整数a和b，用一个空格隔开。

## 输出

按照题目要求输出 $a + b$ 、 $a - b$ 、 $a * b$ 和 $a / b$ 的值，邻近的数值之间用一个空格隔开。除法运算要求小数点后必须保留2位有效数字（四舍五入），不足补零。

```
#include <stdio.h>
void fun(int a, int b, int *sum, int *diff, int *product,
float *divide){
    *sum = a + b;
    *diff = a - b;
    *product = a * b;
    *divide = a / (float)b;
}
int main() {
    int a, b, sum, diff, product;
    float divide;
    scanf("%d%d", &a, &b);
    fun(a, b, &sum, &diff, &product, &divide);
    printf("%d %d %d %.2f", sum, diff, product, divide);
}
```