Compilation of C Program Questions — **Pointer（指针）**

by Jaser Li (HDU) Dec. 2023

* 行指针做函数参数

找出任意的一个m×n矩阵每一行上的最大值存放在数组max中，并在主函数显示出来。

#include <stdio.h>

#define N 20

void max\_row(int (\*p)[N], int m, int n, int \*max);

int main(void)

{

    int i, j, juZhen[N][N];

    int m, n ,max[N];

    scanf("%d%d", &m, &n);

    for (i = 0; i < m; i++)

        for (j = 0; j < n; j++)

            scanf("%d", &juZhen[i][j]);

① max\_row(juZhen,m,n,max); // 调用 max\_row 函数

    for(i=0; i<m; i++)

        printf("The max in line %d is: %d\n", i+1, ② max[i] );

    return 0;

}

void max\_row(int (\*p)[N], int m, int n, int \*max)

{

    int i, j;

    for (i=0; i<m; i++, p++, max++)

    {

        \*max = \*\*p;

        for (j=0; j<n; j++)

        {

            if (③ \*(\*p+j)>\*max )

                ④ \*max=\*(\*p+j);

// 外层循环中已有p++和max++，故不需要在指针运算中使用i来增加偏移量。

        }

    }

}

* 语句③④分步骤解释如下：

if(\*(\*p+j)>\*max)用于比较当前考察的元素是否大于当前已知的该行的最大值。

\*(\*p+j)：这是一个**双重间接引用**。首先，p是一个指向数组的指针，其中每个数组都有N个元素（这里的N是一个预定义的宏，代表列数的最大可能值）。因为p在函数调用时指向juZhen的第i行（外层循环中有p++，所以它会逐行移动），所以\*p就是对i行的引用。然后，\*p+j是获取i行中第j个元素的地址，\*(\*p+j)是取出那个地址上的值，即i行j列的元素值。

\*max：由于max是指向max数组当前元素的指针（同样地，由于外层循环中有max++，它会随着每次迭代逐个移动），\*max就是当前行的最大值的引用。

如果当前元素的值大于当前已知的最大值，则执行下面的语句。

\*max=\*(\*p+j);更新当前行的最大值。

\*max：和上面一样，它是max数组当前元素的引用，也就是当前行的最大值的存储位置。

\*(\*p+j)：这是当前正在检查的元素的值。

这条语句将当前元素的值赋给\*max，从而更新当前行的最大值。

总结来说，这两行代码遍历二维数组的每一行，不断比较并更新每行的最大值，最终将每行的最大值存储在max数组中。

* 求最大数——用指针访问数组

从键盘输入10个整数，求出其中的最大数。要求用指针变量访问数组的元素。

#include <stdio.h>

int main(void)

{

int a[10],max,① \*p ;

for(p=a;② p<a+10 ;p++)

scanf("%d",p);

③ max=\*a;

for(p=a+1;④ p<a+10 ;p++){

if(\*p>max)

max=\*p;

}

printf("max=%d\n",max);

return 0;

}

* 字符串的连接

使用指针方式实现两个字符串的连接(不能使用strcat函数），并将连接后的字符串输出。

输入格式:

分别输入两行以回车结束的非空字符串（不超过40个字符）。

输出格式:

一行输出连接后新的字符串。

输入样例:

Beijing\_

China

输出样例:

Beijing\_China

#include <stdio.h>

int **main**(){

    char str1[41],str2[41];

    char \*p1 = str1, \*p2 = str2;

**scanf**("%s",str1);

**scanf**("%s",str2);

    while(\*(++p1)); // 确定str1的末位置

    while(\*p2) // 将str2依次接到str1后面

        \*(p1++)=\*(p2++);

    \*p1='\0'; // 补上字符串结束符'\0'

**printf**("%s",str1);

}

* 实数排序

本题要求编写程序，输入n个实数，使用指针引用的方式将它们按从大到小的顺序排列。

输入格式:

输入第一行给出一个正整数n（2≤n≤10）,输入第二行给出n个实数，其间以空格分隔。

输出格式:

输出从大到小排好序的n个数（保留2位小数），每个数之间空一格，行末没有空格。

输入样例:

5

3.2 5.4 6.12 2.51 4.23

输出样例:

6.12 5.40 4.23 3.20 2.51

# include<stdio.h>

# include<stdlib.h>

int **main**(){

    int n;

    double r;

    double \*a=(double \*)**malloc**(n \* sizeof(double));

**scanf**("%d",&n);

    for(int t=0;t<n;t++)

**scanf**("%lf",&a[t]);

    for(int i=0;i<n-1;i++){

        for(int j=0;j<n-i-1;j++){

            if(a[j]<a[j+1]){

                r=a[j];a[j]=a[j+1];a[j+1]=r;

            }

        }

    }

    for(int k=0;k<n;k++){

**printf**("%.2f",a[k]);

        if(k<n-1) **printf**(" ");

    }

**free**(a);

}

* swap函数

从键盘任意输入两个整数，用指针作函数参数，编程实现将其交换后再重新输出。

函数接口:

void Swap(int \*x, int \*y)

输入格式:

空格分隔的两个整数

输出格式:

空格分隔的两个整数

输入样例:

32 90

输出样例:

90 32

# include<stdio.h>

# include<stdlib.h>

void **Swap**(int \**x*,int \**y*){

    int r=\**x*;

    \**x*=\**y*;

    \**y*=r;

}

int **main**(){

    int a,b;

**scanf**("%d %d",&a,&b);

**Swap**(&a,&b);

**printf**("%d %d",a,b);

}

* 计算指针字符串长度

从键盘任意输入一个字符串，用指针的方式计算其实际字符个数并打印出来，即不使用字符串处理函数strlen（）编程实现strlen的功能。

函数接口:

unsigned int MyStrlen(const char \*pStr)

输入格式:

字符串

输出格式:

字符串长度

输入样例:

hello world

输出样例:

11

# include<stdio.h>

# include<stdlib.h>

unsigned int **MyStrlen**(const char \**pStr*){

    unsigned int len=0;

    while((\**pStr*++)!='\0'){

        len+=1;

    }

    return len;

}

int **main**(){

    char s[1000];

**fgets**(s,sizeof(s),**stdin**);

**printf**("%u",**MyStrlen**(s));

    return 0;

}

* 删除数列中的指定数

请使用指针的方法编写程序，程序的功能是先输入10个整数存储到数组a中，再输入一个指定的数x，把数组中的x数据删除掉，并保证数组中剩余的数还是连续存储的。

比如输入10个整数：3 2 8 6 5 8 7 9 8 5，存入数组，然后输入一个数8，把数组中的8全部删除。最后数组中剩下的数：3 2 6 5 7 9 5，在数组a中还是连续存储的。

要求定义函数实现删除功能：int del\_num(int \*p, int n, int x)；其中函数的返回值为删除指定数后数组中剩余数据的个数。

输入格式:

输入10个整数，和1个指定数。

输出格式：

输出剩余数，以空格隔开，最后一个数后面没有空格。

输入样例：

3 2 8 6 5 8 7 9 8 5

8

输出样例：

3 2 6 5 7 9 5

#include <stdio.h>

int **del\_num**(int \**p*, int *n*, int *x*){

    int i,j;

    for(i=0,j=0;i<*n*;i++){

        if(\*(*p*+i)!=*x*){

            \*(*p*+j)=\*(*p*+i);

            j++;

        }

    }

    return j;

}

int **main**(){

    int a[10],x,n=0;

    for(int i=0;i<10;i++)

**scanf**("%d",&a[i]);

**scanf**("%d",&x);

    int \*p=a;

    int len=**del\_num**(p,10,x);

    for(int j=0;j<len;j++)

**printf**("%d%c",a[j],(j==len-1)?'\n':' ');

}

* 求一组数中的最大值、最小值及总和

题目内容：

求一组数中的最大值、最小值及总和。要求定义函数

int f(int a[],int n,int \*pmax,int \*pmin)

函数返回总和，并分别通过指针pmax 和 pmin将最大值、最小值赋值给主函数中的对应变量。

输入格式:

第1行为正整数n（n>2）,表示一组数的个数 第2行包含n个整数，用空格分隔。

输出格式：

一行包含3个整数，分别为最大值、最小值及总和

输入样例：

10

2 5 4 8 6 9 1 3 7 0

输出样例：

9 0 45

#include <stdio.h>

int **f**(int *a*[],int *n*,int \**pmax*,int \**pmin*){

    int sum=0,\*p=*a*;

    for(int i=0;i<*n*;i++){

        if(\*(p+i)>\**pmax*) *pmax*=p+i;

        if(\*(p+i)<\**pmin*) *pmin*=p+i;

        sum+=\*(p+i);

    }

**printf**("%d %d %d",\**pmax*,\**pmin*,sum);

}

int **main**(){

    int n;

**scanf**("%d",&n);

    int a[n];

    for(int i=0;i<n;i++)

**scanf**("%d",&a[i]);

**f**(a,n,a,a); // 注意第一个参数a指的是数组a的首地址

}