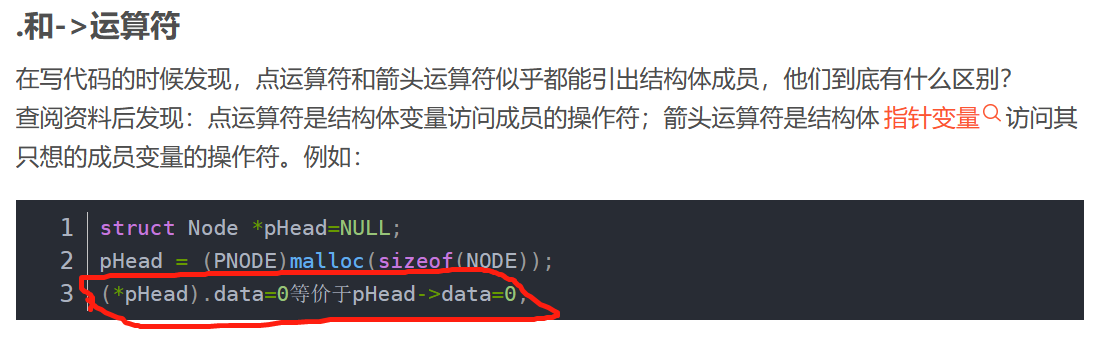
第五题



1001 Apple 300.5 2.0

1002 Banana 100.5 3.5

1003 Lemon 205.2 2.5

1004 cherry 70.65 16.2

1005 grape 500.62 6.6

1006 orange 10000.2 5.15

/\*定义农副产品信息的结构体 fruitinfo，包括产品编号（正整数）、产品名称（由不含空格的 26 个英文字母的大小写构成，不超过 20 个字符）、销量及单价

其中销量和单价是浮点数\*/

#include <stdio.h>

struct fruitinfo

{

int bianhao;

char name[21];

float num,price;

}t[6];

//定义 void set( struct fruitinfo \*t )函数，用来输入一种农副产品的数据；

void set( struct fruitinfo \*t )

{

scanf("%d%s%f%f",&t->bianhao,&t->name,&t->num,&t->price);

}

//定义 void print( struct fruitinfo \*t )函数用来输出一种农副产品的数据（其中浮点数保留小数点后 2 位)

void print( struct fruitinfo \*t )

{

printf("%d %s %.2f %.2f",t->bianhao,t->name,t->num,t->price);

}

//定义 void sortfruit( struct fruitinfo \*t )函数，用来对所有农副产品按照销售额（销售额等于单价乘以销量）

//从小到大排序（销售额相同时保持原来顺序）

void sortfruit( struct fruitinfo \*t )

{

float a[6],temp\_a;

int i,j,k;

struct fruitinfo temp;

for(i=0;i<6;i++)

{

a[i]=t[i].num\*t[i].price; //这里用数组运算比较方便

}

for(i=0;i<5;i++) //选择排序！

{

k=i;

for(j=i+1;j<6;j++)

if(a[k]>a[j]) k=j;

if(i!=k)

{

temp=t[i];

t[i]=t[k];

t[k]=temp;

temp\_a=a[i]; //这里的a值也要交换

a[i]=a[k];

a[k]=temp\_a;

}

}

}

int main()

{

int i;

for(i=0;i<6;i++) set(&t[i]);

sortfruit(t);

for(i=0;i<6;i++)

{

print(&t[i]);

printf("\n");

}

return 0;

}

第一题

/\*请利用嵌套循环输出平行四边形，平行四边形行数为 n，每行的"\*"的数目也为 n

n由键盘输入\*/

#include <stdio.h>

int main()

{

int n,i,j,k;

scanf("%d",&n);

for(i=1;i<=n;i++)

{

for(j=1;j<=i;j++) printf(" ");

for(k=1;k<=n;k++) printf("\*");

printf("\n");

}

return 0;

}

第二题

/\*输入一个 3\*5 矩阵，求出每一行每一列的最小值，将这些最小值中的最大值输出。

例如行和列的最小值为 11 1 6 1 2 3 4 5，这 8 个最小值中 11 最大 \*/

#include <stdio.h>

int main()

{

int i,j,a[4][6],max1,max2,min1,min2;

for(i=0;i<3;i++)

{

for(j=0;j<5;j++)

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

}

for(i=0;i<3;i++)

{

min1=a[i][0];

for(j=1;j<5;j++)

{

if(a[i][j]<min1) min1=a[i][j]; //找出每一行的最小值

}

a[i][5]=min1; //把最小值填在那一行的最后一个

}

for(j=0;j<5;j++)

{

min2=a[0][j];

for(i=1;i<3;i++)

{

if(a[i][j]<min2) min2=a[i][j]; //找出每一列的最小值

}

a[3][j]=min2; //把最小值填在那一列的最后一个

}

max1=a[0][5]; //找出每一行最小值的最大值 这个max的位置在循环外面

for(i=1;i<3;i++)

{

if(a[i][5]>max1) max1=a[i][5];

}

max2=a[3][0];

for(j=1;j<5;j++)

{

if(a[3][j]>max2) max2=a[3][j];

}

if(max1>max2) printf("%d",max1);

else printf("%d",max2);

return 0;

}

简便方法：用一维数组

#include <stdio.h>

int main()

{

int i,j,a[3][5],max,min1,min2;

int b[8];

int p=0;

for(i=0;i<3;i++)

{

for(j=0;j<5;j++)

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

}

for(i=0;i<3;i++)

{

min1=a[i][0];

for(j=1;j<5;j++)

{

if(a[i][j]<min1) min1=a[i][j]; //找出每一行的最小值

}

b[p]=min1; //把最小值填存到数组

p++;

}

for(j=0;j<5;j++)

{

min2=a[0][j];

for(i=1;i<3;i++)

{

if(a[i][j]<min2) min2=a[i][j]; //找出每一列的最小值

}

b[p]=min2; //把最小值存到数组

p++;

}

max=b[0]; //找出数组里边的最大值

for(i=1;i<=8;i++)

{

if(max<b[i])

{

max=b[i];

}

}

printf("%d",max);

return 0;

}

第三题

/\*编写函数将整数 k 变换为一个字母串，规则如下：设 0 对应 A，1 对应 B，......，25 对应Z。

从左向右考察 k 的各位数字，若连续两位对应上面一种变换，则直接换掉两位数字，否则一位数字换成对应的字母。变换结果存入 res 反馈给调用者。

函数原型为：void DigiTran( int k, char res[] ); 在主函数输入一些非负整数，最后输入-1 代表结束，利用上面函数对每个数字变换后输出 \*/

#include <stdio.h>

思路：从个位开始存入数组，调用数组数据的时候从后往前调用

void DigiTran( int k, char res[] )

{

int i=0,p=0,m,a[100];

if(k>0)

{

while(k>0) //k=0的时候循环结束

{

a[i]=k%10;

i++;

k=k/10;

}

while(i>1) //开始执行循环时，i的值=k的位数+1

{

m=10\*a[i-1]+a[i-2];

if(m>=10&&m<=25)

{

res[p]=(char)(m+'A');

p++;

i=i-2;

}

else

{

res[p]=(char)(a[i-1]+'A');

p++;

i--;

}

}

}

Else //k有可能是0

{

res[0]='A';

}

}

int main()

{

int k;

char res[200];

while(1) //永远循环，直到break

{

scanf("%d",&k);

if(k==-1) break;

else

{

memset(res,0,sizeof(res)); //清空res，防止残留数据影响下一次循环

DigiTran(k,res);

printf("%s",res);

printf(" ");

}

}

return 0;

}

第四题

/\*编写一个函数 int sum(int \*p, int n)计算一组数据的和，在 main 函数中输入 5 个 int 型数并放入数组

调用 sum 函数计算并输出该 5 个数的和。\*/

#include <stdio.h>

int sum(int \*p, int n)

{

int i;

n=0;

for(i=0;i<n;i++)

{

n=n+\*(p+i);

}

return n;

}

int main()

{

int i,a[5],s,\*p;

p=a; //指针引用数组的方法

for(i=0;i<5;i++)

scanf("%d",&\*(p+i));

sum(p,n);

printf("%d",n);

return 0;

}