**2022年程序设计基础7**

**一．单选题**

1. 数组定义为 int a[3][2]={1, 2, 3, 4, 5, 6}，数组元素（B ）的值为6。

A.a[3][2] B. a[2][1] C.a[1][2] D.a[2][3]

2.在int a[ ][3] = { {1,4}, {3,2}, {4,5,6}, {0} };中，a[2][1]的值是（A ）。

A.5 B.0 C.6 D.2

3．有两个字符数组a,b，则以下正确的输入语句是（B）

A.gets(a,b); B.scanf("%s%s",a,b);

C.scanf("%s%s",&a,&b); D.gets("a"),gets("b");

4．以下对一维数组a的正确说明是（ D）。

A.char a（10）; B.int a[ ];

C.int k＝5，a[k]; D.char a[3]={‘a’,’b’,’c’};

5.下列数据中属于C语言的“字符串常量”的是（ A ）。

A.“a” B.{ABC}   C.‘abc\0’ D.‘a’

6.下列定义数组正确的是（B）

A.double a[6] = (1,2,3,4,5,6); B.int b[5] = {0};

C.float c[5] = {1,2,3,4,5,6}; D.char d[5] = {"China"};

**二、程序填空题**

1.计算二维数组第一列元素之和

从键盘输入一个4行4列的二维数组元素，求第一列元素之和。

#include <stdio.h>

#include <math.h>

int main( )

{

int i, j ; 2分

int a[4][4],sum;

sum= 0 ; 2分

for(i=0; i<4;i++)

for(j=0;j<4;j++)

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

for(i=0;i<4;i++)

sum+=a[i][0]; 2分

printf("%d\n",sum);

return 0;

}

2.求二维数组中奇数元素之和

下面这段程序是计算并打印一个二维数组（数组的数组）中值为奇数的元素之和。题目保证输入的元素均为绝对值不超过10000的整数。请填写空缺的代码。

#include <stdio.h>

int sumOdd(int (\*array)[5], int row){

int sum;

sum = 0 ; 2分

for(int i=0; i<row; i++)

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

if( array[i][j]%2 ) sum += array[i][j] ; 4分

return sum;

}

int main(){

int array[4][5];

for(int i=0; i<4; i++)

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

scanf("%d", &array[i][j] ); 2分

printf("%d\n", sumOdd( array, 4 ) ); 2分//注意：所填的空之外两端均已有括号，只需填写函数的参数

return 0;

}

3．在一个字符数组中查找一个指定的字符

在一个字符数组中查找一个指定的字符，若数组中含有该字符则输出该字符在数组中第一次出现的位置（下标值）否则输出-1。

#include<stdio.h>

void main()

{

char ch='a',cstr[50];

int inum,ii,iflag=1;

gets(cstr);

inum= strlen(cstr) ; 2分

for(ii=0;ii<inum;ii++)

if( ch==cstr[ii] ) {2分

iflag=0;

break;

}

if(iflag==1)

printf("%d",-1);

else

printf("%d",ii) ; 2分

}

4.对角线元素之和

下面程序可求出矩阵a的主对角线上的元素(行标和列标相等的元素)之和.

程序写在这里。下面的空分值是2分。

例如：

#include <stdio.h>

int main() {

int a[3][3]={1,3,5,7,9,11,13,15,17},sum=0,i,j;

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

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

if( i==j ) 2分

sum=sum+a[i][j];

printf("sum=%d",sum);

return 0;

}

5.学生成绩按总分排序

从键盘输入*n*(*n*≤10)个同学的三门功课成绩，调用排序函数按成绩总分由高到低进行排序，并输出排序后的结果。

#include <stdio.h>

typedef struct student

{

char name[20];

float score[3];

}STUDENT;

void swap ( STUDENT \*p, STUDENT \*q ) 2分

{

STUDENT t;

t = \*p;

\*p = \*q ; 2分

\*q = t;

}

void sortStu(STUDENT \*s, int n)

{

int i, j, k, ind;

float sum, sum1;

for (i = 0; i < n - 1; i++)

{

ind = i;

sum = 0;

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

sum += s[i].score[j] ; 2分

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

{

sum1 = 0;

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

sum1 += s[j].score[k];

if ( sum1 > sum ) 2分

ind = j;

}

if (ind != i)

swap ( s + i, s + ind ); 2分

}

}

int main()

{

STUDENT stu[10];

int n, i;

scanf("%d", &n);

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

scanf("%s %f %f %f", stu[i].name, &stu[i].score[0], &stu[i].score[1], &stu[i].score[2]);

sortStu(stu, n);

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

printf("%s %.2f %.2f %.2f\n", stu[i].name, stu[i].score[0], stu[i].score[1], stu[i].score[2]);

return 0;

}

### 输入样例

|  |
| --- |
| 3  Liming 83 92 77  Zhaolin 98 75 88  Hanxu 76 82 63 |

### 输出样例

|  |
| --- |
| Zhaolin 98.00 75.00 88.00  Liming 83.00 92.00 77.00  Hanxu 76.00 82.00 63.00 |

**三、编程题**

**1.藏头诗**

本题要求编写一个解密藏头诗的程序。

注：在 2022 年 7 月 14 日 16 点 50 分以后，该题数据修改为 UTF-8 编码。

### 输入格式：

输入为一首中文藏头诗，一共四句，每句一行。注意：一个汉字占三个字节。

### 输出格式：

取出每句的第一个汉字并连接在一起形成一个字符串并输出。同时在末尾输入一个换行符。

### 输入样例：

|  |
| --- |
| 一叶轻舟向东流  帆稍轻握杨柳手  风纤碧波微起舞  顺水任从雅客流 |

### 输出样例：

|  |
| --- |
| 一帆风顺 |

**参考代码：**

|  |
| --- |
| **#include <stdio.h>**  **int main(){**  **int i;**  **char poem[4][22];**  **for (i=0; i<4; i++)**  **gets(poem[i]);**  **char str[13];**  **int k = 0;**  **for (i=0; i<4; i++){**  **str[k++] = poem[i][0];**  **str[k++] = poem[i][1];**  **str[k++] = poem[i][2];**  **}**  **puts(str);**  **return 0;**  **}** |

**2.数组元素移动**

完成数组元素的移动功能：假设数组有n个元素，输入一个数x，把数组的第x个位置的元素先保存起来，然后把x+1到n的元素，依次往前移一位，最后将原来的第x个位置的元素放在数组的最后。  
重复若干次这样的移动，得到最后的结果。

### 输入格式:

第一行包括一个整数n（1<=n<=100），表示数组元素的个数。  
第二行输入n个数组元素，均为整数，用空格隔开。  
第三行输入一个数k（1<=k<=100），表示要进行k次移动。  
接下来k行，每行一个数x，表示要移动第x个元素。

### 输出格式:

输出经过k次移动后的数组，每两个元素之间用空格隔开。

### 输入样例:

|  |
| --- |
| 10  1 2 3 4 5 6 7 8 9 10  4  4  3  2  5 |

### 输出样例:

|  |
| --- |
| 1 5 6 7 9 10 4 3 2 8 |

**参考代码：**

|  |
| --- |
| **int main(){**  **int n,i,k,x,j;**  **int arr[100];**  **scanf("%d", &n);**  **for (i = 0;i < n;i++) scanf("%d", &arr[i]);**  **scanf("%d", &k);**  **for (;k>0;k--) {**  **scanf("%d", &x);**  **j = arr[x - 1];**  **for (i = x ;i < n;i++) arr[i - 1] = arr[i];**  **arr[n-1] = j;**  **}**  **for (i = 0;i < n;i++) {**  **if(i==0) printf("%d",arr[i]);**  **else printf(" %d", arr[i]);**  **}**  **return 0;**  **}** |

**3.切割绳子**

有*N*条绳子，它们的长度分别为*Li*​,*i*=1,⋯,*N*。如果从它们中切割出*K*条长度相同的绳子，这*K*条绳子每条最长能有多长？

### 输入格式:

第一行两个整数*n*和*k*（1<*n*<10000;1<*k*<10000）。  
接下来*n*行，描述了每条绳子的长度*Li*​，*Li*​也是整数。

### 输出格式:

切割后每条绳子的最大长度。

### 输入样例:

|  |
| --- |
| 4 11  802  743  457  539 |

### 输出样例:

|  |
| --- |
| 200 |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **int max\_f(int a[],int n){**  **int i;**  **int temp=a[0];**  **for(i=1;i<n;i++) if(a[i]>temp) temp=a[i];**  **return temp;**  **}**  **int main() {**  **int n, li,max, temp,i,j;**  **scanf("%d %d",&n,&li);**  **int a[n];**  **for(i=0;i<n;i++) scanf("%d",a+i);**  **max=max\_f(a,n);**  **for(i=max;i>0;i--) {**  **temp=0;**  **for(j=0;j<n;j++) temp+=a[j]/i;**  **if(temp==li)**  **break;**  **}**  **printf("%d\n",i);**  **return 0;**  **}** |

**4．输出Fibonacci数列中两项之和**

#### 任务描述

输出Fibonacci数列中任意两项之和。

#### 输入格式:

两个整数a,b(大于0小于 等于60的正整数)，以空格分隔。

#### 输出格式:

输出Fibonacci数列中第a项和第b项之和。

#### 输入样例:

|  |
| --- |
| 4 6 |

#### 输出样例:

|  |
| --- |
| 11 |

#### 输入样例:

|  |
| --- |
| 20 30 |

#### 输出样例:

|  |
| --- |
| 838805 |

#### 输入样例:

|  |
| --- |
| 56 52 |

#### 输出样例:

|  |
| --- |
| 258802713816 |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int a,b,i,t;**  **long f[60]={1,1};**  **scanf("%d%d",&a,&b);**  **if(a>b){t=a;a=b;b=t;}**  **for(i=2;i<b;i++) f[i]=f[i-1]+f[i-2];**  **printf("%ld\n",f[a-1]+f[b-1]);**  **return 0;**  **}** |

**5． 打印“杨辉三角“ 品中国数学史 增民族自豪感**

**背景介绍：** 北宋人贾宪约1050年首先使用“贾宪三角”进行高次开方运算。

南宋数学家杨辉在《详解九章算法》（1261年）记载并保存了“贾宪三角”，故称杨辉三角。杨辉三角是中国数学史上的一个伟大成就。  
杨辉三角，是中国古代数学的杰出研究成果之一，它把二项式系数图形化，把组合数内在的一些代数性质直观地从图形中体现出来，是一种离散型的数与形的结合。

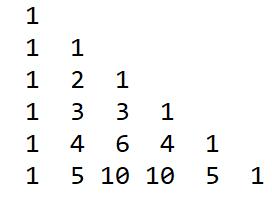
中国南宋数学家杨辉1261年所著的《详解九章算法》一书中出现。在欧洲，帕斯卡（1623----1662）在1654年发现这一规律，所以这个表又叫做帕斯卡三角形。帕斯卡的发现比杨辉要迟393年，比贾宪迟600年。

**杨辉三角数字的特点为：**

（1）在三角形的首列和对角线上，数值均为1；

（2）其余数据为：每个数字等于上一行的左右两个数字之和，第n+1行的第i个数等于第n行的第i-1个数和第i个数之和，用公式表示为： C(n+1,i)=C(n,i)+C(n,i-1)。

图示为：



**杨辉三角的应用：**(a+b)的n次方，展开式中的各项系数依次对应杨辉三角的第(n+1)行中的每一项。

### 输入格式:

欲打印杨辉三角的行数n（1<=n<=13)。

### 输出格式:

每个数字占据4个字符的位置，数字左对齐，数字不足4位的右边留出空格。

### 输入样例:

|  |
| --- |
| 13 |

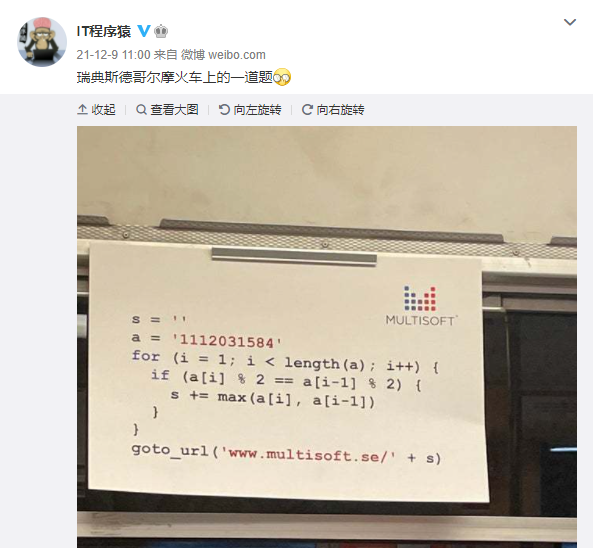
### 输出样例:

|  |
| --- |
| 1  1 1  1 2 1  1 3 3 1  1 4 6 4 1  1 5 10 10 5 1  1 6 15 20 15 6 1  1 7 21 35 35 21 7 1  1 8 28 56 70 56 28 8 1  1 9 36 84 126 126 84 36 9 1  1 10 45 120 210 252 210 120 45 10 1  1 11 55 165 330 462 462 330 165 55 11 1  1 12 66 220 495 792 924 792 495 220 66 12 1 |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **int main() {**  **int a[14][14] = {1}, n;**  **scanf("%d",&n);**  **for (int i = 0; i < n; i++) {**  **a[i][i] = 1;**  **for (int j = 0; j <=i; j++) printf("%-4d", a[i][j]);**  **for (int k = 1; k <=i; k++)a[i+1][k] = a[i][k] + a[i][k-1];**  **a[i+1][0] = 1;**  **printf("\n");**  **}**  **return 0;**  **}** |

**6．斯德哥尔摩火车上的题**



上图是新浪微博上的一则趣闻，是瑞典斯德哥尔摩火车上的一道题，看上去是段伪代码：

s = ''

a = '1112031584'

for (i = 1; i < length(a); i++) {

if (a[i] % 2 == a[i-1] % 2) {

s += max(a[i], a[i-1])

}

}

goto\_url('www.multisoft.se/' + s)

其中字符串的 + 操作是连接两个字符串的意思。所以这道题其实是让大家访问网站 www.multisoft.se/112358（**注意：比赛中千万不要访问这个网址！！！**）。

当然，能通过上述算法得到 112358 的原始字符串 a 是不唯一的。本题就请你判断，两个给定的原始字符串，能否通过上述算法得到相同的输出？

### 输入格式：

输入为两行仅由数字组成的非空字符串，长度均不超过 104，以回车结束。

### 输出格式：

对两个字符串分别采用上述斯德哥尔摩火车上的算法进行处理。如果两个结果是一样的，则在一行中输出那个结果；否则分别输出各自对应的处理结果，每个占一行。题目保证输出结果不为空。

### 输入样例 1：

|  |
| --- |
| 1112031584  011102315849 |

### 输出样例 1：

|  |
| --- |
| 112358 |

### 输入样例 2：

|  |
| --- |
| 111203158412334  12341112031584 |

### 输出样例 2：

|  |
| --- |
| 1123583  112358 |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **#include<string.h>**  **int main(){**  **char a[10001], b[10001],a1[10001]="",b1[10001]="";**  **int j = 0,i;**  **scanf("%s %s",a,b);**  **for( i = 1; i < strlen(a); i++){**  **if (a[i] % 2 == a[i-1] % 2)a1[j++] = a[i]>a[i-1]?a[i]:a[i-1];}**  **j = 0;**  **for( i = 1; i < strlen(b); i++){**  **if(b[i] % 2 == b[i - 1] % 2)b1[j++] = b[i]>b[i-1]?b[i]:b[i-1];}**  **if (strcmp(a1, b1) == 0)printf("%s\n", a1);**  **else printf("%s\n%s\n", a1, b1);**  **return 0;**  **}** |