

目录

样题-初级题: 从考试成绩中划出及格线.....	2
样题-中级题: 亮着电灯的盏数.....	2
样题-高级题: 地铁换乘.....	3
8.29 去掉最大值、最小值之后剩下的个数.....	3
8.29 从 5 个人中选取 2 个人作为礼仪.....	4
8.29 验证括号是否匹配.....	5
8.31 回文数.....	5
8.31 将第一行中含有第二行中“23”的数输出并排序.....	6
8.31 翻译电话号码.....	6
9.1.AM 将整数倒序输出, 剔除重复数据.....	7
9.1.AM 大数相减.....	8
9.1.AM 判断 if 语句括号是否合法.....	9
9.1.PM.....	10
9.1.PM.....	11
9.1.PM.....	11

样题-初级题：从考试成绩中划出及格线

10 个学生考完期末考试评卷完成后，A 老师需要划出及格线，要求如下：

- (1) 及格线是 10 的倍数；
- (2) 保证至少有 60% 的学生及格；
- (3) 如果所有的学生都高于 60 分，则及格线为 60 分

```
#include<stdio.h>
int main()
{
    int a[10],i,s[7]={0};
    for(i=0;i<10;i++)
    {
        scanf("%d",&a[i]);
        switch(a[i]/10){
            case(10):
            case(9):
            case(8):
            case(7):
            case(6):s[6]++;break;
            case(5):s[5]++;break;
            case(4):s[4]++;break;
            case(3):s[3]++;break;
            case(2):s[2]++;break;
            case(1):s[1]++;break;
            case(0):s[0]++;break;
        }
    }
    for(i=6;i>0;i--)
    {
        if(s[i]>=6) break;
        else s[i-1]+=s[i];
    }
    printf("%d",i*10);
    return 0;
}
```

样题-中级题：亮着电灯的盏数

一条长廊里依次装有 n ($1 \leq n \leq 65535$) 盏电灯，从头到尾编号 1、2、3、... $n-1$ 、 n 。每盏电灯由一个拉线开关控制。开始，电灯全部关着。

有 n 个学生从长廊穿过。第一个学生把号码凡是 1 的倍数的电灯的开关拉一下；接着第二个学生把号码凡是 2 的倍数的电灯的开关拉一下；接着第三个学生把号码凡是 3 的倍数的电灯的开关拉一下；如此继续下去，最后第 n 个学生把号码凡是 n 的倍数的电灯的开关拉一下。 n 个学生按此规定走完后，长廊里电灯有几盏亮着。

注：电灯数和学生数一致。

```
#include<stdio.h>
#define MAX 65535
int main()
{
    int n,i,j=1,count,m=0;
    scanf("%d",&n);
    int a[MAX]={0};
    while(j<=n)
    {
        count=0;
        for(i=1;i<=n;i++)
        {
            if(j%i==0) count++;
        }
        if(count%2!=0) m++;
        j++;
    }
    printf("%d",m);
    return 0;
}
```

样题-高级题：地铁换乘

已知 2 条地铁线路，其中 A 为环线，B 为东西向线路，线路都是双向的。经过的站点名分别如下，两条线交叉的换乘点用 T1、T2 表示。编写程序，任意输入两个站点名称，输出乘坐地铁最少需要经过的车站数量（含输入的起点和终点，换乘站点只计算一次）。

地 铁 线 A （ 环 线 ） 经 过 车 站：

A1 A2 A3 A4 A5 A6 A7 A8 A9 T1 A10 A11 A12 A13 T2 A14 A15 A16 A17 A18

地 铁 线 B （ 直 线 ） 经 过 车 站：

B1 B2 B3 B4 B5 T1 B6 B7 B8 B9 B10 T2 B11 B12 B13 B14 B15

8.29 去掉最大值、最小值之后剩下的个数

1、输入一串数，以','分隔，输出所有数中去掉最大值、最小值之后剩下的个数。（其中最大值与最小值可能有多个）

Smple input: 3,3,5,3,6,9,7,9 Sample outPut: 3

```
8_29_1.cpp
#include<stdio.h>
#define N 100
int main()
{
    int a[N],n=0,i,temp,min,max;
    char c='0';
    max=0;min=100;
    while(c!='\n')
    {
        scanf("%d",&temp);
        a[n]=temp;
        if(temp>max) max=temp;
        if(temp<min) min=temp;
        n++;
        scanf("%c",&c);
    }
    temp=0;
    for(i=0;i<n;i++)
    {
        if(a[i]<max && a[i]>min) temp++;
    }
    printf("%d",temp);
    return 0;
}
```

8.29 从 5 个人中选取 2 个人作为礼仪

2、要从 5 个人中选取 2 个人作为礼仪，其中每个人的身高范围为 160-190，要求 2 个人的身高差值最小（如果差值相同的话，选取其中最高的两人），以升序输出两个人的身高。

Smple input: 161 189 167 172 188 Sample outPut: 188 189

```
#include<stdio.h>
#include<math.h>
int main()
{
    int a[5],i,j,temp=30,n=-1,m=-1;
    for(i=0;i<5;i++)
    {
        scanf("%d",&a[i]);
        for(j=0;j<i;j++)
        {
            if(abs(a[i]-a[j])<temp)
            {
                temp=abs(a[i]-a[j]);
                if(a[i]>a[j]) {n=i;m=j;}
                else{n=j;m=i;}
            }
            else if(abs(a[i]-a[j])==temp && a[i]>a[m] && a[i]>a[n])
            {
                // a[i]>a[j]?n=i,m=j:n=j,m=i;
                if(a[i]>a[j]) {n=i;m=j;}
                else{n=j;m=i;}
            }
        }
    }
    printf("%d %d",a[m],a[n]);
    return 0;
}
```

8.29 验证括号是否匹配

3、输入一串字符串，其中有普通的字符与括号组成（包括‘（’、‘）’、‘[’、‘]’），要求验证括号是否匹配，如果匹配则输出 0、否则输出 1。

Smple input: dfa(sdf)df[dfds(df)] Smple outPut:0

```
#include<stdio.h>
int main()
{
    char a[100],c,i=0;
    int flag=0;
    scanf("%c",&c);
    while(c!='\n' && flag==0)
    {
        switch(c){
            case('('):
            case '[':a[i]=c;i++;break;
            case(')'):
                if(a[i-1]=='(') {a[i-1]='\0';i--;}
                else flag=1;
                break;
            case(']'):
                if(a[i-1]=='[') {a[i-1]='\0';i--;}
                else flag=1;
                break;
            // default:
            }
            scanf("%c",&c);
        }
        printf("%d",flag);
        return 0;
    }
```

8.31 回文数

1. 判断回文数，是返回 1

```
#include<stdio.h>
#define M 100
int main()
{
    int n,m=0,temp;
    scanf("%d",&n);
    temp=n;
    while(temp>0)
    {
        m*=10;
        m+=temp%10;
        temp=temp/10;
    }
    if(m==n) printf("1");
    else printf("0");
    return 0;
}
```

8.31 将第一行中含有第二行中“23”的数输出并排序

2. 输入一行数字: 123 423 5645 875 186523
在输入第二行: 23
将第一行中含有第二行中“23”的数输出并排序
结果即: 123 423 186523

```
#include<stdio.h>
#define M 100
int main()
{
    int a[M],temp,s,i,j;
    char c=' ';
    int sort[M],t=0;
    i=0;
    while(c!='\n')//存储数据
    {
        scanf("%d%c",&temp,&c);
        a[i]=temp;
        i++;
    }
    scanf("%d",&s);
    for(j=0;j<=i;j++)//查找关键数据
    {
        temp=a[j];
        while(temp>0)
        {
            if(temp%100==s)
            {
                // printf("%d ",a[j]);
                sort[t]=a[j];
                t++;
                break;
            }
            else temp=temp/10;
        }
    }
    for(i=0;i<t-1;i++)//排序
    {
        for(j=0;j<t-i-1;j++)
            if(sort[j]>sort[j+1])
            {
                temp=sort[j];
                sort[j]=sort[j+1];
                sort[j+1]=temp;
            }
    }
    for(i=0;i<t;i++)
        printf("%d ",sort[i]);
    return 0;
}
```

8.31 翻译电话号码

3. 将 电话号码 one two 。 。 。 nine zero

翻译成 1 2 ... 9 0

中间会有 double

例如输入: OneTwoThree

输出: 123

输入: OneTwoDoubleTwo

输出: 1222

输入: 1Two2 输出: ERROR

输入: DoubleDoubleTwo 输出: ERROR

第三题: 有空格, 非法字符, 两个 Double 相连, Double 位于最后一个单词 都错误

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[11][11]={"zero","one","two","three","four","five","six","seven","eight","nine","double"};
    char temp[11];char c=' ';
    int i,d=0,f;
    // while(temp!="\n")
    while(c!='\n')
    {
        scanf("%s%c",&temp,&c);
        f=0;
        for(i=0;i<11;i++)
        {
            if(!strcmp(temp,a[i]) && i<10) //错误: if(temp==a[i][0])
            {
                printf("%d ",i);
                f=1;
                if(d==1)
                {
                    printf("%d ",i);
                    d=0;
                }
            }
            else if(!strcmp(temp,a[i]) && i==10)
            {
                d=1;f=1;
            }
        }
        if(f==0) break;
    }
    if(d==1 || f==0) printf("ERROR\n");//最后一个人单词为double
    return 0;
}
```

9.1.AM 将整数倒序输出, 剔除重复数据

输入一个整数, 如 12336544, 或 1750, 然后从最后一位开始倒过来输出, 最后如果是 0, 则不输出, 输出的数字是不带重复数字的, 所以上面的输出是 456321 和 571。如果是负数, 比如输入-175, 输出-571。

```
#include<stdio.h>
int main()
{
    int Num,n[50],a[10]={0},temp,i=0,flag=0;
    char c;scanf("%c",&c);
    scanf("%d",&Num);

    while(Num>0)
    {
        temp=Num%10;
        if(a[temp]==0)
        {
            a[temp]=1;
            n[i]=temp;
            i++;
        }
        Num/=10;
    }
    if(c=='-') printf("%c",c);
    for(temp=0;temp<i;temp++)
    {
        if(n[temp]!=0 || flag!=0)
        {
            printf("%d",n[temp]);
            flag=1;
        }
    }
    if(c!='-') printf("%d",(int)c-48);
    return 0;
}
```

9.1.AM 大数相减

输入两行字符串正整数，第一行是被减数，第二行是减数，输出第一行减去第二行的结果。

备注：1、两个整数都是正整数，被减数大于减数

示例：

输入：10000000000000001

1

输出：10000000000000000

注意大数用 `char a[]` 存储，用 `%s` 接收，一位一位的运算。注意 `a[0]` 里的正负号。

```
#include<stdio.h>
#include<string.h>
int main()
{
    char s1[1010]={'\0'},s2[1010]={'\0'};
    scanf("%s %s",&s1,&s2);
    int len1=strlen(s1),len2=strlen(s2),i,j,m1,m2,f=0;
    int answer[1010],k=0;
    for(i=len1-1,j=len2-1;i>=0;i--,j--)
    {
        m1=(int)s1[i]-48;
        if(j>=0) m2=(int)s2[j]-48;
        else m2=0;
        if(f==1)
        {
            m1=m1-1;
            f=0;
        }
        if(m1>=m2)
        {
            answer[k++]=m1-m2;
            f=0;
        }
        else
        {
            answer[k++]=m1+10-m2;
            f=1;
        }
    }
    f=0; //此时f用来帮助判断结果数的最高位是否为0
    for(k=k-1;k>=0;k--)
    {
        if(f==0 && answer[k]==0) continue;
        else
        {
            f=1;
            printf("%d",answer[k]);
        }
    }
    return 0;
}
```

9.1.AM 判断 if 语句括号是否合法

编程的时候，if 条件里面的“(”、“)”括号经常出现不匹配的情况导致编译不过，请编写程序检测输入一行 if 语句中的圆括号是否匹配正确。同时输出语句中出现的左括号和右括号数量，如 if((a==1)&&(b==1)) 是正确的，而 if((a==1))&&(b==1))是错误的。注意 if 语句的最外面至少有一对括号。提示：用堆栈来做。

输入：if((a==1)&&(b==1))

输出：RIGTH 3 3

输入：if((a==1))&&(b==1))

输出：WRONG 3 4

```
#include<stdio.h>
#include<string.h>
int main()
{
    char s[800]={'\0'};
    scanf("%s",&s);
    int len=strlen(s),l=0,r=0;
    int a[50],k=0,flag=1;
    for(i=0;i<len;i++)
    {
        if(s[i]=='(')
        {
            l++;
            a[k]=1;
            k++;
        }
        else if(s[i]==')')
        {
            r++;
            if(a[k-1]==1 && k>0)
            {
                a[k-1]=0;
                k--;
            }
            else flag=0;
        }
        if((i==2 && s[i]!='(')||(i==len-1 && s[i]!=''))
            flag=0;
    }
    if(a[0]==0 && flag!=0) printf("RIGHT ");
    else printf("WRONG ");
    printf("%d %d",l,r);
    return 0;
}
```

9.1.PM 字符串 M 化成以 N 为单位的段

输入 m 个字符串 和一个整数 n, 把字符串 M 化成以 N 为单位的段, 不足的位数用 0 补齐。

如 n=8 m=9 ,

123456789 划分为: 12345678
90000000

123 化为: 12300000

```
#include<stdio.h>
#include<string.h>
int main()
{
    char c[200]={'\0'};
    scanf("%s",&c);
    int n,i,j,len=strlen(c);
    scanf("%d",&n);
    for(i=1;i<=len;i++)
    {
        j=i%n;
        printf("%c",c[i-1]);
        if(j==0) printf("\n");
    }
    for(i=j+1;i<=n;i++)
        printf("0");
    // printf("%s,%d",c,n);
    return 0;
}
```

9.1.PM 整数化为 2 进制数，32 位长度。然后逆序输出
整数化为 2 进制数，32 位长度。然后逆序输出

```
#include<stdio.h>
int main()
{
    int M,a[32]={0},i=0,j;
    scanf("%d",&M);
    while(M>0)
    {
        a[i]=M%2;
        i++;
        M/=2;
    }
    for(j=0;j<i;j++)
    {
        printf("%d",a[j]);
    }
    /*如果固定输出32位长度
    for(j=0;j<32;j++) printf("%d",a[j]);
    */
    return 0;
}
```

9.1.PM 背包

01 背包

给定一个数，比如 20

然后 再给定几个数字 1 3 5 7 8

1 3 5 7 8

0 0 1 1 1

因为 $5+7+8=20$

```
#include<stdio.h>
#include<string.h>
int a[50];
bool f[50]={0};
int find(int w,int s)
{
    if(w==0) return 1;//刚好，递归结束
    if(w<0 || w>0 && s==0) return 0;
    if(find(w-a[s-1],s-1)) {f[s-1]=1;return 1;}
    return find(w,s-1);//选择下一个重新递归
}
int main()
{
    int n,m,i;
    scanf("%d %d",&n,&m);
    for(i=0;i<m;i++) scanf("%d",&a[i]);
    if(find(n,m))
    {
        for(i=0;i<m;i++)
            printf("%d ",f[i]);
    }
    else
        printf("No\n");
    return 0;
}
```