第 3 章

程序控制结构习题解析

3.1 选择题解析

1. C/C++过程化程序设计的三种基本程序控制结构是（ ）。

- A. 顺序结构、选择结构、循环结构
- B. 输入、处理、输出
- C. for、while、do...while
- D. 复合语句、基本语句、空语句

解析:

C/C++过程化程序设计的三种基本程序控制结构是顺序结构、选择结构、循环结构，答案选 A。

选项 B 是简单算法的基本步骤，选项 C 是三种循环语句，选项 D 是三种语句形式。

2. 在 C/C++语言中，if 语句中的 else 子句总是与（ ）尚无 else 匹配的 if 相配对。

- A. 缩排位置相同
- B. 其之前最近
- C. 其之后最近
- D. 同一行上

解析:

在 C/C++语言中，if 语句中的 else 子句总是与其之前最近尚无 else 匹配的 if 相配对，答案选 B。

3. 以下代码段的输出结果是（ ）。

```
int i=1, j=2;
//C++风格代码
if (--i && --j) cout<<i<<" ";
else cout<<j<<" ";
if (++i || ++j) cout<<i<<endl;
else cout<<j<<endl;
//C 风格代码
if (--i && --j) printf("%d ",i);
else printf("%d ",j);
if (++i || ++j) printf("%d\n",i)
else printf("%d\n",j)
```

A. 1 1

B. 1 2

C. 2 3

D. 2 1

解析:

自增++和自减--运算符构成的表达式的求值规则如下。

(1) 根据++位置决定增1的时机, ++在前则先增1, 再取值; ++在后则先取值, 再增1。

(2) 根据--位置决定减1的时机, --在前则先减1, 再取值; --在后则先取值, 再减1。

逻辑运算符&和||的运算规则如下。

(1) $a \&\& b$: 若 a 、 b 同时为 true (或 1) 则 $a \&\& b$ 为 true (或 1), 否则为 false (或 0); 若 a 为 false (或 0), 则 b 不执行。

(2) $a \parallel b$: 若 a 、 b 同时为 false (或 0) 则 $a \parallel b$ 为 false (或 0), 否则为 true (或 1); 若 a 为 true (或 1), 则 b 不执行。

根据以上规则, 第一个 if 语句中, 因 $--i$ 的值为 0, 故 $\&\&$ 运算符之后的 $--j$ 不执行, if 后的条件不成立, 执行 else 子句后的语句, 输出 j 的值为 2; 第二个 if 语句中, 因第一个 if 语句使得 i 的值为 0, $++i$ 的值为 1, 故 $\parallel$ 运算符之后的 $++j$ 不执行, if 后的条件成立, 执行 if 子句后的语句, 输出 i 的值为 1; 答案选 D。

4. 执行以下代码段后, 变量 i 的值为 ()。

```
int i=1;
switch (i) {
    case 1: i+=10;
    case 2: i+=20;
    case 3: i++; break;
    default: i++; break;
}
```

A. 11

B. 31

C. 33

D. 32

解析:

i 的初值为 1, 与 “case 1:” 中的 1 相匹配, 故执行 $i+=10$ 使得 i 的值为 11, 但因没有 break 语句, 代码顺序执行下去, i 的值依次再增 20 和 1 而变为 32, 接着遇到 break, switch 语句执行结束, 最终 i 的值为 32, 答案选 D。

5. 下面有关 for 循环的正确描述是 ()。

A. for 循环只能用于循环次数确定的情况

B. for 循环先执行循环体语句, 后判断循环条件

C. 在 for 循环中, 不能用 break 语句跳出循环体

D. for 循环的循环体可以包含多条语句, 但多条语句必须构成复合语句

解析:

for 循环既可用于循环次数确定的情况, 也可用于循环次数不确定的情况, 选项 A 有误; for 循环先判断循环条件, 若条件成立则再执行循环体语句, 选项 B 有误; 在 for 循环中, 可用 break 语句跳出循环体, 选项 C 有误; 故答案选 D。

注意, 若循环体包含多条语句, 需用大括号将其括起来而构成复合语句。

6. 执行语句“for(s=0,k=0;s<=20||k<=10;k+=2) s+=k;”后, k 、 s 的值为 ()。

- A. 30、12 B. 12、30 C. 20、10 D. 10、20

解析:

循环在 s 不大于 20 或者 k 不大于 10 时执行, s 依次累加 0、2、4、6、8、10 而变为 30, k 的值最终变为 12, 从而使得循环条件不成立而结束循环, 故答案选 B。

7. k 、 s 的当前值为 5、0, 执行语句“while(--k) s+=k;”后, k 、 s 值分别为 ()。

- A. 15、0 B. 0、15 C. 10、0 D. 0、10

解析:

s 依次累加 4、3、2、1 而变为 10, 当 k 等于 1 并把其加到 s 中后, 下次判断循环条件时 $--k$ 的值为 0, 使得循环条件不成立而结束循环, 故答案选 D。

8. 关于 do...while 循环, 以下说法中正确的是 ()。

- A. 循环体语句只能有一条基本语句
B. 在while(循环条件)后面不能写分号
C. 当while后面循环条件的值为false (或0) 时结束循环
D. 根据情况可以省略while

解析:

do...while 循环的循环体语句可以有多条基本语句, 故选项 A 有误; do...while 循环的 while(循环条件)后面应加分号, 故选项 B 有误; do...while 循环的 do 和 while 是配套使用的, 都不能缺少, 故选项 D 有误; do...while 循环在循环条件成立时执行循环体, 在循环条件不成立时结束循环, 故答案选 C。

9. 以下不是无限循环的语句为 ()。

- A. `for(int y=10,x=1;x<=++y;x++);`
B. `for( ; ; );`
C. `while(1){x++;}`
D. `//C++风格代码`
`for(i=10;true;i--) sum+=i;`
`//C 风格代码`
`for(i=10;1;i--) sum+=i;`

解析:

选项 A 的循环体为空语句, 循环条件“ $x<=++y;$ ”在 y 等于 2147483647 时 $++y$ 等于 2147483648, 超出 int 范围从而变为 -2147483648, 此时 x 等于 2147483638, “2147483638 < -2147483648”这个循环条件不成立而结束循环, 故答案选 A。

for 循环的循环条件省略时, 相当于该条件为 true (或 1), 则其他选项的循环条件都是 true (或 1), 表示循环条件永真而使得相应的循环成为无限循环 (死循环)。

10. 下列代码段中循环体执行的次数为 ()。

```
int k=10;
while (k=1) k=k-1;
```

- A. 循环体语句一次都不执行 B. 循环体语句执行无数次
C. 循环体语句执行一次 D. 循环体语句执行9次

解析:

因“=”是赋值运算符,循环条件“k=1”使得 k 的值总是等于1而使循环条件永远成立,故出现死循环,答案选B。

3.2 编程题解析

1. 输入输出练习(1)—— n 个整数之和(T 组测试)

共有 T 组测试数据,每组测试求 n 个整数之和。

输入格式:

首先输入一个正整数 T ,表示测试数据的组数,然后是 T 组测试数据。每组测试先输入数据个数 n ,然后再输入 n 个整数,数据之间以一个空格间隔。

输出格式:

对于每组测试,在一行上输出 n 个整数之和。

输入样例:

```
2
4 1 2 3 4
5 1 8 3 4 5
```

输出样例:

```
10
21
```

解析:

使用 T 组测试的在线做题控制结构,对于每组测试,先输入数据个数 n ,再输入 n 个整数并逐个累加到累加单元 sum (初值为0)中,最后输出 sum 的值。

具体代码如下。

```
//C++风格代码
#include <iostream>
using namespace std;
int main() {
    int T;
    cin>>T;
    while(T--){ //控制T组测试
        int n, t, sum=0; //累加单元sum清0
```

```

        cin>>n;
        for(int i=0; i<n; i++) { //输入 n 个整数并累加到 sum 中
            cin>>t;
            sum+=t;
        }
        cout<<sum<<endl; //输出 sum 的值
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int T, i, n, t, sum;
    scanf("%d",&T);
    while(T--){ //控制 T 组测试
        sum=0; //累加单元 sum 清 0
        scanf("%d",&n);
        for(i=0; i<n; i++){ //输入 n 个整数并累加到 sum 中
            scanf("%d",&t);
            sum+=t;
        }
        printf("%d\n",sum); //输出 sum 的值
    }
    return 0;
}

```

2. 输入输出练习(2)—— n 个整数之和(处理到文件尾)

测试数据有多组, 处理到文件尾。每组测试求 n 个整数之和。

输入格式:

测试数据有多组, 处理到文件尾。每组测试先输入数据个数 n , 然后再输入 n 个整数, 数据之间以一个空格间隔。

输出格式:

对于每组测试, 在一行上输出 n 个整数之和。

输入样例:

```
5 1 8 3 4 5
```

输出样例:

```
21
```

解析:

使用控制到文件尾的在线做题控制结构, 对于每组测试, 先输入数据个数 n , 再输入 n 个整数并逐个累加到累加单元 `sum` (初值为 0) 中, 最后输出 `sum` 的值。

具体代码如下。

```
//C++风格代码
#include <iostream>
using namespace std;
int main() {
    int n;
    while(cin>>n) { //控制到文件尾, cin 在读不到数据时返回 NULL
        int t, sum=0; //累加单元 sum 清 0
        for(int i=0; i<n; i++){ //输入 n 个整数并累加到 sum 中
            cin>>t;
            sum+=t;
        }
        cout<<sum<<endl; //输出 sum 的值
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int i, n, t, sum;
    while(~scanf("%d",&n)) { //控制到文件尾, scanf 在读不到数据时返回-1, ~-1=0
        sum=0; //累加单元 sum 清 0
        for(i=0; i<n; i++){ //输入 n 个整数并累加到 sum 中
            scanf("%d",&t);
            sum+=t;
        }
        printf("%d\n",sum); //输出 sum 的值
    }
    return 0;
}
```

3. 输入输出练习(3)—— n 个整数之和(特值结束)

测试数据有多组, 每组测试求 n 个整数之和, 处理到输入的 n 是 0 为止。

输入格式:

测试数据有多组。每组测试先输入数据个数 n , 然后再输入 n 个整数, 数据之间以一个空格间隔, 当 n 为 0 时, 输入结束。

输出格式:

对于每组测试, 在一行上输出 n 个整数之和。

输入样例:

```
5 1 8 3 4 5
0
```

输出样例:

解析:

使用特值结束的在线做题控制结构,对于每组测试,先输入数据个数 n ,再输入 n 个数并逐个累加到累加单元 sum (初值为 0) 中,最后输出 sum 的值。当 n 为特值 0 时,用 `break` 语句跳出循环从而结束循环。

具体代码如下。

```
//C++风格代码
#include <iostream>
using namespace std;
int main() {
    while(true) { //永真循环
        int n, t, sum=0; //累加单元 sum 清 0
        cin>>n;
        if (!n) break; //在 n 等于 0 时结束循环, !n 表示 n==0
        for(int i=0; i<n; i++) { //输入 n 个整数并累加到 sum 中
            cin>>t;
            sum+=t;
        }
        cout<<sum<<endl; //输出 sum 的值
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    while(1) { //永真循环
        scanf("%d",&n);
        if (!n) break; //在 n 等于 0 时结束循环, !n 表示 n==0
        sum=0; //累加单元 sum 清 0
        for(i=0; i<n; i++) { //输入 n 个整数并累加到 sum 中
            scanf("%d",&t);
            sum+=t;
        }
        printf("%d\n",sum); //输出 sum 的值
    }
    return 0;
}
```

4. 输入输出练习 (4) —— n 个整数之和 (空行间隔)

求 n 个整数之和。 T 组测试,且要求每两组输出之间空一行。

输入格式:

首先输入一个正整数 T ,表示测试数据的组数,然后是 T 组测试数据。每组测试先输入数据个数 n ,然后再输入 n 个整数,数据之间以一个空格间隔。

输出格式:

对于每组测试, 在一行上输出 n 个整数之和, 每两组输出结果之间留一个空行。

输入样例:

```
2
4 1 2 3 4
5 1 8 3 4 5
```

输出样例:

```
10

21
```

解析:

使用 T 组测试的在线做题控制结构, 对于每组测试, 先输入数据个数 n , 再输入 n 个数并逐个累加到累加单元 sum (初值为 0) 中, 最后输出 sum 的值。对于“每两组输出结果之间留一个空行”, 这里采用“最后一组除外, 输出每组结果之后, 再输出一个空行”的策略。

具体代码如下。

```
//C++风格代码
#include <iostream>
using namespace std;
int main() {
    int T;
    cin>>T;
    while(T-->0) { //控制 T 组测试
        int n, t, sum=0; //累加单元 sum 清 0
        cin>>n;
        for(int i=0; i<n; i++) { //输入 n 个整数并累加到 sum 中
            cin>>t;
            sum+=t;
        }
        cout<<sum<<endl; //输出 sum 的值
        //若不是最后一组数据, 则在输出 sum 后再输出一个空行
        if (T) cout<<endl; //最后一组测试时 T 变为 0, T 表示 T!=0
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int T, i, n, t, sum;
    scanf("%d",&T); //控制 T 组测试
```

```

while(T--) {
    sum=0; //累加单元 sum 清 0
    scanf("%d",&n);
    for(i=0; i<n; i++) { //输入 n 个整数并累加到 sum 中
        scanf("%d",&t);
        sum+=t;
    }
    printf("%d\n",sum); //输出 sum 的值
    //若不是最后一组数据,则在输出 sum 后再输出一个空行
    if (T) printf("\n"); //最后一组测试时 T 变为 0, T 表示 T!=0
}
return 0;
}

```

5. 应缴电费

春节前后,电费大增。查询之后得知收费标准如下:

- 月用电量在 230 千瓦时及以下的部分按每千瓦时 0.4983 元收费;
- 月用电量在 231~420 千瓦时的部分按每千瓦时 0.5483 元收费;
- 月用电量在 421 千瓦时及以上的部分按每千瓦时 0.7983 元收费。

请根据月用电量(单位:千瓦时),按收费标准计算应缴的电费(单位:元)。

输入格式:

首先输入一个正整数 T ,表示测试数据的组数,然后是 T 组测试数据。对于每组测试,输入一个整数 n ($0 \leq n \leq 10000$),表示月用电量。

输出格式:

对于每组测试,输出一行,包含一个实数,表示应缴的电费。结果保留 2 位小数。

输入样例:

```

2
270
416

```

输出样例:

```

136.54
216.59

```

解析:

根据题面所说的收费标准,用 if 语句判断月用电量的范围并分段进行计算并输出。具体代码如下。

```

//C++风格代码
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    int T;

```

```

    cin>>T;
    while(T--){
        int a;
        cin>>a; //输入月用电量
        cout<<fixed<<setprecision(2); //为结果保留2位小数作准备
        //分段计算并输出
        if(a<=230) cout<<a*0.4983<<endl;
        else if(a<=420) cout<<230*0.4983+(a-230)*0.5483<<endl;
        else cout<<230*0.4983+(420-230)*0.5483+(a-420)*0.7983<<endl;
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int T,a;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&a); //输入月用电量
        //分段计算并输出
        if(a<=230) printf("%.2lf\n",a*0.4983);
        else if(a<=420) printf("%.2lf\n",230*0.4983+(a-230)*0.5483);
        else printf("%.2lf\n",230*0.4983+(420-230)*0.5483+(a-420)*0.7983);
    }
    return 0;
}

```

6. 小游戏

有一个小游戏,6个人上台计算手中扑克牌点数之和是否5的倍数,据说是小学生玩的。这里稍微修改一下玩法, n 个人上台,计算手中数字之和是否同时是3、5、7的倍数。

输入格式:

首先输入一个正整数 T ,表示测试数据的组数,然后是 T 组测试数据。每组测试先输入1个整数 n ($1 \leq n \leq 15$),再输入 n 个整数,数值都小于1000。

输出格式:

对于每组测试,若 n 个整数之和同时是3、5、7的倍数则输出YES,否则输出NO。

输入样例:

```

2
3 123 27 60
3 23 27 60

```

输出样例:

```

YES
NO

```

解析:

对输入的 n 个数累加求和到累加单元 sum (初值为 0) 中, 判断 sum 是否同时是 3、5、7 的倍数, 若是则输出 YES, 否则输出 NO。

具体代码如下。

```
//C++风格代码
#include<iostream>
using namespace std;
int main() {
    int T, sum, n, a;
    cin>>T;
    while(T--){
        sum=0; //累加单元清 0
        cin>>n;
        while(n--){ //输入 n 个数并累加到 sum 中
            cin>>a;
            sum+=a;
        }
        if(sum%5==0&&sum%7==0&&sum%3==0) //判断 sum 是否同时为 3、5、7 的倍数
            cout<<"YES\n";
        else
            cout<<"NO\n";
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int T, sum, n, a;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&n);
        sum=0; //累加单元清 0
        while(n--){ //输入 n 个数并累加到 sum 中
            scanf("%d",&a);
            sum+=a;
        }
        if(sum%5==0&&sum%7==0&&sum%3==0) //判断 sum 是否同时为 3、5、7 的倍数
            puts("YES");
        else
            puts("NO");
    }
    return 0;
}
```

因 3、5、7 中任意两个都是互质数，故同时是 3、5、7 的倍数可以表示为是 105 ($3 \times 5 \times 7$) 的倍数。

7. 购物

小明购物之后搞不清最贵的物品价格和所有物品的平均价格，请帮他编写一个程序实现。

输入格式:

测试数据有多组，处理到文件尾。每组测试先输入 1 个整数 n ($1 \leq n \leq 100$)，接下来的 n 行中每行输入 1 个英文字母表示的物品名及该物品的价格。测试数据保证最贵的物品只有 1 个。

输出格式:

对于每组测试，在一行上输出最贵的物品名和所有物品的平均价格，两者之间留一个空格，平均价格保留 1 位小数。

输入样例:

```
3
a 1.8
b 2.5
c 1.5
```

输出样例:

```
b 1.9
```

解析:

求最贵物品名可在每次输入物品名 c 和价格 t 时，判断 t 是否大于假设的最高价格 m (初值为 0)，若是则将 m 更新为 t 并保存 c 到结果物品名 r 中；求平均价格则可把每次输入的 t 累加到累加单元 s (初值为 0) 中，最终平均价格为 s/n 。

具体代码如下。

```
//C++风格代码
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    int n;
    while(cin>>n) {
        int i;
        double s=0, t, m=0; //价格总和 s 清 0, 假设的最高价格 m 置 0
        char c,r;
        for(i=1; i<=n; i++) {
            cin>>c>>t;
            s+=t; //将 t 累加到 s 中
            if(t>m) { //若 t 大于 m, 则将 m 更新为 t, 并保存 c 到 r 中
                m=t;
            }
        }
    }
}
```

```

        r=c;
    }
}
cout<<r<<" "<<fixed<<setprecision(1)<<s/n<<endl;
}
return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int i, n;
    double s, t, m;
    while(~scanf("%d",&n)) {
        s=m=0; //价格总和 s 清 0, 假设的最高价格 m 置 0
        char c,r;
        for(i=1; i<=n; i++) {
            scanf("%*c%c%lf",&c,&t); //注意需先将待输入字符之前的换行符过滤掉
            s+=t; //将 t 累加到 s 中
            if(t>m) { //若 t 大于 m, 则将 m 更新为 t, 并保存 c 到 r 中
                m=t;
                r=c;
            }
        }
        printf("%c %.1lf\n",r,s/n);
    }
    return 0;
}

```

上述 C 风格代码中, 注意需将输入物品名 c 之前确认输入 (物品数 n 及前一种物品名称和价格) 的回车键中的换行符吸收掉, 否则 c 将得到换行符而出错。

8. 等边三角形面积

数学基础对于程序设计能力而言很重要。请选择合适的方法计算等边三角形面积。

输入格式:

测试数据有多组, 处理到文件尾。每组测试输入 1 个实数表示等边三角形的边长。

输出格式:

对于每组测试, 在一行上输出等边三角形的面积, 结果保留 2 位小数。

输入样例:

```

1.0
2.0

```

输出样例:

```

0.43
1.73

```

解析:

设等边三角形的边长为 a , 则其面积 $s=a*a*\text{sqrt}(3)/4$, 其中 $\text{sqrt}(3)$ 表示 3 的平方根。系统函数 sqrt 的头文件为 C++ 形式的 cmath 或 C 语言形式的 math.h 。

具体代码如下。

```
//C++风格代码
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
int main() {
    double a;
    while(cin>>a) {
        cout<<setprecision(2)<<fixed<<sqrt(3.0)/4*a*a<<endl;
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
#include <math.h>
int main() {
    double a;
    while(~scanf("%lf",&a)) {
        printf("%.2lf\n",sqrt(3.0)/4*a*a);
    }
    return 0;
}
```

另外, 3 的平方根也可表达为 $3^{0.5}$, 故也可用 $\text{pow}(3, 0.5)$ 表示。等边三角形的面积还可利用其他方法求解, 请读者自行思考并编程实现。

9. 三七二十一

某天, 诺诺看到“三七二十一”(3721)数, 觉得很神奇, 这种数除以 3 余 2, 而除以 7 则余 1。例如 8 是一个 3721 数, 因为 8 除以 3 余 2, 8 除以 7 余 1。现在给出两个整数 a 、 b , 求区间 $[a, b]$ 中的所有 3721 数, 若区间内不存在 3721 数则输出 none。

输入格式:

首先输入一个正整数 T , 表示测试数据的组数, 然后是 T 组测试数据。每组测试输入两个整数 a, b ($1 \leq a < b < 2000$)。

输出格式:

对于每组测试, 在一行上输出区间 $[a, b]$ 中所有的 3721 数, 每两个数据之间留一个空格。如果给定区间不存 3721 数, 则输出 none。

输入样例:

```
2
1 7
1 100
```

输出样例:

```
none
8 29 50 71 92
```

解析:

可对区间 $[a, b]$ 中所有数逐个判断是否满足 3721 数的条件, 若满足则输出该数。为控制每两个 3721 数之间留一个空格, 可使用一个计数器变量 `cnt` (初值为 0), 若某个数是 3721 数, 则 `cnt` 增 1, 在输出该 3721 数前判断 `cnt` 是否大于 1, 若是则先输出一个空格 (采用“第一个数据除外, 输出每个数据之前, 先输出一个空格”的策略)。显然, 若区间内不存在 3721 数, 则 `cnt` 为 0。如此可在 `cnt` 等于 0 时输出 `none`。

具体代码如下。

```
//C++风格代码
#include <iostream>
using namespace std;
int main() {
    int T;
    cin>>T;
    while(T-->0) {
        int m, n, cnt=0; //计数器 cnt 清 0
        cin>>m>>n;
        for(int j=m; j<=n; j++) { //对区间内的所有数逐个判断
            if (j%3==2 && j%7==1){ //若 j 满足 3721 数的条件, 则 cnt 增 1 并输出 j
                cnt++; //计数器增 1
                if (cnt>1) cout<<" "; //若不是第一个数, 则先输出一个空格
                cout<<j;
            }
        }
        if (cnt==0) cout<<"none"; //若 cnt 等于 0, 则不存在 3721 数, 输出 none
        cout<<endl;
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
int main() {
    int T, j, m, n, cnt;
    scanf("%d", &T);
    while(T-->0) {
```

```

scanf("%d%d",&m,&n);
cnt=0; //计数器 cnt 清 0
for(j=m; j<=n; j++) { //对区间内的所有数逐个判断
    if (j%3==2 && j%7==1){ //若 j 满足 3721 数的条件, 则 cnt 增 1 并输出 j
        cnt++; //计数器增 1
        if (cnt>1) printf(" "); //若不是第一个数, 则先输出一个空格
        printf("%d",j);
    }
}
if (cnt==0) puts("none"); //若 cnt 等于 0, 则不存在 3721 数, 输出 none
else puts("");
}
return 0;
}

```

10. 胜者

Sg 和 Gs 进行乒乓球比赛, 进行若干局之后, 想确定最后谁是胜者(赢的局数多者胜)。

输入格式:

测试数据有多组, 处理到文件尾。每组测试先输入一个整数 n , 接下来的 n 行中每行输入两个整数 a 、 b ($0 \leq a, b \leq 20$), 表示 Sg 与 Gs 的比分是 a 比 b 。

输出格式:

对于每组测试数据, 若还不能确定胜负则输出 CONTINUE, 否则在一行上输出胜者 Sg 或 Gs。

输入样例:

```

3
3 11
13 11
11 9

```

输出样例:

```
Sg
```

解析:

使用两个计数器 cnt1 (初值为 0) 和 cnt2 (初值为 0), 分别统计 Sg 和 Gs 的胜局数, 再根据 cnt1 和 cnt2 的比较情况输出结果。

具体代码如下。

```

//C++风格代码
#include <iostream>
using namespace std;
int main() {
    int n;
    while(cin>>n) {
        int cnt1=0; //Sg 的胜局计数器清 0

```

```

        int cnt2=0;                //Gs 的胜局计数器清 0
        for(int i=0; i<n; i++) {
            int a,b;
            cin>>a>>b;
            if(a>b) cnt1++;        //Sg 胜 Gs, cnt1 增 1
            if(a<b) cnt2++;        //Gs 胜 Sg, cnt2 增 1
        }
        //根据 cnt1 和 cnt2 的比较情况输出结果
        if(cnt1==cnt2) cout<<"CONTINUE"<<endl;
        else if(cnt1>cnt2) cout<<"Sg"<<endl;
        else cout<<"Gs"<<endl;
    }
    return 0;
}

//C 风格代码
#include <stdio.h>
#include <math.h>
int main() {
    int a,b,i,n,cnt1,cnt2;
    while(~scanf("%d",&n)) {
        cnt1=cnt2=0;                //Sg 和 GS 的胜局计数器都清 0
        for(i=0; i<n; i++) {
            scanf("%d%d",&a,&b);
            if(a>b) cnt1++;        //Sg 胜 Gs, cnt1 增 1
            if(a<b) cnt2++;        //Gs 胜 Sg, cnt2 增 1
        }
        //根据 cnt1 和 cnt2 的比较情况输出结果
        if(cnt1==cnt2) puts("CONTINUE");
        else if(cnt1>cnt2) puts("Sg");
        else puts("Gs");
    }
    return 0;
}

```

11. 加密

信息安全很重要，特别是密码。给定一个 5 位的正整数 n 和一个长度为 5 的字母构成的字符串 s ，加密规则很简单，字符串 s 的每个字符变为它后面的第 k 个字符，其中 k 是 n 的每个数位上的数字。第一个字符对应 n 的万位上的数字，最后一个字符对应 n 的个位上的数字。简单起见， s 中的每个字符为 ABCDE 中的一个。

输入格式：

测试数据有多组，处理到文件尾。每组测试数据在一行上输入非负的整数 n 和字符串 s 。

输出格式：

对于每组测试数据，在一行上输出加密后的字符串。

输入样例:

12345 ABCDE

输出样例:

BDFHJ

解析:

因字符串相关知识在下一章介绍,本题暂不用字符串求解。这里对字符串 s 采用逐个字符输入的方式,每个字符加密后直接输出。为方便表达,先构造整数 n 的逆序数 m ,再用数位分离的方法($m\%10$ 取 m 的个位, $m/=10$ 去掉 m 的个位)依序取得原来的 n 的第1位至第5位加到字符串 s 的第1个至第5个字符上。

具体代码如下。

```
//C++风格代码
#include<iostream>
using namespace std;
int main() {
    int n;
    while(cin>>n) {
        int m=0;
        while(n>0) { //构造n的逆序数m
            m=m*10+n%10;
            n/=10;
        }
        char c;
        for(int i=0; i<=4; i++) { //逐个字符输入并加密输出
            cin>>c;
            c+=m%10;
            m/=10;
            cout<<c;
        }
        cout<<endl;
    }
    return 0;
}
```

```
//C 风格代码
#include <stdio.h>
int main() {
    int i,m,n;
    char c;
    while(~scanf("%d",&n)) {
        m=0;
        while(n>0) { //构造n的逆序数m
            m=m*10+n%10;
```

```

        n/=10;
    }
    getchar(); //注意此处要吸收确认 n 输入的换行符
    for(i=0; i<=4; i++) { //逐个字符输入并加密输出
        scanf("%c",&c);
        c+=m%10;
        m/=10;
        putchar(c);
    }
    putchar('\n');
}
return 0;
}

```

本题若用 C++ 的 `string` 变量或 C 语言的字符数组处理, 则代码可更简洁。读者可在学完 `string` 变量或字符数组的相关知识后换一种方法求解。

12. 比例

某班同学在操场上列队, 请计算男、女同学的比例。

输入格式:

测试数据有多组, 处理到文件尾。每组测试数据输入一个以“.”结束的字符串, 串中每个字符可能是 MmFf 中的一个, m 或 M 表示男生, f 或 F 表示女生。

输出格式:

对于每组测试数据, 在一行上输出男、女生的百分比, 结果四舍五入到 1 位小数。输出形式参照输出样例。

输入样例:

```
FFfm.
MfF.
```

输出样例:

```
25.0 75.0
33.3 66.7
```

解析:

对于字符串, 采用逐个字符输入的方式, 在遇到字符'!'时结束输入, 每输入一个非'!'的字符, 则总人数计数器 n (初值为 0) 增 1。对于输入的字符逐个判断是否为'f'或'F', 是则女生计数器变量 cnt (初值为 0) 增 1。最终女生人数为 cnt 、男生人数为 $n-cnt$ 。

具体代码如下。

```

//C++风格代码
#include<iostream>
#include<iomanip>
using namespace std;
int main() {

```

```

char c;
while(cin>>c) { //若无法读入第一个字符,则表示到文件尾
    int cnt=0,n=0; //女生计数器 cnt、总人数计数器 n 清 0
    while(true) {
        if (c=='.') break; //若输入 '.', 则一组数据输入结束,跳出循环
        if (c=='f' || c=='F') cnt++; //若是女生,则女生计数器 cnt 增 1
        n++; //总人数计数器 n 增 1
        cin>>c; //继续输入下一个字符
    }
    cout<<fixed<<setprecision(1)<<(n-cnt)*100.0/n<<" "<<cnt*100.0/n<<endl;
}

//C 风格代码
#include <stdio.h>
#include <string.h>
int main() {
    char c;
    int i,n,cnt;
    while(~scanf("%c",&c)) { //若无法读入第一个字符,则表示到文件尾
        cnt=0,n=0; //女生计数器 cnt、总人数计数器 n 清 0
        while(1) {
            if (c=='.') break; //若输入 '.', 则一组数据输入结束,跳出循环
            if (c=='f' || c=='F') cnt++; //若是女生,则女生计数器 cnt 增 1
            n++; //总人数计数器 n 增 1
            scanf("%c",&c); //继续输入下一个字符
        }
        getchar(); //吸收行末的换行符
        printf("%.11f %.11f\n", (n-cnt)*100.0/n, cnt*100.0/n);
    }
    return 0;
}

```

本题若用 C++ 的 `string` 变量或 C 语言的字符数组处理,则代码可更简洁。读者可在学完 `string` 变量或字符数组的相关知识后换一种方法求解。

13. 某校人数

某学校教职工人数不足 n 人,在操场排队,7 人一排余 5 人,5 人一排余 3 人,3 人一排余 2 人;请问该校人数有多少种可能?最多可能有多少人?

输入格式:

测试数据有多组,处理到文件尾。每组测试输入一个整数 n ($1 \leq n \leq 10000$)。

输出格式:

对于每组测试,输出一行,包含 2 个以一个空格间隔的整数,分别表示该校教职工人数有几种可能和最多可能的人数。