

**注意事項**

- ・必要な変数は適宜用意してください

**問1 最大と最小 / 難易度：低 / 目標タイム：15分**

**問題**

標準入力から数の個数  $n$  と  $n$  個の数を受け取り、その数の中の最大値と最小値を標準出力へ求めるプログラムを作成しましょう。

**入力される値**

```
n
a1
a2
a3
...
an
```

- ・数の個数  $n$  と、 $n$  個の数  $a_1, a_2, a_3, \dots, a_n$  が半角の整数で与えられます。

**入力制約**

以下の範囲で入力が行われます（範囲外の値に対するコーディングは不要）。

- ・数の個数  $0 \leq n \leq 100$
- ・各数  $0 \leq a_1, a_2, a_3, \dots, a_n \leq 100$

**出力する値**

```
最大値
最小値
```

- ・最大値、最小値には、半角の整数が表示されます。

**サンプルケース 1**

入力

```
5
23
6
9
59
34
```

出力

```
59
6
```

説明

入力された数を順に最大値・最小値と比較し、最大値・最小値を更新していくと、最大値が 59、最小値が 6 となります。

**素材（デフォルトのソース）**

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

public class Main {
    public static void main(String args[] ) throws Exception {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        System.out.println(s);
    }
}
```

## 問2 金種計算 / 難易度：低 / 目標タイム：15分

### 問題

標準入力から金額を受け取り、紙幣、硬貨の枚数を標準出力へ求めるプログラムを作成しましょう。

- ・用意されている金種は、10,000円、5,000円、1,000円、500円、100円、50円、10円、5円、1円とします。
- ・紙幣、硬貨の枚数は、大きい金額の金種から数えます（1,000円の場合、500円×2枚ではなく、1,000円×1枚）。

### 入力される値

price

- ・金額 price が半角の整数で与えられます。

### 入力制約

以下の範囲で入力が行われます（範囲外の値に対するコーディングは不要）。

- ・金額  $0 \leq \text{price} \leq 100,000$

### 出力する値

10000円の枚数  
5000円の枚数  
1000円の枚数  
500円の枚数  
100円の枚数  
50円の枚数  
10円の枚数  
5円の枚数  
1円の枚数

- ・各枚数には半角の整数が表示されます。

### サンプルケース 1

入力

57887

出力

5  
1  
2  
1  
3  
1  
3  
1  
2

説明

57,887 は  $10,000 \text{円} \times 5 + 5,000 \text{円} \times 1 + 1,000 \text{円} \times 2 + 500 \text{円} \times 1 + 100 \text{円} \times 3 + 50 \text{円} \times 1 + 10 \text{円} \times 3 + 5 \text{円} \times 1 + 1 \text{円} \times 2$  と分けることができます。

### 素材

//クラス名は必ず Main にする必要があります。

```
import java.util.*;
```

```
class Main {  
    public static void main(String[] args) throws Exception {  
  
        int price; //金額  
        int count; //枚数  
        int types[] = {10000, 5000, 1000, 500, 100, 50, 10, 5, 1 }; //金種  
  
        Scanner sc = new Scanner(System.in);  
  
        //以下にコードを追記してください。  
    }  
}
```

```
//ここまで  
sc.close();  
}  
}
```

### 問3 標準体重 / 難易度：中 / 目標タイム：20分

#### 問題

標準入力から身長、体重、標準体重の計算方法の数値を受け取った後、身長から標準体重を求め、標準体重の計算結果と、体重が標準体重を越えているかどうかの結果を標準出力へ求めるプログラムを作成しましょう。

標準体重の計算方法は以下の2通りです。いずれの場合も、標準体重は小数第二位以下を切り捨て、小数第一位までの小数としてください。

- ・計算方法1：  $(身長 - 100) \times 0.9$
- ・計算方法2：  $(身長 / 100) \times (身長 / 100) \times 22$

#### 入力される値

H  
W  
M

- ・身長 H、体重 W が半角の浮動小数点形式で与えられます。
- ・標準体重の計算方法 M が半角の整数で与えられ、M が 1 のとき計算方法 1、M が 2 のとき計算方法 2 で計算します。

#### 入力制約

以下の範囲で入力が行われます（範囲外の値に対するコーディングは不要）。

- ・身長 H  $100.0 \leq H \leq 300.0$
- ・体重 W  $0.0 \leq W \leq 1,000.0$
- ・標準体重の計算方法 M 1 または 2

#### 出力する値

①標準体重を超えている場合

●●  
over

②標準体重を超えていない場合

●●

- ・●●には半角の数値が入ります。数値は小数点第二位以下を切り捨て、小数第一位までの小数で表示してください。
- ・英数字は半角とし、大文字・小文字などは正確に指定してください。

#### サンプルケース 1

入力

165.5  
60.3  
1

出力

58.9  
over

説明

標準体重の計算方法 1  
 $(165.5 - 100) \times 0.9 = 65.5 \times 0.9 = 58.95$   
小数点第二位切り捨てで 58.9

#### サンプルケース 2

入力

155.5  
53.0  
2

出力

53.1

説明

標準体重の計算方法 2  
 $(155.5 \div 100) \times (155.5 \div 100) \times 22 = 1.555 \times 1.555 \times 22 = 53.19655$   
小数点第二位切り捨てで 53.1

## 素材

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

class Main {
    public static void main(String[] args) throws Exception {

        double height,weight; //身長、体重
        int howto; //計算方法

        Scanner sc = new Scanner(System.in);
        //以下にコードを追記してください。

        //ここまで

        WeightCheck wc = new WeightCheck(height,weight,howto);
        wc.calcStdWeight();
        wc.display();
        sc.close();
    }
}

//標準体重計算と標準出力のクラス
class WeightCheck {
    double height,weight,stdWeight; //身長、体重、標準体重
    int howto; //計算方法

    //入力値のセット (コンストラクタ)
    WeightCheck(double height, double weight, int howto) {
        //以下にコードを追記してください。

        //ここまで
    }

    //標準体重の計算
    void calcStdWeight() {
        //以下にコードを追記してください。

        //ここまで
    }

    //標準出力
    void display() {
        //以下にコードを追記してください。

        //ここまで
    }
}
```

#### 問 4 年齢の計算 / 難易度 : 中 / 目標タイム : 20 分

##### 問題

標準入力から入力された開始年月、終了年月を基に、経過年数を計算し、経過年数を標準出力へ求めるプログラムを作成しましょう。  
以下の誤った入力があった場合はエラーメッセージを出力し、プログラムを終了します。

- ・ 月が 1~12 以外の数で入力された場合
- ・ 開始年月が終了年月よりも後の年月で入力された場合

##### 入力される値

sYear  
sMonth  
eYear  
eMonth

- ・ 開始年と開始月を表す整数 sYear、sMonth、終了年と終了月を表す整数 eYear、eMonth が与えられます。

##### 入力制約

以下の範囲で入力が行われます。

範囲外の値に対するコーディングを行ってください。

- ・  $0 \leq sYear, eYear \leq 9,999$
- ・  $1 \leq sMonth, eMonth \leq 12$

##### 出力する値

①正しい入力の場合

●●年●●か月

- ・ ●●には半角の整数が入ります。

②誤った入力の場合

入力値不正

##### サンプルケース 1

入力

2000  
7  
2020  
4

出力

19 年 9 か月

説明

なし

##### サンプルケース 2

入力

-1  
7  
2020  
4

出力

入力値不正

説明

年の範囲外チェック

##### サンプルケース 3

入力

2000  
0  
2020  
4

出力

入力値不正

説明

月の範囲外チェック

## サンプルケース 4

入力

```
2030
7
2020
4
```

出力

```
入力値不正
```

説明

```
開始年月が終了年月よりも後のものが指定されているかチェック
```

## 素材 (デフォルトのソース)

//クラス名は必ず Main にする必要があります。

```
import java.util.*;

public class Main {
    public static void main(String args[] ) throws Exception {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        System.out.println(s);
    }
}
```

## 問 5 鉄道の料金 / 難易度 : 高 / 目標タイム : 25 分

### 問題

現行仕様のプログラムを、変更仕様に合わせて修正しましょう。

#### 現行仕様

ある鉄道会社では、標準入力から入力された駅間の距離を基に、運賃を標準出力へ出力するプログラムを使用しています。

距離と運賃の対応は以下の通りです。

駅間の距離 (km)	運賃 (円)	駅間の距離 (km)	運賃 (円)
0~3	140	21~25	400
4~6	180	26~30	480
7~10	190	31~35	570
11~15	230	36~40	650
16~20	320	41~50	740

#### 入力例

28

#### 出力例

480

#### 変更仕様

標準入力から乗車駅の駅番号と降車駅の駅番号を指定すると距離が計算され、運賃が計算されるように、プログラムの仕様変更を行いましょう。

隣り合う駅とその距離は以下の通りです。

隣り合う駅	距離 (km)	隣り合う駅	距離 (km)
A 駅~B 駅	1	F 駅~G 駅	2
B 駅~C 駅	8	G 駅~H 駅	3
C 駅~D 駅	3	H 駅~I 駅	9
D 駅~E 駅	4	I 駅~J 駅	4
E 駅~F 駅	5	J 駅~K 駅	3

- ・ A 駅から K 駅まで 1 本でつながっています。
- ・ A 駅から K 駅へ向かう方向だけでなく、K 駅から A 駅へ向かう方向でも正しく運賃が計算されるようにしてください。

#### 入力される値

Start  
End

- ・ 乗車駅の駅番号 Start と降車駅の駅番号 End が整数で与えられます。
- ・ 駅番号は A 駅から K 駅まで順に 0~10 が割り当てられています。

#### 入力制約

以下の範囲で入力が行われます (範囲外の値に対するコーディングは不要)。

- ・  $0 \leq \text{Start} \leq 10$
- ・  $0 \leq \text{End} \leq 10$
- ・ 乗車駅と降車駅は必ず異なる駅が指定されます。

#### 出力する値

●●

- ・ ●●には半角の整数が表示されます。

#### サンプルケース 1

入力

0  
1

出力

140



#### 説明

A 駅(0)から B 駅(1)の距離は 1km となり、0~3km の距離の料金が適用されます。

#### サンプルケース 2

##### 入力

4  
8

##### 出力

320

##### 説明

E 駅(4)から I 駅(8)の距離は 19km となり、16~20km の距離の料金が適用されます。

#### 素材

//クラス名は必ず Main にする必要があります。

```
import java.util.*;
```

```
class Main {
    //距離別料金
    static final int tariff[][] = { { 3, 140 }, { 6, 180 }, { 10, 190 }, { 15, 230 }, { 20, 320 },
                                     { 25, 400 }, { 30, 480 }, { 35, 570 }, { 40, 650 }, { 50, 740 }
    };

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int distance = sc.nextInt();

        System.out.print(calcCharge(distance));
    }

    //料金を計算
    static int calcCharge(int distance){
        int charge = 0;
        for ( int i = 0; i < tariff.length; i++ ){
            if(distance <= tariff[i][0] ){
                charge = tariff[i][1];
                break;
            }
        }
        return charge;
    }
}
```

## 問 6 文字のカウント / 難易度 : 高 / 目標タイム : 25 分

### 問題

標準入力から入力した文字列中に指定の文字が登場する回数をカウントするプログラムを作っていますが、以下のようなエラーが出て、正しく表示できません。  
このプログラムをデバッグし、正しく表示されるように修正しましょう。

### 入力例

```
AbAbcdecAbcDedE
```

### 出力例 (エラー)

```
入力された文字:null  
カウント結果  
e:0回 Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5  
    at CountCode.display(Main.java:39)  
    at Main.main(Main.java:11)
```

### 入力される値

```
String
```

- String は半角英字の文字列で構成されます。

### 入力制約

以下の範囲で入力が行われます (範囲外の値に対するコーディングは不要)。

- 入力する文字列の文字数は 1~100 文字までとします。
- 文字列中に含まれる文字は ABCDEabcde とします。

### 出力する値

```
入力された文字列:▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲  
カウント結果  
a:●回  
b:●回  
c:●回  
d:●回  
e:●回
```

- 大文字は小文字としてカウントします。
- ▲には英字、●には数値が入ります。
- 英字は辞書順で表示してください。
- 英数字、記号は半角を指定してください。

### サンプルケース 1

入力

```
AbAbcdecAbcDedE
```

出力

```
入力された文字:AbAbcdecAbcDedE  
カウント結果  
a:3回  
b:3回  
c:3回  
d:3回  
e:3回
```

説明

入力された文字が表示され、その下に各文字の出現回数が表示されます。大文字は小文字としてカウントされます。

### 素材

```
//クラス名は必ず Main にする必要があります。  
import java.util.*;
```

```

public class Main {
    public static void main(String args[]) throws Exception {

        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        CountCode cs = new CountCode();
        cs.display();
        sc.close();
    }
}

//文字をカウント、表示するクラス
class CountCode {
    char ch[] = { 'a', 'b', 'c', 'd', 'e' };
    int count[] = new int[ch.length + 1];
    String str;

    //文字列から各文字をカウント
    void count() {
        for (int i=0; i<str.length(); i++){
            for(int j=0; j<ch.length; j++){
                if(ch[j] == str.charAt(i)){
                    count[j]++;
                    break;
                }
            }
        }
    }

    //画面出力
    void display() {
        System.out.println("入力された文字:" + str);
        System.out.println("カウント結果");
        for (int i=count.length-1; i>0; i--) {
            System.out.print(ch[i] + ":" + count[i] + "回" );
        }
    }
}

```

**問1 最大と最小 / 難易度：低 / 目標タイム：15分**

正答例

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

class Main {
    public static void main(String[] args) throws Exception {

        int min = Integer.MAX_VALUE, max = 0, value, num;

        Scanner sc = new Scanner(System.in);
        num = sc.nextInt();

        for(int i=0; i<num; i++){
            value = sc.nextInt();
            if(min > value){
                min = value;
            }
            if(max < value){
                max = value;
            }
        }

        System.out.println(max + "\n" + min);
        sc.close();
    }
}
```

**問2 金種計算 / 難易度：低 / 目標タイム：15分**

正答例

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

class Main {
    public static void main(String[] args) throws Exception {

        int price; //金額
        int count; //枚数
        int types[] = {10000, 5000, 1000, 500, 100, 50, 10, 5, 1 }; //金種

        Scanner sc = new Scanner(System.in);

        //以下にコードを追記してください。
        price = sc.nextInt();

        for(int i=0; i<types.length;i++){
            count = price / types[i];
            price %= types[i];
            System.out.println(count);
        }
        //ここまで
        sc.close();
    }
}
```

### 問3 標準体重 / 難易度：中 / 目標タイム：20分

#### 正答例

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

class Main {
    public static void main(String[] args) throws Exception {

        double height,weight; //身長、体重
        int howto; //計算方法

        Scanner sc = new Scanner(System.in);
        //以下にコードを追記してください。
        height = sc.nextDouble();
        weight = sc.nextDouble();
        howto = sc.nextInt();
        //ここまで

        WeightCheck wc = new WeightCheck(height,weight,howto);
        wc.calcStdWeight();
        wc.display();
        sc.close();
    }
}

//標準体重計算と標準出力のクラス
class WeightCheck {
    double height,weight,stdWeight; //身長、体重、標準体重
    int howto; //計算方法

    //入力値のセット (コンストラクタ)
    WeightCheck(double height, double weight, int howto) {
        //以下にコードを追記してください。
        this.height = height;
        this.weight = weight;
        this.howto = howto;
        //ここまで
    }

    //標準体重の計算
    void calcStdWeight() {
        //以下にコードを追記してください。
        if (howto == 1) {
            stdWeight = Math.floor((height - 100) * 0.9 * 10) / 10;
        } else if (howto == 2) {
            stdWeight = Math.floor((height / 100) * (height / 100) * 22 * 10) / 10;
        }
        //ここまで
    }

    //標準出力
    void display() {
        //以下にコードを追記してください。
        System.out.println(stdWeight);

        if(weight > stdWeight){
            System.out.println("over");
        }
        //ここまで
    }
}
```

#### 問 4 年齢の計算 / 難易度 : 中 / 目標タイム : 20 分

##### 正答例

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

public class Main {
    public static void main(String args[]) throws Exception {

        int dYear,dMonth,err=0;

        Scanner sc = new Scanner(System.in);
        int sYear = sc.nextInt();
        int sMonth = sc.nextInt();
        int eYear = sc.nextInt();
        int eMonth = sc.nextInt();

        if(sYear < 0 || sYear > 9999 || eYear < 0 || eYear > 9999){
            err = 1;
        }
        if(sMonth < 1 || sMonth > 12 || eMonth < 1 || eMonth > 12){
            err = 1;
        }
        if((sYear * 100 + sMonth) > (eYear * 100 + eMonth)){
            err = 1;
        }

        if(err == 1){
            System.out.println("入力値不正");
        }else{
            dYear = eYear - sYear;
            if ( eMonth < sMonth ) {
                dYear--;
                dMonth = 12 + eMonth - sMonth;
            }else{
                dMonth = eMonth - sMonth;
            }
            System.out.println(dYear + "年" + dMonth + "か月");
        }
    }
}
```

#### 問 5 鉄道の料金 / 難易度 : 高 / 目標タイム : 25 分

##### 正答例

```
//クラス名は必ず Main にする必要があります。
import java.util.*;

class Main {
    //距離別料金
    static final int tariff[][] = { { 3, 140 }, { 6, 180 }, { 10, 190 }, { 15, 230 }, { 20, 320 },
                                     { 25, 400 }, { 30, 480 }, { 35, 570 }, { 40, 650 }, { 50, 740 }
    };
    static final int stationsDistance[] = { 1, 8, 3, 4, 5, 2, 3, 9, 4, 3 };

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int start = sc.nextInt();
        int end = sc.nextInt();

        System.out.print(calcCharge(calcDistance(start, end)));
    }
}
```

```

static int calcDistance(int start, int end){
    int station1,station2,distance=0;

    if(start < end){
        station1 = start;
        station2 = end;
    }else{
        station1 = end;
        station2 = start;
    }

    for (int i = station1; i < station2; i++){
        distance += stationsDistance[i];
    }
    return distance;
}

//料金を計算
static int calcCharge(int distance){
    int charge = 0;
    for ( int i = 0; i < tariff.length; i++ ){
        if(distance <= tariff[i][0] ){
            charge = tariff[i][1];
            break;
        }
    }
    return charge;
}
}

```

## 問 6 文字のカウント / 難易度：高 / 目標タイム：25 分

### 正答例

```

//クラス名は必ず Main にする必要があります。
import java.util.*;

public class Main {
    public static void main(String args[]) throws Exception {

        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        CountCode cs = new CountCode(str); //修正
        cs.count(); //追加
        cs.display();
        sc.close();
    }
}

//文字をカウント、表示するクラス
class CountCode {
    char ch[] = { 'a', 'b', 'c', 'd', 'e' };
    int count[] = new int[ch.length + 1];
    String str;

    //追加
    CountCode(String str){
        this.str = str;
    }
}
//ここまで

```

```
//文字列から各文字をカウント
void count() {
    String str_buf = str.toLowerCase(); //追加
    for (int i=0; i<str_buf.length(); i++){ //修正
        for(int j=0; j<ch.length; j++){
            if(ch[j] == str_buf.charAt(i)){ //修正
                count[j]++;
                break;
            }
        }
    }
}

//画面出力
void display() {
    System.out.println("入力された文字:" + str);
    System.out.println("カウント結果");
    for (int i=0; i<count.length-1; i++) { //修正
        System.out.println(ch[i] + ":" + count[i] + "回" ); //修正
    }
}
}
```