

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）

```
/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>

int main()
{
    char s[100];
    scanf("%s", s);
    printf("%s\n", s);
    return 0;
}
```

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

/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。*/

```
#include <stdio.h>
```

```
int main(){
    int price; //金額
    int count; //枚数
    int types[] = {10000, 5000, 1000, 500, 100, 50, 10, 5, 1 }; //金種
    int size = sizeof(types) / sizeof(int); //配列の要素数

    //以下にコードを追記してください。

    //ここまで

    return 0;
}
```

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

```
/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>

double calc_std_weight(double height, int howto);
void display(double weight, double std_weight);

int main(){
    double height, weight, std_weight; //身長、体重、標準体重
    int howto; //計算方法

    //以下にコードを追記してください。

    //ここまで

    std_weight = calc_std_weight(height, howto);
    display(weight, std_weight);

    return 0;
}

//標準体重の計算
double calc_std_weight(double height, int howto){
    //以下にコードを追記してください。

    //ここまで
}

//表示
void display(double weight, double std_weight) {
    //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。

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

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)

```
/* C や C++ などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に 0 を返却してください。 */
#include <stdio.h>

int main()
{
    char s[100];
    scanf("%s", s);
    printf("%s\n", s);
    return 0;
}
```

問 5 単語の並べ替え / 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

標準入力から複数の数値を受け取り、昇順に並べ替えた後、標準出力へ出力するプログラムを使用しています。

入力例

```
5
28
4
7
41
23
```

- ・ 1 行目に、入力する数字の個数を半角で指定します。
- ・ 2 行目以降に、各数字を半角で指定します。

出力例

```
4
7
23
28
41
```

- ・ 数値が昇順に表示されます。

変更仕様

数値の代わりに文字列を標準入力から複数受け取り、辞書順に並べ替えた後、標準出力へ出力するようにプログラムの仕様変更を行いましょう。

入力される値

```
n
W1
W2
W3
...
Wn
```

- ・ 単語の個数 n が整数で与えられます。
- ・ 単語 $W1 \sim Wn$ が英数字の文字列で与えられます。
- ・ 英字は小文字とします。

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）。

- ・ $1 \leq n \leq 50$
- ・ 単語 $W1, W2, \dots, Wn$ 、の文字数はそれぞれ 19 文字以内（終端文字を除く）

出力する値

```
1 個目の単語
2 個目の単語
...
n 個目の単語
```

- ・ 単語は辞書順に上から下に表示されます。

サンプルケース 1

入力

```
3
lemon
cherry
apple
```

出力

```
apple
cherry
lemon
```

説明

3つの単語を順に受け取り、辞書順に出力します。

素材

/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。*/

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void selection_sort(int numbers[], int num);
```

```
int main(){
```

```
    int i, num;
```

```
    int numbers[50]; //数値
```

```
    //入力
```

```
    scanf("%d",&num);
```

```
    for(i=0; i<num; i++){
```

```
        scanf("%d",&numbers[i]);
```

```
    }
```

```
    selection_sort(numbers, num);
```

```
    //出力
```

```
    for (i=0; i<num; i++){
```

```
        printf("%d\n",numbers[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

```
//ソート
```

```
void selection_sort(int numbers[], int num){
```

```
    int i, j, min;
```

```
    int tmp;
```

```
    for (i=0; i<num-1; i++) {
```

```
        min = i;
```

```
        for (j=i+1; j<num; j++) {
```

```
            //数値の比較
```

```
            if (numbers[min] > numbers[j])
```

```
                min = j;
```

```
        }
```

```
        //数値の入れ替え
```

```
        tmp = numbers[i];
```

```
        numbers[i] = numbers[min];
```

```
        numbers[min] = tmp;
```

```
    }
```

```
}
```


問 6 商品別売上集計 / 難易度：高 / 目標タイム：25 分

問題

A 店では、売上情報をもとに、商品ごとに売上集計を出すプログラムを作ることになりました。

標準入力から売上レコードの件数と、複数の売上レコードを受け取り、商品ごとに売上金額を算出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。

商品は以下の通りです。

商品番号	商品名	単価
1	pen	450
2	eraser	120
3	notebook	260

売上レコードの形式は以下の通りです。

商品番号	販売数
------	-----

入力例

```
5
3 30
1 45
2 10
3 10
2 15
```

- ・売上レコードの件数は 5 件で、商品番号 1 の売上レコードが 1 件、商品番号 2 の売上レコードが 2 件、商品番号 3 の売上レコードが 2 件となります。

出力例（エラー）

```
1 pen 13500
2 eraser 1200
3 notebook 6500
```

- ・商品に対応した正しい売上集計結果が出ていない。

入力される値

```
n
商品番号 販売数
商品番号 販売数
...
商品番号 販売数
```

- ・売上レコードの件数、商品番号、販売数は半角の整数で入力されます。
- ・1 商品に対する売上レコードは、1 レコードもない場合や複数レコードある場合があります。

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）。

- ・ $1 \leq n \leq 50$
- ・ $1 \leq \text{商品番号} \leq 3$
- ・ $1 \leq \text{販売数} \leq 100$

出力する値

```
商品番号 商品名 売上金額
商品番号 商品名 売上金額
商品番号 商品名 売上金額
```

- ・商品番号の昇順に表示します。
- ・売上レコードのない商品の情報は表示しないでください。
- ・商品番号、売上金額は半角の整数を指定します。

サンプルケース 1

入力

```
5
3 30
```

```
1 45
2 10
3 10
2 15
```

出力

```
1 pen 20250
2 eraser 3000
3 notebook 10400
```

説明

売上はそれぞれ以下の通りです。
商品番号 1 の pen は、 $30 \times 450 = 20250$
商品番号 2 の eraser は、 $25 \times 120 = 3000$
商品番号 3 の notebook は、 $40 \times 260 = 10400$

素材

```
/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>
```

```
//商品マスタ
```

```
struct _item {
    int item_no; //商品番号
    char name[10]; //商品名
    int price; //単価
};
```

```
//売上レコード
```

```
struct _record {
    int item_no; //商品番号
    int s_num; //売上個数
};
```

```
void sort_records(struct _record records[], int r_num);
```

```
void sum_records(struct _record records[], int r_num, struct _item items[], int i_num);
```

```
int main(){
```

```
    int i, r_num, i_num;
    struct _record records[100];
    struct _item items[] = {
        {1,"pen",450},
        {2,"eraser",120},
        {3,"notebook",260}
    };
```

```
//商品マスタの商品数
```

```
i_num = sizeof(items) / sizeof(struct _item);
```

```
//入力
```

```
scanf("%d",&r_num);
for(i=0; i<r_num; i++){
    scanf("%d %d",&records[i].item_no, &records[i].s_num);
}
```

```
sort_records(records, r_num);
```

```
sum_records(records, r_num, items, i_num);
```

```
return 0;
```

```
}
```

```
//データを商品番号の昇順にソート
```

```
void sort_records(struct _record records[], int r_num){
```

```

struct _record tmp;
int i, j;
for (i=0; i<r_num-1; i++) {
    for (j=r_num-1; j>i; j--) {
        if (records[j].item_no > records[j-1].item_no) {
            tmp = records[j];
            records[j] = records[j-1];
            records[j-1] = tmp;
        }
    }
}

//データを商品番号ごとにソート、集計、表示
void sum_records(struct _record records[], int r_num, struct _item items[], int i_num){
    int i, j;
    unsigned long sum[i_num];

    for(i=0;i<i_num;i++)
        sum[i] = 0;

    j=0;
    for (i = 0; i < r_num; i++) {
        if (items[j].item_no != items[i].item_no)
            j++;
        sum[j] += items[j].price * records[i].s_num;
    }

    for (i = 0; i < i_num; i++) {
        printf("%d %s %d¥n",items[i].item_no, items[i].name, sum[i]);
    }
}

```

問 1 最大と最小 / 難易度 : 低 / 目標タイム : 15 分

正答例

```

/* C や C++ などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に 0 を返却してください。 */
#include <stdio.h>
#include <limits.h>

int main(){
    int min = INT_MAX, max = 0, value, num, i;

    scanf("%d",&num);

    for(i=0; i<num; i++){
        scanf("%d",&value);
        if(min > value){
            min = value;
        }
        if(max < value){
            max = value;
        }
    }
    printf("%d\n%d", max, min);

    return 0;
}

```

問 2 金種計算 / 難易度 : 低 / 目標タイム : 15 分

正答例

```

/* C や C++ などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に 0 を返却してください。 */
#include <stdio.h>

int main(){
    int price; //金額
    int count; //枚数
    int types[] = {10000, 5000, 1000, 500, 100, 50, 10, 5, 1 }; //金種
    int size = sizeof(types) / sizeof(int); //配列の要素数

    //以下にコードを追記してください。
    int i;

    scanf("%d", &price);

    for(i=0; i<size; i++){
        count = price / types[i];
        price %= types[i];
        printf("%d\n", count);
    }
    //ここまで

    return 0;
}

```

問3 標準体重 / 難易度：中 / 目標タイム：20分

正答例

```
/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>

double calc_std_weight(double height, int howto);
void display(double weight, double std_weight);

int main(){
    double height, weight, std_weight; //身長、体重、標準体重
    int howto; //計算方法

    //以下にコードを追記してください。
    scanf("%lf", &height);
    scanf("%lf", &weight);
    scanf("%d", &howto);
    //ここまで

    std_weight = calc_std_weight(height, howto);
    display(weight, std_weight);

    return 0;
}

//標準体重の計算
double calc_std_weight(double height, int howto){
    //以下にコードを追記してください。
    if (howto == 1) {
        return (double) (int) ((height - 100) * 0.9 * 10) / 10;
    } else {
        return (double) (int) ((height / 100) * (height / 100) * 22 * 10) / 10;
    }
    //ここまで
}

//表示
void display(double weight, double std_weight) {
    //以下にコードを追記してください。
    printf("%.1lf¥n", std_weight);
    if(weight > std_weight){
        printf("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

```
/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>

int main(){
    int sYear, sMonth, eYear, eMonth, dYear, dMonth, err=0;

    scanf("%d", &sYear);
    scanf("%d", &sMonth);
    scanf("%d", &eYear);
    scanf("%d", &eMonth);

    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){
    printf("入力値不正");
}else{
    dYear = eYear - sYear;
    if (eMonth < sMonth) {
        dYear--;
        dMonth = 12 + eMonth - sMonth;
    }else{
        dMonth = eMonth - sMonth;
    }
    printf("%d年%dか月", dYear, dMonth);
}
return 0;
}

```

問5 単語の並べ替え / 難易度：高 / 目標タイム：25分

正答例

```

/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>
#include <string.h>

void selection_sort(char w[][20], int num);

int main(){
    int i, num;
    char words[50][20]; //文字列

    //入力
    scanf("%d",&num);
    for(i=0; i<num; i++){
        scanf("%19s",words[i]);
    }

    selection_sort(words, num);

    //出力
    for (i=0; i<num; i++){
        printf("%s\n", words[i]);
    }
    return 0;
}

//ソート
void selection_sort(char w[][20], int num){
    int i, j, min;
    char tmp[20];

    for (i=0; i<num-1; i++) {
        min = i;
        for (j=i+1; j<num; j++) {
            // 文字列の比較
            if (strcmp(w[min], w[j]) > 0)
                min = j;
        }
        // 交換
        strcpy(tmp, w[min]);
        strcpy(w[min], w[j]);
        strcpy(w[j], tmp);
    }
}

```

```

    }
    strcpy(tmp, w[i]);
    strcpy(w[i], w[min]);
    strcpy(w[min], tmp);
}
}

```

問6 商品別売上集計 / 難易度：高 / 目標タイム：25分

正答例

```

/* CやC++などシェルに実行結果コード返却を明示する言語を利用する場合 基本的に0を返却してください。 */
#include <stdio.h>

```

```

//商品マスタ

```

```

struct _item {
    int item_no; //商品番号
    char name[10]; //商品名
    int price; //単価
};

```

```

//売上レコード

```

```

struct _record {
    int item_no; //商品番号
    int s_num; //売上個数
};

```

```

void sort_records(struct _record records[], int r_num);

```

```

void sum_records(struct _record records[], int r_num, struct _item items[], int i_num);

```

```

int main(){

```

```

    int i, r_num, i_num;
    struct _record records[100];
    struct _item items[] = {
        {1,"pen",450},
        {2,"eraser",120},
        {3,"notebook",260}
    };

```

```

//商品マスタの商品数

```

```

i_num = sizeof(items) / sizeof(struct _item);

```

```

//入力

```

```

scanf("%d",&r_num);
for(i=0; i<r_num; i++){
    scanf("%d %d",&records[i].item_no, &records[i].s_num);
}

```

```

//データソート、集計、表示

```

```

sum_records(records, r_num, items, i_num);

```

```

return 0;

```

```

}

```

```

//データを商品番号の昇順にソート

```

```

void sort_records(struct _record records[], int r_num){
    struct _record tmp;
    int i, j;
    for (i=0; i<r_num-1; i++) {
        for (j=r_num-1; j>i; j--) {
            if (records[j].item_no < records[j-1].item_no) {
                tmp = records[j];
                records[j] = records[j-1];
            }
        }
    }
}

```

```

        records[j-1] = tmp;
    }
}
}

//データを商品番号ごとにソート、集計、表示
void sum_records(struct _record records[], int r_num, struct _item items[], int i_num){
    int i, j;
    unsigned long sum[i_num];

    for (i=0; i<i_num; i++)
        sum[i] = 0;

    sort_records(records, r_num);

    j=0;
    for (i = 0; i < r_num; i++) {
        if (items[j].item_no != records[i].item_no)
            j++;
        sum[j] += items[j].price * records[i].s_num;
    }

    for (i = 0; i < i_num; i++) {
        if(sum[i] != 0)
            printf("%d %s %d¥n",items[i].item_no, items[i].name, sum[i]);
    }
}

```