

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）

```
s = input()
print(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

types = [10000, 5000, 1000, 500, 100, 50, 10, 5, 1] #金種

#以下にコードを追記してください。

#ここまで

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

```
import math

#標準体重計算
def calcStdWeight(height,howto):
    #以下にコードを追記してください。

    #ここまで

#表示
def display(weight,stdWeight):
    #以下にコードを追記してください。

    #ここまで

#以下にコードを追記してください。

#ここまで

stdWeight = calcStdWeight(height,howto)
display(weight,stdWeight)
```

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)

```
s = input()
print(s)
```

問5 三角形の面積 / 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

標準入力から入力された三角形の3辺の長さを基に、三角形の面積を標準出力へ出力するプログラムを使用しています。

三角形の面積は、以下の公式を用いて算出しています。

3辺の長さが a 、 b 、 c である三角形の面積 S は

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{ただし、} s = \frac{a+b+c}{2}$$

入力例

3.0
3.0
4.2

出力例

4.5

- 面積 S は小数第二位以下を最近接偶数への丸め（銀行丸め）、小数第一位までの小数で表しています。

変更仕様

標準入力から三角形の3つの頂点の座標を指定すると、各辺の長さが計算され、三角形の面積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。

2つの頂点の座標 (a_1, b_1) 、 (a_2, b_2) 間の距離 D は、以下の公式を用いて算出します。

$$D = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2}$$

入力される値

x1 y1
x2 y2
x3 y3

- 3つの頂点の座標 (x_1, y_1) 、 (x_2, y_2) 、 (x_3, y_3) が入力されます。
- x_1 、 y_1 、 x_2 、 y_2 、 x_3 、 y_3 は半角の整数とします。
- ある頂点の座標が他の頂点の座標と等しくなることはありません。

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）。

- $-50 \leq x_1, x_2, x_3, y_1, y_2, y_3 \leq 50$

出力する値

S

- 面積 S は小数第二位以下を最近接偶数への丸め（銀行丸め）とし、小数第一位までの小数で表します。

サンプルケース 1

入力

1 2
1 5
4 2

出力

4.5

説明

三角形の3つの頂点 $(1,2)$ 、 $(1,5)$ 、 $(4,2)$ の辺の長さを基に、三角形の面積 4.5 を算出します。

素材

```
import math

#三角形の面積
def heron(a, b, c):
    s = (a + b + c) / 2
    return math.sqrt(s * (s - a) * (s - b) * (s - c))

#各辺の長さを入力
distances = [float(input()) for i in range(3)]

#三辺の長さを基に三角形の面積を算出
print(round(heron(*distances),1))
```


問6 データ整形と集計 / 難易度：高 / 目標タイム：25分

問題

標準入力からレコード件数とレコード件数分の名前・身長・体重のレコードデータを受け取り、平均身長と平均体重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。

平均身長・平均体重を求める際の注意点は以下の通りです。

- ・平均身長はセンチメートル単位、平均体重はキログラム単位で表示します。
- ・身長・体重の少なくともどちらかの値がないレコードは、集計対象から除外します。
- ・レコードデータの身長・体重は、数値の単位が異なる場合があります。
- ・身長の単位は、単位がない場合と cm の場合はセンチメートルとし、m の場合はメートルとします。
- ・体重の単位は、単位がない場合と kg の場合のいずれもキログラムとします。

入力例

```
4
sato,170.0,60kg
tanaka,,50.3
yoshida,168cm,
suzuki,1.65m,58.5
```

出力例（エラー）

```
nan nan
```

入力される値

```
n
N1,H1,W1
N2,H2,W2
...
Nn,Hn,Wn
```

- ・レコード件数 n は、半角の整数で与えられます。
- ・レコードは、各値が,で区切られます。
- ・レコードは、要素に値が存在しない場合があります。
- ・名前 $N1$ 、 $N2$ 、...、 Nn は、英文字列で与えられます。
- ・身長 $H1$ 、 $H2$ 、...、 Hn は、実数、または、実数の末尾に cm または m が付く文字列で与えられます。
- ・体重 $W1$ 、 $W2$ 、...、 Wn は、実数、または、実数の末尾に kg が付く文字列で与えられます。

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）。

- ・ $1 \leq n \leq 50$
- ・ $1 \leq N1$ 、 $N2$ 、...、 Nn の各文字数 ≤ 20
- ・ $0 \leq H1$ 、 $H2$ 、...、 Hn の各実数値 $\leq 1,000$ （データが存在する場合）
- ・ $0 \leq W1$ 、 $W2$ 、...、 Wn の各実数値 $\leq 1,000$ （データが存在する場合）

出力する値

```
身長の平均
体重の平均
```

- ・身長の平均は、単位をセンチメートルとして表します。
- ・体重の平均は、単位をキログラムとして表します。
- ・身長の平均、体重の平均はどちらも小数第二位以下を最近接偶数への丸め（銀行丸め）とし、小数第一位までの小数で表します。

サンプルケース 1

入力

```
4
sato,170.0,60.0
tanaka,,50.3kg
yoshida,168.0cm,
suzuki,1.65m,58.5
```

出力

```
167.5
```

59.2

説明

平均身長

$$(170.0 + 1.65 \times 100) \div 2 = 167.5$$

小数第二位以下を最近接偶数への丸めで 167.5

平均体重

$$(60.0 + 58.5) \div 2 = 59.25$$

小数第二位以下を最近接偶数への丸めで 59.2

素材

```
import pandas as pd

#数値かどうか
def is_num(s):
    try:
        int(s)
        return True
    except ValueError:
        return False

#身長の単位を揃え、数値へ変換
def height_num(height):
    if is_num(height):
        return float(height)
    elif height[:1] == "m":
        return float(height[:1])
    elif height[:2] == "cm":
        return float(height[:2])
    else:
        return None

#体重の単位を揃え、数値へ変換
def weight_num(weight):
    if is_num(weight):
        return float(weight)
    elif weight[:2] == "kg":
        return float(weight[:2])
    else:
        return None

#入力
n = int(input())
s = [input().split(',') for i in range(n)]
df = pd.DataFrame(s)

#列見出し
df.columns = ['name', 'height', 'weight']

#変換
df['height'] = df['height'].apply(height_num)
df['weight'] = df['weight'].apply(weight_num)

#出力
print(df.height.mean())
print(df.weight.mean())
```

問1 最大と最小 / 難易度：低 / 目標タイム：15分

正答例

```
min = 100
max = 0
num = int(input())

for i in range(num):
    value = int(input())
    if min > value:
        min = value
    if max < value:
        max = value

print(f"{max}\n{min}")
```

問2 金種計算 / 難易度：低 / 目標タイム：15分

正答例

```
types = [10000, 5000, 1000, 500, 100, 50, 10, 5, 1] #金種

#以下にコードを追記してください。

price = int(input())

for type in types:
    count = price // type
    price %= type
    print(count)

#ここまで
```

問3 標準体重 / 難易度：中 / 目標タイム：20分

正答例

```
import math

#標準体重計算
def calcStdWeight(height, howto):
    #以下にコードを追記してください。
    if howto == 1:
        return math.floor((height - 100) * 0.9 * 10) / 10
    else:
        return math.floor((height / 100) * (height / 100) * 22 * 10) / 10
    #ここまで

#表示
def display(weight, stdWeight):
    #以下にコードを追記してください。
    print(stdWeight)
    if weight > stdWeight :
        print("over")
    #ここまで

#以下にコードを追記してください。
height = float(input())
weight = float(input())
howto = int(input())
#ここまで
```

```
stdWeight = calcStdWeight(height, howto)
display(weight, stdWeight)
```

問 4 年齢の計算 / 難易度 : 中 / 目標タイム : 20 分

正答例

```
err = 0

s_year = int(input())
s_month = int(input())
e_year = int(input())
e_month = int(input())

if not(1 <= s_year <= 9999 and 1 <= e_year <= 9999) :
    err = 1
if not(1 <= s_month <= 12 and 1 <= e_month <= 12) :
    err = 1
if (s_year * 100 + s_month) > (e_year * 100 + e_month) :
    err = 1

if err == 1 :
    print("入力値不正")
    exit()

dYear = e_year - s_year
if e_month < s_month :
    dYear -= 1
    dMonth = 12 + e_month - s_month
else:
    dMonth = e_month - s_month

print(f"{dYear}年{dMonth}か月")
```

問 5 三角形の面積 / 難易度 : 高 / 目標タイム : 25 分

正答例

```
import math

#三角形の面積
def heron(a, b, c):
    s = (a + b + c) / 2
    return math.sqrt(s * (s - a) * (s - b) * (s - c))

#2点間の距離
def calc_distance(vartex1, vartex2):
    dx = vartex2[0] - vartex1[0]
    dy = vartex2[1] - vartex1[1]
    return math.sqrt(dx**2 + dy**2)

#頂点の座標入力
vertexes = [list(map(int,input().split())) for i in range(3)]

#各辺の長さを算出
distances = [calc_distance(vertexes[i], vertexes[(i+1)%3]) for i in range(3)]

#三辺の長さを基に三角形の面積を算出
print(round(heron(*distances), 1))
```

問6 データ整形と集計 / 難易度：高 / 目標タイム：25分

正答例

```
import pandas as pd

#数値かどうか
def is_num(s):
    try:
        float(s)
        return True
    except ValueError:
        return False

#身長単位を揃え、数値へ変換
def height_num(height):
    if is_num(height):
        return float(height)
    elif height[-2:] == "cm":
        return float(height[:-2])
    elif height[-1:] == "m":
        return float(height[:-1]) * 100
    else:
        return None

#体重単位を揃え、数値へ変換
def weight_num(weight):
    if is_num(weight):
        return float(weight)
    elif weight[-2:] == "kg":
        return float(weight[:-2])
    else:
        return None

#入力
n = int(input())
s = [input().split(',') for i in range(n)]
df = pd.DataFrame(s)

#列見出し
df.columns = ['name', 'height', 'weight']

#変換
df['height'] = df['height'].apply(height_num)
df['weight'] = df['weight'].apply(weight_num)

df.dropna(inplace=True)

#出力
print(df.height.mean().round(1))
print(df.weight.mean().round(1))
```