
立志舎

「SUKI-1 GRAND PRIX」

プログラミングバトル

TechFUL例題・解答解説

【問題1】アメの分配

問題文

A 個のアメが入った袋を B 個買いました。
全てのアメを C 人で等しく分けることができますでしょうか？
できる場合は **Yes** を、できない場合は **No** を出力してください。

入力される値

A B C

- 1行目に、正の整数 A, B, C が空白区切りで与えられます。

期待される出力値

-
- 全てのアメを C 人で等しく分けることができる場合は **Yes** を、できない場合は **No** を1行に出力してください。

制約

- 入力はすべて整数
- $1 \leq A, B, C \leq 100$

サンプルケース1

入力値

44 1 2

期待される出力値

Yes

サンプルケース2

入力値

44 1 3

期待される出力値

No

サンプルケース3

入力値

44 3 3

期待される出力値

Yes

解説

A×の合計数(candy_total)は次のように計算できます。

$\text{candy_total} = A * B$

「C人で等しく分けることができるか？」の部分は、プログラム上では「Cで割った余りが0か？」と言い換えて実装することができます。

つまり、candy_totalをCで割った余りが0であるなら **Yes** を、0でなければ **No** を出力するコードを実装すれば、この問題に合格することができます。

python

```
Python
a, b, c = map(int, input().split())

candy_total = a * b

if candy_total % c == 0:
    print("Yes")
else:
    print("No")
```

C言語

```
C/C++
#include <stdio.h>

int main() {
    int a, b, c;
    int candy_total;

    scanf("%d %d %d", &a, &b, &c);

    candy_total = a * b;

    if (candy_total % c == 0) {
        printf("Yes\n");
    } else {
        printf("No\n");
    }

    return 0;
}
```

Java

```
Java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int a = scanner.nextInt();
        int b = scanner.nextInt();
        int c = scanner.nextInt();

        int candyTotal = a * b;

        if (candyTotal % c == 0) {
            System.out.println("Yes");
        } else {
            System.out.println("No");
        }

        scanner.close();
    }
}
```

【問題2】テストの点差

問題文

N 人の学生がテストを受けました。 i 番目の学生の点数は A_i 点でした ($1 \leq i \leq N$)。最低点と最高点の差は何点か求めてください。

入力される値

N

$A_1 A_2 \dots A_N$

- 1 行目に、学生数 N が与えられます。
- 2 行目に、 i 番目の学生の点数 A_i が空白区切りで与えられます ($1 \leq i \leq N$)。

期待される出力値

- 最低点と最高点の差を 1 行に出力してください。

制約

- 入力はすべて整数
- $2 \leq N \leq 100$
- $0 \leq A_i \leq 1000$ ($1 \leq i \leq N$)

サンプルケース1

入力値

2

33 44

期待される出力値

11

サンプルケース2

入力値

2

44 44

期待される出力値

0

サンプルケース3

入力値

3
50 60 40

期待される出力値

20

解説

※問題文では配列は A_1 から始まっていますが、プログラム中の配列の添字は 0 から始まるため、解説では A_0 から始めています。

配列Aの最大値と最小値を求め、その差を出力すればよいです。

最大値は次のように求められます。

1. 最大値を管理する変数(a_max)を用意する。
2. a_maxに最初 A_0 を代入する。
3. a_maxより A_1 の方が大きければ、a_maxに A_1 を代入する。
4. $A_2, A_3, \dots, A_{(N-1)}$ に対しても同じように処理をすると、最大値が求まる。

同様の手順で最小値も求め、最大値 - 最小値を出力すれば、問題に合格することができます。

python

```
Python
n = int(input())
a = list(map(int, input().split()))

a_max = a_min = a[0]
for i in range(1, n):
    if a_max < a[i]:
        a_max = a[i]
    if a_min > a[i]:
        a_min = a[i]

print(a_max - a_min)
```

C言語

```
C/C++
#include <stdio.h>

int main() {
    int n;
```

```

int a[100];

scanf("%d", &n);

for (int i = 0; i < n; i++) {
    scanf("%d", &a[i]);
}

int a_max = a[0], a_min = a[0];
for (int i = 1; i < n; i++) {
    if (a_max < a[i]) {
        a_max = a[i];
    }
    if (a_min > a[i]) {
        a_min = a[i];
    }
}

printf("%d\n", a_max - a_min);
return 0;
}

```

Java

```

Java

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n = scanner.nextInt();
        int[] a = new int[100];
        for (int i = 0; i < n; i++) {
            a[i] = scanner.nextInt();
        }

        int a_max = a[0], a_min = a[0];
        for (int i = 1; i < n; i++) {
            if (a_max < a[i]) {
                a_max = a[i];
            }
            if (a_min > a[i]) {
                a_min = a[i];
            }
        }

        System.out.println(a_max - a_min);
    }
}

```

【問題3】お宝発掘

問題文

$N \times N$ のマスがあり、あるマス 1 箇所にお宝が埋まっています。
上から i 行目、左から j 列目に位置するマスを (i, j) で表します。
各マスには半角英大文字が 1 つ書かれています。
そのマスがある行を左から順に並べた文字列と、そのマスがある列を上から順に並べた文字列
が同じ文字列になることがお宝の目印です。
お宝が上から何行目、左から何列目に埋まっているかを求めてください。

入力される値

N
 $C_{\{1, 1\}} C_{\{1, 2\}} \dots C_{\{1, N\}}$
 $C_{\{2, 1\}} C_{\{2, 2\}} \dots C_{\{2, N\}}$
...
...
 $C_{\{N, 1\}} C_{\{N, 2\}} \dots C_{\{N, N\}}$

- 1 行目に、正の整数 N が与えられます。
- $1 + i$ 行目に、マス (i, j) に書かれている半角英大文字 $C_{i,j}$ が空白区切りで与えられます ($1 \leq i, j \leq N$)。

期待される出力値

- お宝が上から何行目、左から何列目に埋まっているかを求めてください。お宝が埋まっているマスを (a, b) とするとき、 a, b をこの順に空白区切りで 1 行に出力してください。

制約

- N は整数で、 $1 \leq N \leq 10$
- $C_{i,j}$ は半角英大文字 ($1 \leq i, j \leq N$)
- お宝はちょうど 1 箇所のマスだけに埋まっていることが保証されます

サンプルケース1

入力値

3
X A Y
A B C
Z C X

期待される出力値

2 2

サンプルケース2

入力値

4
A B A B
A A B B
A B A B
A B A A

期待される出力値

4 3

解説

※問題文では配列の添字は1から始まっていますが、プログラム中の配列の添字は0から始まるため、解説では0から始めています。

全ての (i, j) の組に関して、その組がお宝の目印の条件を満たしているかを判定します。

お宝の目印の条件は「そのマスがある行を左から順に並べた文字列と、そのマスがある列を上から順に並べた文字列が同じ文字列になること」と問題文に書かれています。

今回の条件をプログラムで判定するには、ループ変数 k を用意し、 k を $0, 1, 2, \dots (N - 1)$ と増やしながら、全ての k について、マス (i, k) とマス (k, i) が同じ文字であるかを確認します。

全ての k について上記の条件が成り立つ場合(同じ文字である場合)に、 $i + 1, j + 1$ を出力することで、この問題に合格することができます。

python

```
Python
n = int(input())
c = [input().split() for i in range(n)]

for i in range(n):
    for j in range(n):
        flag = True
        for k in range(n):
            if c[i][k] != c[k][j]:
                flag = False
                break
```

```
if flag:
    print(i+1, j+1)
```

C言語

```
C/C++
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    char c[10][10];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf(" %c", &c[i][j]);
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            int flag = 1;
            for (int k = 0; k < n; k++) {
                if (c[i][k] != c[k][j]) {
                    flag = 0;
                    break;
                }
            }
            if (flag) {
                printf("%d %d\n", i + 1, j + 1);
            }
        }
    }

    return 0;
}
```

Java

```
Java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();

        char[][] c = new char[10][10];

        for (int i = 0; i < n; i++) {
```

```

        for(int j = 0; j < n; j++){
            c[i][j] = scanner.next().charAt(0);
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            boolean flag = true;
            for (int k = 0; k < n; k++) {
                if (c[i][k] != c[k][j]) {
                    flag = false;
                    break;
                }
            }
            if (flag) {
                System.out.println((i + 1) + " " + (j + 1));
            }
        }
    }

    scanner.close();
}
}

```