
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】文字列の長さ

問題文

文字列 s が与えられます。文字列 s の長さを出力してください。

入力される値

S

- 1行目に、文字列 s が与えられます。

期待される出力値

- 文字列 s の長さを出力してください。

制約

- 文字列 s の長さは 1 以上 100 以下

サンプルケース1

入力値

Hello

期待される出力値

5

サンプルケース2

入力値

banana

期待される出力値

6

解説

与えられた文字列の長さを出力すれば、正解できます。

Pythonならlen()、C言語ならstrlen()、Javaならlength()を用いることで、文字列の長さを取得することができます。

Python

```
Python
S = input()
print(len(S))
```

C言語

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

int main()
{
    char S[101];
    scanf("%s", S);
    printf("%d\n", strlen(S));
    return 0;
}
```

Java

```
Java
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.length());
    }
}
```

【問題2】偶数？奇数？

問題文

正の整数 N が与えられます。 N が偶数なら `Even`、奇数なら `Odd` と出力してください。

入力される値

N

- 1行目に整数 N が与えられます。

期待される出力値

- N が偶数なら `Even`、奇数なら `Odd` と出力してください。

制約

- 整数 N は 1 以上 100 以下

サンプルケース1

入力値

2

期待される出力値

Even

サンプルケース2

入力値

5

期待される出力値

Odd

解説

与えられた整数を2で割った時の余りが0なら偶数、そうでないなら奇数と判定すればよいです。余りは「%」を使うと求めることができます。

(例: x を y で割った余りの場合、 $x \% y$ と書く)

Python

```
Python
N = int(input())

if N % 2 == 0:
    print("Even")
else:
    print("Odd")
```

C言語

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

int main()
{
    int N;
    scanf("%d", &N);
    if(N % 2 == 0) {
        printf("Even\n");
    } else {
        printf("Odd\n");
    }
    return 0;
}
```

Java

```
Java
import java.util.*;

public class Main {
    public static void main(String args[] ) throws Exception {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        if(N % 2 == 0) {
            System.out.println("Even");
        }
    }
}
```

```
    } else {  
        System.out.println("Odd");  
    }  
}  
}
```

【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 (i, j は 1 以上 N 以下)。

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

-
- 整数 N は 1 以上 10 以下
 - $C_{i,j}$ は半角英大文字 (i, j は 1 以上 N 以下)
 - お宝は 1 箇所のマスのみ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

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 $1, 2, 3, \dots, N$ と増やしなから、全ての k について、 $c_{ik} = c_{kj}$ が成り立つかどうかを確認します。

全ての k について成り立つ場合に、 i, j を出力することで、この問題に正解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 sc = new Scanner(System.in);
        int n = sc.nextInt();

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

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

```

        c[i][j] = sc.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));
        }
    }
}

sc.close();
}
}

```