

1. Морской бой [5]

Вы участвуете в разработке подсистемы проверки поля для игры "Морской бой". Вам требуется написать проверку корректности количества кораблей на поле, учитывая их размеры. Напомним, что на поле должны быть:

- 1 четырёхпалубный корабль
- 2 трёхпалубных корабля
- 3 двухпалубных корабля
- 4 однопалубных корабля

Вам заданы 10 целых чисел от 1 до 4. Вам необходимо проверить, что количество кораблей каждого типа соответствует правилам игры.

Для каждого набора входных данных в отдельной строке выведите:

YES, если заданные размеры кораблей на поле соответствуют правилам игры; NO в противном случае.

Входные данные:

```
2 1 3 1 2 3 1 1 4 2
1 1 1 2 2 2 3 3 3 4
1 1 1 1 2 2 2 3 3 4
4 3 3 2 2 2 1 1 1 1
4 4 4 4 4 4 4 4 4 4
```

Выходные данные:

```
YES
NO
YES
YES
NO
```

```
import sys

def check_ints(arr):
    board = {}
    for item in arr:
        if 1 <= item <= 4:
            board[item] = board.get(item, 0) + 1
        else:
            return False
    return (
        board.get(1, 0) == 4
        and board.get(2, 0) == 3
        and board.get(3, 0) == 2
        and board.get(4, 0) == 1
    )

def check_board(board):
    print("YES" if check_ints(board) else "NO")

if __name__ == "__main__":
    for line in sys.stdin:
        line = line.strip()
        if not line:
            break
        check_board(list(map(int, line.split())))
```

2. Проверка даты [10]

Задана дата в формате "день месяц год" в виде трёх целых чисел. Гарантируется, что:

день — это целое число от 1 до 31;
месяц — это целое число от 1 до 12;
год — это целое число от 1950 до 2300.

Проверьте, что заданные три числа соответствуют корректной дате.

Напоминаем, что в соответствии с современным календарём год считается високосным, если для этого года верно хотя бы одно из утверждений: делится на 4, но при этом не делится на 100; делится на 400. Например, годы 2012 и 2000 являются високосными, но годы 1999, 2022 и 2100 - нет.

Для каждого набора входных данных выведите YES, если соответствующая дата является корректной (т.е. существует такая дата в современном календаре). Выведите NO в противном случае.

Входные данные:

```
10 9 2022
21 9 2022
29 2 2022
31 2 2022
29 2 2000
29 2 2100
31 11 1999
31 12 1999
29 2 2024
29 2 2023
```

Выходные данные:

```
YES
YES
NO
NO
YES
NO
NO
YES
YES
NO
```

```
import sys
from datetime import datetime

def is_valid_date(day, month, year):
    try:
        datetime(year, month, day)
        return True
    except ValueError:
        return False

def main():
    for line in sys.stdin:
        line = line.strip()
        if not line:
            break
        date = list(map(int, line.split()))
        if is_valid_date(*date):
            print("YES")
        else:
            print("NO")

if __name__ == "__main__":
    main()
```

3. Автомобильные номера [15]

Наши номера состоят из цифр и прописных букв латинского алфавита. Они бывают двух видов:

буква–цифра–цифра–буква–буква (примеры корректных номеров первого вида: R48FA, O00OO, A99OK);
буква–цифра–буква–буква (примеры корректных номеров второго вида: T7RR, A9PQ, O0OO).

Таким образом, каждый автомобильный номер является строкой либо первого, либо второго вида.

Вам задана строка из цифр и прописных букв латинского алфавита. Можно ли разделить её пробелами на последовательность корректных автомобильных номеров? Иными словами, проверьте, что заданная строка может быть образована как последовательность корректных автомобильных номеров, которые записаны подряд без пробелов. В случае положительного ответа выведите любое такое разбиение.

Каждый набор входных данных — непустая строка s , которая состоит из цифр и прописных букв латинского алфавита. Длина строки — от 1 до 50 символов. Выведите n строк: очередная строка должна содержать ответ для соответствующего набора входных данных. Если ответ отрицательный — то есть заданную строку s невозможно представить как последовательность номеров автомобилей — строка в выводе должна содержать единственный символ "-".

В случае положительного ответа выведите любое разбиение заданной строки s на последовательность корректных номеров. Каждый номер должен соответствовать одному из двух видов (см. условие). Номера разделяйте пробелами. Вы можете выводить произвольное количество пробелов и даже лишние пробелы после последнего номера.

Входные данные:

```
R48FAO00000000A99OKA99OK
R48FAO00000000A99OKA99O
A9PQ
A9PQA
A99AAA99AAA99AAA99AA
AP9QA
```

Выходные данные:

```
R48FA O00OO O0OO A99OK A99OK
-
A9PQ
-
A99AA A99AA A99AA A99AA
-
```

```
import re
import sys

def check_line(line):
    pattern = re.compile(r"([A-Z]\d{2}[A-Z]{2}|[A-Z]\d[A-Z]{2})")
    matches = pattern.findall(line)
    if not matches or "".join(matches) != line:
        print("-")
        return

    else:
        print(*matches)

def main():
    for line in sys.stdin:
        line = line.strip()
        if not line:
            break
        check_line(line)

if __name__ == "__main__":
    main()
```

4. Битва за кондиционер [15]

В офисе стоит кондиционер, на котором можно установить температуру от 15 до 30 градусов. В офис по очереди приходят n сотрудников. i -й из них желает температуру не больше или не меньше a_i .

После прихода каждого сотрудника определите, можно ли выставить температуру, которая удовлетворит всех в офисе.

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 10^3$) — количество сотрудников.

i -я из следующих n строк каждого набора входных данных содержит требование к температуре от i -го сотрудника: либо $\geq a_i$, либо $\leq a_i$ ($15 \leq a_i \leq 30$, a_i — целое число). Требование $\geq a_i$ означает, что i -й сотрудник желает температуру не ниже a_i ; требование $\leq a_i$ означает, что i -й сотрудник желает температуру не выше a_i .

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^3 .

Для каждого набора входных данных выведите n строк, i -я из которых содержит температуру, удовлетворяющую всех сотрудников с номерами от 1 до i включительно. Если такой температуры не существует, выведите -1 . После вывода ответа на очередной набор входных данных выводите пустую строку.

Если ответов несколько, выведите любой.

Входные данные:

```
4
1
>= 30
6
>= 18
<= 23
>= 20
<= 27
<= 21
>= 28
3
<= 25
>= 20
>= 25
3
<= 15
>= 30
<= 24
```

Выходные данные:

```
30
18
18
20
20
```

20

-1

15

20

25

15

-1

-1

```
def read_line_int():
    return int(input().strip())

def fight(n):
    min_val, max_val = 15, 30
    limiters = {'>': min_val, '<': max_val}

    for _ in range(n):
        adj = input().strip()
        num = int(adj[3:])
        operator = adj[0]

        if operator == '>' and num > limiters[operator]:
            limiters[operator] = num
        elif operator == '<' and num < limiters[operator]:
            limiters[operator] = num

        if limiters['>'] > limiters['<']:
            print(-1)
        else:
            print(limiters['>'])

def main():
    tasks = read_line_int()
    for _ in range(tasks):
        tasknum = read_line_int()
        fight(tasknum)

if __name__ == "__main__":
    main()
```