

## Link

<https://www.hackerrank.com/challenges/python-tuples/problem>

## Task

Given an integer,  $n$ , and  $n$  space-separated integers as input, create a tuple,  $t$ , of those integers. Then compute and print the result of `hash(t)`.

Note: `hash()` is one of the functions in the `__builtins__` module, so it need not be imported.

## Input Format

The first line contains an integer,  $n$ , denoting the number of elements in the tuple.

The second line contains  $n$  space-separated integers describing the elements in tuple  $t$ .

## Output Format

Print the result of `hash(t)`.

## Sample Input 0

```
2
1 2
```

## Sample Output 0

```
3713081631934410656
```

## Solution

```
if __name__ == '__main__':
    n = int(input())
    b=input()
    a = tuple(map(int, b.split(' ')))
    print(hash(a))
```

## Link

<https://www.hackerrank.com/challenges/list-comprehensions/problem>

### Task

You are given three integers  $x, y$  and  $z$  representing the dimensions of a cuboid along with an integer  $n$ . Print a list of all possible coordinates given by  $(i, j, k)$  on a 3D grid where the sum of  $i+j+k$  is not equal to  $n$ . Here,  $0 \leq i \leq x$ ;  $0 \leq j \leq y$ ;  $0 \leq k \leq z$ . Please use list comprehensions rather than multiple loops, as a learning exercise.

### Solution

```
if __name__ == '__main__':  
  
    x = int(input())  
  
    y = int(input())  
  
    z = int(input())  
  
    n = int(input())  
  
  
    newlist = [[i,j,k] for i in range(x+ 1) for j in range(y + 1) for k in range(z + 1) if i+j+k != n]  
  
  
    print(newlist)
```

### Link

<https://www.hackerrank.com/challenges/find-second-maximum-number-in-a-list>

### Task

Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. You are given  $n$  scores. Store them in a list and find the score of the runner-up.

---

### Input Format :

The first line contains  $n$ . The second line contains an array  $A[]$  of  $n$  integers each separated by a space.

### Constraints :

- $2 \leq n \leq 10$

- $-100 \leq A[i] \leq 100$

**Output Format :**

Print the runner-up score.

---

**Sample Input :**

5

2 3 6 6 5

**Sample Output :**

5

**Explanation :**

Given list is  $[2, 3, 6, 6, 5]$ . The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

---

**Solution :**

```
if __name__ == '__main__':  
    n = int(input())  
    arr = map(int, input().split())  
    print(sorted(list(set(arr)))[-2])
```

**Link**

<https://www.hackerrank.com/challenges/finding-the-percentage/problem>

## Task

The provided code stub will read in a dictionary containing key/value pairs of name:[marks] for a list of students. Print the average of the marks array for the student name provided, showing 2 places after the decimal.

Example:

```
marks key: value pairs are
'alpha': [20,30,40]
'beta': [30,50,70]
query_name = 'beta'
```

The query\_name is 'beta'. beta's average score is  $(20+50+70)/3 = 50$ .

Input Format:

The first line contains the integer n, the number of students' records. The next n lines contain the names and marks obtained by a student, each value separated by a space. The final line contains query\_name, the name of a student to query.

Constraints:

```
0<=n<=10
0<=marks[i]<=100
length of marks array=3
```

Output Format:

Print one line: The average of the marks obtained by the particular student correct to 2 decimal places.

**Solution :**

```
if __name__ == '__main__':
    n = int(input())
    student_marks = {}
    for _ in range(n):
        name, *line = input().split()
        scores = list(map(float, line))
        student_marks[name] = scores
    query_name = input()
    marks=student_marks[query_name]
    sum=0.00
    for i in range(0,3):
        sum+=marks[i]
    print("{:.2f}".format(sum/3))
```

## Link

<https://www.hackerrank.com/challenges/nested-list/problem>

## Task

Given the names and grades for each student in a class of  $N$  students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

Note: If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

## Example

```
records = [[ "chi", 20.0], [ "beta", 50.0], [ "alpha", 50.0]
```

The ordered list of scores is **[20.0, 50.0]**, so the second lowest score is **50.0**. There are two students with that score: **[ "beta", "alpha"]**. Ordered alphabetically, the names are printed as:

```
alpha
```

```
beta
```

## Input Format

The first line contains an integer,  $n$ , the number of students.

The  $n$  subsequent lines describe each student over  $n$  lines.

- The first line contains a student's name.
- The second line contains their grade.

## Constraints

- $2 \leq N \leq 5$
- There will always be one or more students having the second lowest grade.

## Output Format

Print the name(s) of any student(s) having the second lowest grade in. If there are multiple students, order their names alphabetically and print each one on a new line.

### Sample Input 0

```
5
Harry
37.21
Berry
37.21
Tina
37.2
Akriti
41
Harsh
39
```

### Sample Output 0

```
Berry
Harry
```

### Explanation 0

There are **5** students in this class whose names and grades are assembled to build the following list:

```
python students = [['Harry', 37.21], ['Berry', 37.21], ['Tina', 37.2], ['Akriti', 41], ['Harsh', 39]]
```

The lowest grade of **37.2** belongs to Tina. The second lowest grade of **37.21** belongs to both Harry and Berry, so we order their names alphabetically and print each name on a new line.

### Solution

```
Result = []
```

```
scorelist = []
```

```
if __name__ == '__main__':
```

```
    for _ in range(int(input())):
```

```
        name = input()
```

```
        score = float(input())
```

```
Result+=[[name,score]]
```

```
scorelist+= [score]
```

```
b=sorted(list(set(scorelist)))[1]
```

```
for a,c in sorted(Result):
```

```
    if c==b:
```

```
        print(a)
```

#### Link

<https://www.hackerrank.com/challenges/python-lists/problem>

#### Task

Consider a list (list = []). You can perform the following commands:

insert i e: Insert integer e at position i. print: Print the list. remove e: Delete the first occurrence of integer e.

append e: Insert integer e at the end of the list. sort: Sort the list. pop: Pop the last element from the list.

reverse: Reverse the list.

Initialize your list and read in the value of n followed by lines of commands where each command will be of the types listed above. Iterate through each command in order and perform the corresponding operation on your list.

Sample Input:

```
12 insert 0 5 insert 1 10 insert 0 6 print remove 6 append 9 append 1 sort print pop reverse print
```

#### Solution

```
if __name__ == '__main__':
```

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

```
    commands = {
```

```
        "insert": lambda x, y, z: x.insert(y, z),
```

```
        "print": lambda x: print(x),
```

```
        "remove": lambda x, y: x.remove(y),
```

```
        "append": lambda x, y: x.append(y),
```

```
        "sort": lambda x: x.sort(),
```

```
        "pop": lambda x: x.pop(),
```

```
        "reverse": lambda x: x.reverse(),
```

```
}  
out = []  
for i in range(N):  
    a = input()  
    split_a = a.split(' ')  
    command = split_a[0]  
    try:  
        commands[command](out, int(split_a[1]), int(split_a[2]))  
    except IndexError:  
        try:  
            commands[command](out, int(split_a[1]))  
        except IndexError:  
            commands[command](out)
```