

Ex. No. : 7.4

Date: 18.05.24

Register No.: 231901035

Name Nitheesh K K

Print repeated no

Given an array of integers `nums` containing `n + 1` integers where each integer is in the range `[1, n]` inclusive. There is only **one repeated number** in `nums`, return *this repeated number*. Solve the problem using [set](#).

Example 1:

Input: `nums = [1,3,4,2,2]`

Output: 2

Example 2:

Input: `nums = [3,1,3,4,2]`

Output: 3

For example:

Input	Result
1 3 4 4 2	4

Program:

```
n=input().split(" ")
n = list(n)
for i in range(len(n)):
    for j in range(i+1,len(n)):
        if n[i] == n[j]:
            print(n[i])
            exit(0)
```



	Input	Expected	Got	
✓	1 3 4 4 2	4	4	✓
✓	1 2 2 3 4 5 6 7	2	2	✓



Check Pair

Given a tuple and a positive integer k , the task is to find the count of distinct pairs in the tuple whose sum is equal to K .

Examples:

Input: $t = (5, 6, 5, 7, 7, 8)$, $K = 13$

Output: 2

Explanation:

Pairs with sum $K (= 13)$ are $\{(5, 8), (6, 7), (6, 7)\}$.

Therefore, distinct pairs with sum $K (= 13)$ are $\{(5, 8), (6, 7)\}$.

Therefore, the required output is 2.

For example:

Input	Result
1,2,1,2,5 3	1
1,2 0	0

Program:

```
def count_distinct_pairs(t, K):  
    distinct_pairs = set()  
    for i in range(len(t)):  
        for j in range(i + 1, len(t)):  
            if t[i] + t[j] == K:
```



```

        distinct_pairs.add((min(t[i], t[j]), max(t[i], t[j])))

    return len(distinct_pairs)

t_input = input()
t = tuple(map(int, t_input.split(',')))
K = int(input())
print(count_distinct_pairs(t, K))

```

	Input	Expected	Got	
✓	5,6,5,7,7,8 13	2	2	✓
✓	1,2,1,2,5 3	1	1	✓
✓	1,2 0	0	0	✓



08 – Dictionary



Ex. No. : 8.1

Date: 25.05.24

Register No.:231901035

Name: Nitheesh K K

Sort Dictionary by Values Summation

Give a dictionary with value lists, sort the keys by summation of values in value list.

Input : test_dict = {'Gfg' : [6, 7, 4], 'best' : [7, 6, 5]}

Output : {'Gfg': 17, 'best': 18}

Explanation : Sorted by sum, and replaced.

Input : test_dict = {'Gfg' : [8,8], 'best' : [5,5]}

Output : {'best': 10, 'Gfg': 16}

Explanation : Sorted by sum, and replaced.

Sample Input:

2

Gfg 6 7 4

Best 7 6 5

Sample Output

Gfg 17

Best 18

For example:

Input	Result
2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18



Program:

```
n = int(input())
d = {}
for i in range(n):
    s = input().split()
    d[s[0]] = list(map(int, s[1:]))
d1 = {k: sum(v) for k, v in d.items()}
sorted_d = dict(sorted(d1.items(), key=lambda x: x[1]))
for k, v in sorted_d.items():
    print(k, v)
```

	Input	Expected	Got	
✓	2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18	Gfg 17 Best 18	✓
✓	2 Gfg 6 6 Best 5 5	Best 10 Gfg 12	Best 10 Gfg 12	✓



Ex. No. : 8.2

Date: 25.05.24

Register No.: 231901035

Name: Nitheesh K K

Student Record

Create a student dictionary for n students with the student name as key and their test mark assignment mark and lab mark as values. Do the following computations and display the result.

1. Identify the student with the highest average score
2. Identify the student who has the highest Assignment marks
3. Identify the student with the Lowest lab marks
4. Identify the student with the lowest average score

Note:

If more than one student has the same score display all the student names

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

For example:

Input	Result
4	Ram
James 67 89 56	James Ram
Lalith 89 45 45	Lalith
Ram 89 89 89	Lalith
Sita 70 70 70	

Program:




```

n=int(input())
d={ }
for i in range(n):
    na=input().split()
    d[na[0]]=[int(na[1]),int(na[2]),int(na[3])]
    l=int(na[3])

```

```

h=0
for i in d:
    if h< sum(d[i]):
        h=sum(d[i])
        j=i
        h1=sum(d[i])
print(j)
h=0

```

```

for i in d:
    if(h<d[i][1]):
        h=d[i][1]
        j=i
for i in d:
    if(h==d[i][1]):
        print(i,end=" ")
l1=[]
k=[]
print()
for i in d:

```



```

if(l>d[i][2]):
    l=d[i][2]
    j=i
for i in d:
    if(l==d[i][2]):
        l1.append(i)
for i in range(-1,-len(l1)-1,-1):
    print(l1[i],end=" ")
print()

for i in d:
    if h1> sum(d[i]):
        h1=sum(d[i])
        j=i
print(j)

```

	Input	Expected	Got	
✓	4 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith	Ram James Ram Lalith Lalith	✓
✓	3 Raja 95 67 90 Aarav 89 90 90 Shadhana 95 95 91	Shadhana Shadhana Aarav Raja Raja	Shadhana Shadhana Aarav Raja Raja	✓

Ex. No. : 8.3

Date: 25.05.24

Register No.: 231901035

Name: Nitheesh K K

Scramble Score

In the game of Scrabble™, each letter has points associated with it. The total score of a word is the sum of the scores of its letters. More common letters are worth fewer points while less common letters are worth more points.

Write a program that computes and displays the Scrabble™ score for a word. Create a dictionary that maps from letters to point values. Then use the dictionary to compute the score.

A Scrabble™ board includes some squares that multiply the value of a letter or the value of an entire word. We will ignore these squares in this exercise.

The points associated with each letter are shown below:

Points Letters

1 A, E, I, L, N, O, R, S, T and U

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

Sample Input

REC

Sample Output

REC is worth 5 points.

For example:



Input	Result
REC	REC is worth 5 points.

Program:

```
def calculate_scrabble_score(word):
    # Dictionary mapping letters to points
    letter_points = {
        'A': 1, 'B': 3, 'C': 3, 'D': 2, 'E': 1, 'F': 4, 'G': 2, 'H': 4,
        'I': 1, 'J': 8, 'K': 5, 'L': 1, 'M': 3, 'N': 1, 'O': 1, 'P': 3,
        'Q': 10, 'R': 1, 'S': 1, 'T': 1, 'U': 1, 'V': 4, 'W': 4, 'X': 8,
        'Y': 4, 'Z': 10
    }

    score = 0
    for letter in word:
        letter = letter.upper()
        score += letter_points.get(letter, 0) # Add the points for each letter, defaulting to 0 if not
        found

    return score

word=input()
score = calculate_scrabble_score(word)
print(f'{ word} is worth {score} points.')
```

