

**Programming Lab Management System**  
 King Mongkut's Institute of Technology Ladkrabang

[Home](#)
[Exercise](#)
[Edit profile](#)
[Help](#)
[Log out](#)



ภาณุณี : 1  
 รหัสนักศึกษา : 63010224  
 นาม Chanwant Nuchyoo  
 ปีที่  
 วิศวกรรมคอมพิวเตอร์

### Chapter : 10 - item : 1 - หาค่า Binary Search

คะแนน : 2 / 2 ส่งเวลา 1 นาที

ให้ลองเขียน Binary Search โดยใช้ Recursive เพื่อหาค่าที่ซ่อนอยู่ใน list หรือไม่ ถ้าหากมีให้ตอบ True หากไม่มีให้ตอบ False

\*\*\*\*\*อธิบาย Input


- ค่าซ่อน จะเป็น list ของ Data
- ค่าค้นหา จะเป็นค่าที่เราต้องการค้นหา

```

def bi_search(l, r, arr, x):
    # code Here

inp = input('Enter Input : ').split('/')
arr, k = list(map(int, inp[0].split())), int(inp[1])
print(bi_search(0, len(arr) - 1, sorted(arr), k))
  
```


You have got full mark !!!

**Last submission :** 

```


1 ...
2 * ภาณุณี : 21010001
3 * 63010224 Chanwant Nuchyoo
4 * chapter : 10 item : 1 รหัสที่ : 0001
5 * Assigned : Monday 15th of November 2021 01:11:03 PM --> Submission : Sunday 21st of November 2021 10:29:23 PM
6 * Elapsed time : 9198 minutes.
7 * filename : 63010224_lab10_1.py
8 ...
9 def bi_search(l, r, arr, x):
10     if r >= l:
11         mid = (r + l) // 2
12
13         if arr[mid] == x:
14             return True
15         elif arr[mid] > x:
  
```

Number of testcase : 4

Testcase student: #1/4 1 


Enter Input : 33 2 11 82 77 28 15 76 9 64/28  
True

Enter Input : 33 2 11 82 77 28 15 76 9 64/28  
True


Testcase student: #2/4 2 

Enter Input : 33 2 11 82 77 28 15 76 9 64/50  
False

Enter Input : 33 2 11 82 77 28 15 76 9 64/50  
False

Testcase student: #3/4 3 

This testcase is hidden.

Testcase student: #4/4 4 

This testcase is hidden.

Page rendered in 0.2776 seconds. Codeigniter Version 3.1.2 Tue Nov 23 12:45:46



กลุ่มที่ : 1  
รหัสนักศึกษา : 63010224  
นาม Chanwant Nuchyoo  
ชื่อ  
โปรแกรมเมอร์

## Chapter : 10 - item : 2 - First Greater Value

จำนวน : 2 / 2

สถานะ : 1 หน้า

ให้เขียนโปรแกรมหาค่าที่น้อยที่สุดที่มากกว่าค่าที่เลือกจะหา ถ้าหากไม่มีให้แสดงว่า No First Greater Value โดยรับและส่งออก 2 list รับประกันว่าไม่เกิน 1000000

\*\*\*\* ตัวอย่าง Test Case 2:

Left : [3, 2, 7, 6, 8] Right : [5, 6, 12]

1. หากค่าที่น้อยที่สุดที่มากกว่า 5 จาก list (Left) จะได้เป็น 6

2. หากค่าที่น้อยที่สุดที่มากกว่า 6 จาก list (Left) จะได้เป็น 7

3. หากค่าที่น้อยที่สุดที่มากกว่า 12 จาก list (Left) จะเห็นว่าไม่มีค่าที่มากกว่า 12 จะแสดงเป็น No First Greater Value

You have got full mark !!!

## Last submission :



```

1 ...
2 * กลุ่มที่ : 21010001
3 * 63010224 Chanwant Nuchyoo
4 * chapter : 10 item : 2 รหัสที่ : 0001
5 * Assigned : Monday 15th of November 2021 04:02:19 PM --> Submission : Sunday 21st of November 2021 10:53:53 PM
6 * Elapsed time : 9051 minutes.
7 * filename : 63010224_Lab10_2.py
8 ...
9 inp, x = input("Enter Input : ").split("/")
10 x = list(map(int, x.split(" ")))
11 inp = list(map(int, inp.split(" ")))
12
13 for num in x:
14     answer = None
15     for i in inp:

```

Number of testcase : 4

Testcase student: #1/4 1


 Enter Input : 3 2 7 6 8/5  
 6

 Enter Input : 3 2 7 6 8/5  
 6

Testcase student: #2/4 2


 Enter Input : 3 2 7 6 8/5 6 12  
 6  
 7  
 No First Greater Value

 Enter Input : 3 2 7 6 8/5 6 12  
 6  
 7  
 No First Greater Value

Testcase student: #3/4 3




This testcase is hidden.

Testcase student: #4/4 4



This testcase is hidden.



hasht : 1  
รหัสนักเรียน : 63010224  
ชื่อ Charanwit Nuchyoo  
ชื่อ  
โรงเรียนขอนแก่น

### Chapter : 10 - item : 3 - Fun with hashing

คะแนน : 2 / 2

ให้เขียน Hashing โดยมีกำหนดดังนี้

- เรา index use table จากเลขของ ASCII จากค่า key จาก index มา mod จำนวนของ table
- ถ้าพบ Collision ให้ทำรายการ index ของ Quadratic Probing
- ถ้าพบ Collision ให้ลบค่าที่ตรงกันแล้ว ให้ทำ Quadratic Data ขึ้นที่ตรงกัน
- หาก Table หนึ่ง Data เต็มแล้วให้ลบค่าตรงกัน This table is full หากลบจนหมดแล้วค่าที่ลบไปไม่ลงจนลงอีก (ลบจนเหลือ 1 หรือ)

ตัวอย่าง Input

เลข Data (ให้ 2 ชุดก็ได้)

- ถ้าค่าที่ลบจนเหลือ จำนวนของ Table แล้ว table Collision ขาดค่าลบ
- ถ้าลบจนหมดแล้ว Data 0 up Two Data และลบจนหมดแล้ว column Two Data และลบจนหมดแล้ว key 0 up value ขาดค่าลบ

```

class Data:
    def __init__(self, key, value):
        self.key = key
        self.value = value

    def __str__(self):
        return "(0, {})".format(self.key, self.value)

class hash:
    @staticmethod
  
```

You have got full mark !!!

## Last submission :

```

1 ***
2 * กฎที่ : 23100001
3 * 63010224 Charanwit Nuchyoo
4 * chapter : 10 item : 3 - hasht : 0001
5 * Assigned : Sunday 21st of November 2021 18:54:02 PM --> Submission : Tuesday 23rd of November 2021 12:03:54 AM
6 * Elapsed time : 1009 minutes.
7 * filename : 63010224_Lab10_3.py
8 ***
9
10 class Data:
11     def __init__(self, key, value):
12         self.key = key
13         self.value = value
14
15     def __str__(self):
16         return "(0, {})".format(self.key, self.value)
  
```

Number of testcase : 3

## Testcase student: #1/3 1

```

**** fun with hashing ****
Enter Input : 3 2/141 T,Onf love,shdrd T,$$m2 KMTTI,kk KMTTI,7 love
#1 (141, 1)
#2 None
#3 None

collision number 1 at 0
#1 (141, 1)
#2 (Onf, love)
#3 None

collision number 1 at 0
collision number 2 at 1
How of collision chain
#1 (141, 1)
#2 (Onf, love)
#3 None

****
#1 (141, 1)
#2 (Onf, love)
#3 ($$m2, KMTTI)

this table is full !!!!!
  
```

```

**** fun with hashing ****
Enter Input : 3 2/141 T,Onf love,shdrd T,$$m2 KMTTI,kk KMTTI,7 love
#1 (141, 1)
#2 None
#3 None

collision number 1 at 0
#1 (141, 1)
#2 (Onf, love)
#3 None

collision number 1 at 0
collision number 2 at 1
How of collision chain
#1 (141, 1)
#2 (Onf, love)
#3 None

****
#1 (141, 1)
#2 (Onf, love)
#3 ($$m2, KMTTI)

this table is full !!!!!
  
```

## Testcase student: #2/3 2

```

**** fun with hashing ****
Enter Input : 5 5/one Un,Un,Dous,three Truis,four Quatre,five Cinq,ten Dix,eleven Onze
#1 None
#2 None
#3 (one, Un)
#4 None
#5 None

collision number 1 at 1
collision number 2 at 0
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 None
#5 None

collision number 1 at 1
collision number 2 at 2
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 None
#5 (four, Quatre)

collision number 1 at 1
collision number 2 at 2
collision number 3 at 0
collision number 4 at 0
How of collision chain
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 None
#5 (four, Quatre)

collision number 1 at 2
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 (ten, Dix)
#5 (four, Quatre)

this table is full !!!!!
  
```

```

**** fun with hashing ****
Enter Input : 5 5/one Un,Un,Dous,three Truis,four Quatre,five Cinq,ten Dix,eleven Onze
#1 None
#2 None
#3 (one, Un)
#4 None
#5 None

collision number 1 at 1
collision number 2 at 2
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 None
#5 None

collision number 1 at 1
collision number 2 at 2
collision number 3 at 0
collision number 4 at 0
How of collision chain
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 None
#5 (four, Quatre)

collision number 1 at 2
#1 (three, Truis)
#2 (two, Dous)
#3 (one, Un)
#4 (ten, Dix)
#5 (four, Quatre)

this table is full !!!!!
  
```

## Testcase student: #3/3 3

This testcase is hidden.



คุณชื่อ : 1  
 รหัสประจำตัว : 63010224  
 นาม Chanwanit Nuchoo  
 ชื่อ  
 นามสกุลและนามสกุล

## Chapter : 10 - item : 4 - Rehashing

คะแนน : 2 / 2

ดูประวัติ 1 ครั้ง

ให้ลองเขียนการทำ Rehashing ด้วยวิธีต่อไปนี้  
 1. Table เมื่อเต็มจะถึงกำหนด (Threshold (%))  
 2. เมื่อเกิดการ Collision จะทำการขยายขนาด  
 หากเกิดการ Rehashing ไม่ทำการขยาย Table เข้ามา prime ตัวใหม่ที่มากกว่าเดิม 2 เท่า เช่น หาก Table แรกมีขนาด 4 และเกิดการ Rehashing แล้ว Table ใหม่จะมีขนาดเป็น 11 เนื่องจาก 2 คูณ 4 คือ 8 และค่า prime ที่มากกว่า 8 และใกล้ 8 มากที่สุดคือ 11  
 การ Hash หากเกิดการ Collision ให้ใช้ Quadratic Probing ในการแก้ปัญหา Collision  
 ส่วนค่า input  
 และ Data เป็น 2 ชุดดังนี้ /  
 - ถ้าข้อมูลมาคือ ขนาดของ Table ,MaxCollision และ Threshold (สูงสุด 100 %) ตามลำดับ  
 - ถ้าข้อมูลมาคือ Data n ชุด โดย Data แต่ละชุดจะมี spacebar และ Data และส่วนเป็นจำนวนเต็มอยู่แค่ในวงเล็บเท่านั้น และมี Data จำนวนกี่ตัว

You have got full mark !!!

### Last submission :

```
1 ***
2 * รหัสที่ : 21010001
3 * 63010224 Chanwanit Nuchoo
4 * Chapter : 10 Item : 4 - รหัสที่ : 0001
5 * Assigned : Sunday 21st of November 2021 11:05:00 PM --> Submission : Tuesday 23rd of November 2021 12:05:43 PM
6 * Elapsed time : 1500 seconds.
7 * filename : 63010224_Lab10_4.py
8 ***
9
10 class HASH:
11     def __init__(self, size, _max, threshold):
12         self.size = size
13         self._max = _max
14         self.element = [None] * self.size
15         self.member = []
16         self.thr = threshold
```

Number of testcase : 7

Testcase student: #1/7 1

```
***** Rehashing *****
Enter Input : 5 1 0/73 6
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

-----
Add : 1
#1 None
#2 1
#3 None
#4 None
#5 None

-----
Add : 6
collision number 1 at 1
***** Max collision - Rehash !!! *****
#1 None
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None

-----
```

```
***** Rehashing *****
Enter Input : 5 1 0/73 6
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

-----
Add : 1
#1 None
#2 1
#3 None
#4 None
#5 None

-----
Add : 6
collision number 1 at 1
***** Max collision - Rehash !!! *****
#1 None
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None

-----
```

Testcase student: #2/7 2

```
***** Rehashing *****
Enter Input : 5 1 10/1 6
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

-----
Add : 1
***** Data over threshold - Rehash !!! *****
#1 None
#2 1
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None

-----
Add : 6
***** Data over threshold - Rehash !!! *****
#1 None
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None

-----
```

```
***** Rehashing *****
Enter Input : 5 1 10/1 6
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

-----
Add : 1
***** Data over threshold - Rehash !!! *****
#1 None
#2 1
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None

-----
```

```

***** Reshaking *****
Enter Input : 5 1 10/0 1 6 20
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

Add : 0
***** Data over threshold - Rehash !!! *****
#1 0
#2 None
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None

Add : 1
***** Data over threshold - Rehash !!! *****
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None

Add : 6
***** Data over threshold - Rehash !!! *****
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None
#24 None
#25 None
#26 None
#27 None
#28 None
#29 None
#30 None
#31 None
#32 None
#33 None
#34 None
#35 None
#36 None
#37 None
#38 None
#39 None
#40 None
#41 None
#42 None
#43 None
#44 None
#45 None
#46 None
#47 None

Add : 20
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 20
#21 None
#22 None
#23 None
#24 None
#25 None
#26 None
#27 None
#28 None
#29 None
#30 None
#31 None
#32 None
#33 None
#34 None
#35 None
#36 None
#37 None
#38 None
#39 None
#40 None
#41 None
#42 None
#43 None
#44 None
#45 None
#46 None
#47 None

```

```

***** Reshaking *****
Enter Input : 5 1 10/0 1 6 20
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None

Add : 0
***** Data over threshold - Rehash !!! *****
#1 0
#2 None
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None

Add : 1
***** Data over threshold - Rehash !!! *****
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 None
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None

Add : 6
***** Data over threshold - Rehash !!! *****
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 None
#21 None
#22 None
#23 None
#24 None
#25 None
#26 None
#27 None
#28 None
#29 None
#30 None
#31 None
#32 None
#33 None
#34 None
#35 None
#36 None
#37 None
#38 None
#39 None
#40 None
#41 None
#42 None
#43 None
#44 None
#45 None
#46 None
#47 None

Add : 20
#1 0
#2 1
#3 None
#4 None
#5 None
#6 None
#7 6
#8 None
#9 None
#10 None
#11 None
#12 None
#13 None
#14 None
#15 None
#16 None
#17 None
#18 None
#19 None
#20 20
#21 20
#22 None
#23 None
#24 None
#25 None
#26 None
#27 None
#28 None
#29 None
#30 None
#31 None
#32 None
#33 None
#34 None
#35 None
#36 None
#37 None
#38 None
#39 None
#40 None
#41 None
#42 None
#43 None
#44 None
#45 None
#46 None
#47 None

```

Testcase student: #4/7 4

```
***** Rehashing *****
Enter Input : 7 6 78/13 15 6 24 23
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 None
-----
Add : 13
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 15
#1 None
#2 15
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 6
collision number 1 at 6
#1 6
#2 15
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 24
#1 6
#2 15
#3 None
#4 24
#5 None
#6 None
#7 13
-----
Add : 23
***** Data over threshold - Rehash !!! *****
collision number 1 at 6
collision number 2 at 7
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 6
#8 24
#9 None
#10 None
#11 23
#12 None
#13 None
#14 13
#15 None
#16 15
#17 None
-----
```

```
***** Rehashing *****
Enter Input : 7 6 78/13 15 6 24 23
Initial Table :
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 None
-----
Add : 13
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 15
#1 None
#2 15
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 6
collision number 1 at 6
#1 6
#2 15
#3 None
#4 None
#5 None
#6 None
#7 13
-----
Add : 24
#1 6
#2 15
#3 None
#4 24
#5 None
#6 None
#7 13
-----
Add : 23
***** Data over threshold - Rehash !!! *****
collision number 1 at 6
collision number 2 at 7
#1 None
#2 None
#3 None
#4 None
#5 None
#6 None
#7 6
#8 24
#9 None
#10 None
#11 23
#12 None
#13 None
#14 13
#15 None
#16 15
#17 None
-----
```

Testcase student: #5/7 5

This testcase is hidden.

Testcase student: #6/7 6

This testcase is hidden.

[illegible][illegible]

กลุ่มที่ : 1  
รหัสวิชา : 63010224  
นาย Chanwanit Nuchyoo  
พี่  
วิศวกรและเขียนโค้ด

Programming Lab Management System  
King Mongkut's Institute of Technology Ladkrabang

Home Exercise Edit profile Help Log out

Chapter : 10 - item : 5 - กล่องสินค้า

สถานะ : 2 / 2 คำถามที่ 1 ของ 1

มีสินค้าอยู่  $n$  ชิ้น โดยชิ้นที่  $i$  ( $0 \leq i < n$ ) มีน้ำหนัก  $W[i]$  กิโลกรัม หากเลือกบรรจุกล่องสินค้า  $k$  ชิ้น โดยที่เลือกบรรจุ  
 1. มีของน้อยกว่าน้ำหนักบรรจุภัณฑ์ในกล่องสินค้า  
 2. หากมีของน้อยกว่า  $a$  และชิ้นที่  $b$  อยู่ในกล่องสินค้า ( $a \leq b$ ) มีของทุกชิ้นที่ใส่อยู่ในกล่องสินค้า (ทุกชิ้นที่  $i$  ที่  $a < i < b$ ) จะต้องมีของในกล่องสินค้า (นั่นคือเลือกบรรจุภัณฑ์ที่มีน้ำหนักน้อยกว่าหรือเท่ากับน้ำหนักของสินค้า)  
 ถ้าหากกล่องสินค้าที่บรรจุได้ทั้งหมด จะหาว่าสามารถใส่กล่องสินค้าได้กี่กล่องเท่าไร โดยที่เลือกบรรจุภัณฑ์ที่มีน้ำหนักน้อยกว่าหรือเท่ากับน้ำหนักของสินค้า

เขียน Input  
และ Data เป็น 2 ชุดคือ /  
 - ลำดับน้ำหนักสินค้า  $n$  ชิ้น และแต่ละชิ้นมีน้ำหนัก  $W[i]$  กิโลกรัม  
 - ลำดับน้ำหนักของกล่องสินค้า  $m$  ชิ้น

**Output Optimization Problem**

**เขียน Test Case #1**  
 มีสินค้าอยู่ 5 ชิ้น โดยที่น้ำหนักเป็น 6 2 4 3 7 หากใส่ในกล่องสินค้า 3 ชิ้น และน้ำหนักที่น้อยกว่าหรือเท่ากับน้ำหนักของสินค้าได้บรรจุทุกชิ้น และใส่กล่องสินค้าได้ทั้งหมด 8 กิโลกรัม โดยในกล่องที่ 1 จะใส่สินค้า 2 ชิ้นที่มีน้ำหนัก 6 และ 2 กล่องที่ 2 จะใส่สินค้า 2 ชิ้นที่มีน้ำหนัก 4 และ 3 และกล่องที่ 3 จะใส่สินค้า 1 ชิ้นที่มีน้ำหนัก 7

**เขียน Test Case #2**  
 มีสินค้าอยู่ 10 ชิ้น โดยที่น้ำหนักเป็น 8 7 2 5 1 10 9 2 3 5 หากใส่ในกล่องสินค้า 5 ชิ้น และน้ำหนักที่น้อยกว่าหรือเท่ากับน้ำหนักของสินค้าได้บรรจุทุกชิ้น และใส่กล่องสินค้าได้ทั้งหมด 14 กิโลกรัม โดยในกล่องที่ 1 จะใส่สินค้า 1 ชิ้นที่มีน้ำหนัก 8 กล่องที่ 2 จะใส่สินค้า 3 ชิ้นที่มีน้ำหนัก 7 2 และ 5 กล่องที่ 3 จะใส่สินค้า 2 ชิ้นที่มีน้ำหนัก 1 และ 10 กล่องที่ 4 จะใส่สินค้า 3 ชิ้นที่มีน้ำหนัก 9 2 และ 3 และกล่องที่ 5 จะใส่สินค้า 1 ชิ้นที่มีน้ำหนัก 5

You have got full mark !!!

Last submission :

```

1 ...
2 * กลุ่มที่ : 21010001
3 * 63010224 Chanwanit Nuchyoo
4 * chapter : 10 item : 5 รหัสที่ : 0001
5 * Assigned : Sunday 21st of November 2021 11:05:12 PM --> Submission : Tuesday 23rd of November 2021 12:07:19 AM
6 * Elapsed time : 1502 minutes.
7 * filename : 63010224_lab05_3.py
8 ...
9 if __name__ == "__main__":
10     #inp_ = "6 2 4 3 7/3".split("/")
11     inp_ = input("Enter Input : ").split("/")
12     inp = list(map(int, inp_[0].split()))
13     ans = 37
14     max_ = sum(inp)
15     min_ = max(inp)
    
```

Number of testcase : 8

Testcase student: #1/8 1	Enter Input : 6 2 4 3 7/3 Minimum weight for 3 box(es) = 8	Enter Input : 6 2 4 3 7/3 Minimum weight for 3 box(es) = 8
Testcase student: #2/8 2	Enter Input : 8 7 2 5 1 10 9 2 3 5/5 Minimum weight for 5 box(es) = 14	Enter Input : 8 7 2 5 1 10 9 2 3 5/5 Minimum weight for 5 box(es) = 14
Testcase student: #3/8 3	Enter Input : 19 1 2 3 4/1 Minimum weight for 1 box(es) = 29	Enter Input : 19 1 2 3 4/1 Minimum weight for 1 box(es) = 29
Testcase student: #4/8 4	Enter Input : 19 1 2 3 4/2 Minimum weight for 2 box(es) = 19	Enter Input : 19 1 2 3 4/2 Minimum weight for 2 box(es) = 19
Testcase student: #5/8 5	Enter Input : 6 4 9 3 1 8 5 2/5 Minimum weight for 5 box(es) = 10	Enter Input : 6 4 9 3 1 8 5 2/5 Minimum weight for 5 box(es) = 10
Testcase student: #6/8 6	This testcase is hidden.	
Testcase student: #7/8 7	This testcase is hidden.	
Testcase student: #8/8 8	This testcase is hidden.	

Page rendered in 8.2735 seconds. CodeIgniter Version 3.1.2 Tue Nov 23 12:50:31