

[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Experiments based on Variables, Datatypes in Python](#) / [Week1 Coding](#)

Started on Sunday, 17 March 2024, 2:48 PM

State Finished

Completed on Sunday, 17 March 2024, 9:52 PM

Time taken 7 hours 4 mins

Marks 6.00/6.00

Grade **100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Write a program to convert strings to an integer and float and display its type.

Sample Input:

10

10.9

Sample Output:

10,<class 'int'>

10.9,<class 'float'>

For example:

Input	Result
10	10,<class 'int'>
10.9	10.9,<class 'float'>

Answer: (penalty regime: 0 %)

```

1 s=input()
2 s1=input()
3 a=int(s)
4 b=float(s1)
5 b=round(b,1)
6 print(a,end="")
7 print(", ",end="")
8 print(type(a))
9 print(b,end="")
10 print(", ",end="")
11 print(type(b))
```

	Input	Expected	Got	
✓	10 10.9	10,<class 'int'> 10.9,<class 'float'>	10,<class 'int'> 10.9,<class 'float'>	✓
✓	12 12.5	12,<class 'int'> 12.5,<class 'float'>	12,<class 'int'> 12.5,<class 'float'>	✓
✓	89 7.56	89,<class 'int'> 7.6,<class 'float'>	89,<class 'int'> 7.6,<class 'float'>	✓
✓	55000 56.2	55000,<class 'int'> 56.2,<class 'float'>	55000,<class 'int'> 56.2,<class 'float'>	✓
✓	2541 2541.679	2541,<class 'int'> 2541.7,<class 'float'>	2541,<class 'int'> 2541.7,<class 'float'>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Ramesh's basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and his house rent allowance is 20% of his basic salary. Write a program to calculate his gross salary.

Sample Input:

10000

Sample Output:

16000

For example:

Input	Result
10000	16000

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=a*0.40
3 c=a*0.20
4 d=a+b+c
5 print(d)

```

	Input	Expected	Got	
✓	10000	16000	16000.0	✓
✓	20000	32000	32000.0	✓
✓	28000	44800	44800.0	✓
✓	5000	8000	8000.0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Write a simple python program to find the square root of a given floating point number. The output should be displayed with 3 decimal places.

Sample Input:

8.00

Sample Output:

2.828

For example:

Input	Result
14.00	3.742

Answer: (penalty regime: 0 %)

```

1 a=float(input())
2 b=a**0.5
3 print("%.3f"%b)

```

	Input	Expected	Got	
✓	8.00	2.828	2.828	✓
✓	14.00	3.742	3.742	✓
✓	4.00	2.000	2.000	✓
✓	487	22.068	22.068	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Alfred buys an old scooter for Rs. X and spends Rs. Y on its repairs. If he sells the scooter for Rs. Z ($Z > X + Y$). Write a program to help Alfred to find his gain percent. Get all the above-mentioned values through the keyboard and find the gain percent.

Input Format:

The first line contains the Rs X

The second line contains Rs Y

The third line contains Rs Z

Sample Input:

10000

250

15000

Sample Output:

46.34 is the gain percent.

For example:

Input	Result
45500	30.43 is the gain percent.
500	
60000	

Answer: (penalty regime: 0 %)

```

1 x=int(input())
2 y=int(input())
3 z=int(input())
4 a=z-(x+y)
5 b=(a/(x+y))*100
6 print("%.2f"%b,"is the gain percent.")

```

	Input	Expected	Got	
✓	10000 250 15000	46.34 is the gain percent.	46.34 is the gain percent.	✓
✓	45500 500 60000	30.43 is the gain percent.	30.43 is the gain percent.	✓

	Input	Expected	Got	
✓	5000 0 7000	40.00 is the gain percent.	40.00 is the gain percent.	✓
✓	12500 5000 18000	2.86 is the gain percent.	2.86 is the gain percent.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

In many jurisdictions, a small deposit is added to drink containers to encourage people to recycle them. In one particular jurisdiction, drink containers holding one liter or less have a \$0.10 deposit and drink containers holding more than one liter have a \$0.25 deposit. Write a program that reads the number of containers of each size(less and more) from the user. Your program should continue by computing and displaying the refund that will be received for returning those containers. Format the output so that it includes a dollar sign and always displays exactly two decimal places.

Sample Input

10

20

Sample Output

Your total refund will be \$6.00.

For example:

Input	Result
20	Your total refund will be \$7.00.
20	

Answer: (penalty regime: 0 %)

```

1 x=int(input())
2 y=int(input())
3 z=(x*0.10)+(y*0.25)
4 print("Your total refund will be ${z:.2f}"".")
```

	Input	Expected	Got	
✓	20 20	Your total refund will be \$7.00.	Your total refund will be \$7.00.	✓
✓	11 22	Your total refund will be \$6.60.	Your total refund will be \$6.60.	✓
✓	123 200	Your total refund will be \$62.30.	Your total refund will be \$62.30.	✓
✓	76 38	Your total refund will be \$17.10.	Your total refund will be \$17.10.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Justin is a carpenter who works on an hourly basis. He works in a company where he is paid Rs 50 for an hour on weekdays and Rs 80 for an hour on weekends. He works 10 hrs more on weekdays than weekends. If the salary paid for him is given, write a program to find the number of hours he has worked on weekdays and weekends.

Hint:

If the final result(hrs) are in -ve convert that to +ve using abs() function

The `abs()` function returns the absolute value of the given number.

```
number = -20
absolute_number = abs(number)
print(absolute_number)
# Output: 20
```

Sample Input:

450

Sample Output:

weekdays 10.38

weekend 0.38

For example:

Input	Result
450	weekdays 10.38 weekend 0.38

Answer: (penalty regime: 0 %)

```
1 y=float(input())
2 x=(y-500)/130
3 x=abs(x)
4 a=x+10
5 print("weekdays",f"{a:.2f}")
6 print("weekend",f"{x:.2f}")
```

	Input	Expected	Got	
✓	450	weekdays 10.38 weekend 0.38	weekdays 10.38 weekend 0.38	✓
✓	500	weekdays 10.00 weekend 0.00	weekdays 10.00 weekend 0.00	✓

	Input	Expected	Got	
✓	10000	weekdays 83.08 weekend 73.08	weekdays 83.08 weekend 73.08	✓
✓	6789	weekdays 58.38 weekend 48.38	weekdays 58.38 weekend 48.38	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week1_Quiz](#)

Jump to...

[Operators ►](#)

An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of gizmos from the user. Then your program should compute and display the total weight of the parts.

Sample Input:

10

20

Sample Output:

The total weight of all these widgets and gizmos is 2990 grams.

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 c=a*75
4 d=b*112
5 e=c+d
6 print("The total weight of all these widgets and gizmos is",e,"grams." )
```

	Input	Expected	Got	
✓	10 20	The total weight of all these widgets and gizmos is 2990 grams.	The total weight of all these widgets and gizmos is 2990 grams.	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Input	Result
6/19/24, 8:35 PM	
197	7
-197	7

Answer: (penalty regime: 0 %)

```
1 a=input()
2 print(a[-1])
```

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Input Format:
6/19/24, 8:35 PM

Week2_Coding: Attempt review | REC-PS

Input consists of two integers that correspond to the age and weight of a person respectively.

Output Format:

Display True(IF ELIGIBLE)

Display False (if not eligible)

Sample Input

19

45

Sample Output

True

For example:

Input	Result
18	False
40	

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 print(a>=18 and b>40)
```


True if he can buy that packet and False if he can't buy that packet.

SAMPLE INPUT AND OUTPUT:
6/19/24, 8:35 PM

Week2_Coding: Attempt review | REC-PS

5

25

12

10

9

OUTPUT

True False True False

For example:

Input	Result
5	True False True True
25	
23	
20	
10	

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 c=int(input())
4 d=int(input())
5 e=int(input())
6 if b%a==0:
7     print("True",end=" ")
8 else:
9     print("False",end=" ")
10 if c%a==0:
11     print("True",end=" ")
12 else:
13     print("False",end=" ")
14 if d%a==0:
15     print("True",end=" ")
16 else:
17     print("False",end=" ")
18 if e%a==0:
19     print("True",end=" ")
20 else:
21     print("False",end=" ")
```

Correct

Marks for this submission: 1.00/1.00.

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=a+(a*0.04)
3 print(f'Balance as of end of Year 1: ${b:.2f}.')
4 d=b+(b*0.04)
5 print(f'Balance as of end of Year 2: ${d:.2f}.')
6 f=d+(d*0.04)
7 print(f'Balance as of end of Year 3: ${f:.2f}.')
8
9

```

	Input	Expected	Got	
✓	10000	Balance as of end of Year 1: \$10400.00. Balance as of end of Year 2: \$10816.00. Balance as of end of Year 3: \$11248.64.	Balance as of end of Year 1: \$10400.00. Balance as of end of Year 2: \$10816.00. Balance as of end of Year 3: \$11248.64.	✓
✓	20000	Balance as of end of Year 1: \$20800.00. Balance as of end of Year 2: \$21632.00. Balance as of end of Year 3: \$22497.28.	Balance as of end of Year 1: \$20800.00. Balance as of end of Year 2: \$21632.00. Balance as of end of Year 3: \$22497.28.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

For example:

Input	Result
100	The tax is 5.00 and the tip is 18.00, making the total 123.00

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=a*0.05
3 c=a*0.18
4 d=a+b+c
5 print("The tax is",f'{b:.2f}',"and the tip is",f'{c:.2f}","",",making the total",f'{d:.2f}')

```

	Input	Expected	Got	
✓	100	The tax is 5.00 and the tip is 18.00, making the total 123.00	The tax is 5.00 and the tip is 18.00, making the total 123.00	✓
✓	250	The tax is 12.50 and the tip is 45.00, making the total 307.50	The tax is 12.50 and the tip is 45.00, making the total 307.50	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Sample Input:

6/19/24, 8:35 PM

32

43

Sample Output:

False

For example:

Input	Result
32	False
43	

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 print(a%3==0 and b%2==0)

```

	Input	Expected	Got	
✓	32 43	False	False	✓
✓	273 7890	True	True	✓
✓	800 4590	False	False	✓

output a single character "C" or "D" depending on the value of x.

6/19/24 10:35 PM

Week2_Coding: Attempt review | REC-PS

0
Output 1:
C

Input 2:
1

Output 1:
D

For example:

Input	Result
0	C

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 if a==0:
3     print("C")
4 elif a==1:
5     print("D")
```


Explanation:

6/19/2024 8:25 PM an even number and a number between 0 and 100, Week 3 Coding: Attempt review | REC-PS

For example:

Input	Result
101	False

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 print(a%2==0)
```

	Input	Expected	Got	
✓	56	True	True	✓
✓	101	False	False	✓
✓	-1	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Input	Result
3	2

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=(bin(a)).count("1")
3 print(b)
```

	Input	Expected	Got	
✓	3	2	2	✓
✓	5	2	2	✓
✓	15	4	4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week2_MCQ](#)

Jump to...

[Selection control structures ►](#)

6/19/24, 8:36 PM

Week3_coding: Attempt review | REC-PS

Started on Sunday, 7 April 2024, 6:44 PM**State** Finished**Completed on** Sunday, 7 April 2024, 9:09 PM**Time taken** 2 hours 25 mins**Marks** 10.00/10.00**Grade** **100.00** out of 100.00

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third.

For example, 3, 5 and 4 form a Pythagorean triple, since $3^2 + 4^2 = 25 = 5^2$

You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters.

Sample Input

3
5
4

Sample Output

yes

Sample Test Cases

Test Case 1

Input

3
5
4

Output

yes

Test Case 2

Input

5
8
2

Output

no

Answer: (penalty regime: 0 %)

```
1 a=int(input())  
2 b=int(input())  
3 c=int(input())  
4 if a**2+b**2==c**2 or b**2+c**2==a**2 or c**2+a**2==b**2:  
5     print("yes")  
6 else:  
7     print("no")
```

	5		
✓	5	no	no ✓
	8		
	2		

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a program to calculate and print the Electricity bill where the unit consumed by the user is given from test case. It prints the total amount the customer has to pay. The charge are as follows:

Unit	Charge / Unit
Upto 199	@1.20
200 and above but less than 400	@1.50
400 and above but less than 600	@1.80
600 and above	@2.00

If bill exceeds Rs.400 then a surcharge of 15% will be charged and the minimum bill should be of Rs.100/-

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

For example:

Input	Result
100.00	120.00
500	1035.00

Answer: (penalty regime: 0 %)

```

1 u=float(input())
2 cpu=0
3 if u<=199:
4     cpu=1.20
5 elif u<=399:
6     cpu=1.50
7 elif u<=599:
8     cpu=1.80
9 else:
10    cpu=2.00
11
12 tb=u*cpu
13
14 if tb>400:
15     tb+=tb*0.15
16
17 if tb<100:
18     tb=100
19
20 print(tb)
21

```

6/19/24,	8:36	100.00	120.00	120.0	✓
		500	1035.00	1035.0	✓
		700	1610.00	1610.0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

IN / OUT

Ms. Sita, the faculty handling programming lab for you is very strict. Your seniors have told you that she will not allow you to enter the week's lab if you have not completed atleast half the number of problems given last week. Many of you didn't understand this statement and so they requested the good programmers from your batch to write a program to find whether a student will be allowed into a week's lab given the number of problems given last week and the number of problems solved by the student in that week.

Input Format:

Input consists of 2 integers.

The first integer corresponds to the number of problems given and the second integer corresponds to the number of problems solved.

Output Format:

Output consists of the string "IN" or "OUT".

Sample Input and Output:

Input

8

3

Output

OUT

For example:

Input	Result
8	OUT
3	

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 if b>=((int)(a/2)):
4     print("IN")
5 else:
6     print("OUT")

```

	Input	Expected	Got	
✓	8 3	OUT	OUT	✓
✓	8 5	IN	IN	✓
✓	20 9	OUT	OUT	✓
✓	50 31	IN	IN	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

The Chinese zodiac assigns animals to years in a 12 year cycle. One 12 year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the dragon, and 1999 being another year of the hare.

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

Answer: (penalty regime: 0 %)

```

1 def chinese_zodiac(year):
2     #list of animals in the chinese zodiac
3     zodiac_animals={0:"Monkey" ,1:"Rooster",2:"Dog",3:"Pig",4:"Rat" ,5:"Ox" ,6:"Tiger" ,7:"Hare" ,8:"Dragon",9:"Sheep",10:"Monkey",11:"Rooster",12:"Dog"}
4     #Calculate the index of the zodiac animal for the given year
5     animal_index=year%12
6     #Display the corresponding animal
7     if year>=0:
8         return zodiac_animals[animal_index]
9     else:
10        return "invalid year input"
11 year_input=int(input())
12 print(year_input," is the year of the ",(chinese_zodiac(year_input)),sep="",end=".")
```

	Input	Expected	Got	
✓	2010	2010 is the year of the Tiger.	2010 is the year of the Tiger.	✓

Passed all tests! ✓

6/19/24, 8:36 PM

Week3_coding: Attempt review | REC-PS

Correct

Marks for this submission: 1.00/1.00.

Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years. The rules for determining whether or not a year is a leap year follow:

- Any year that is divisible by 400 is a leap year.
- Of the remaining years, any year that is divisible by 100 is not a leap year.
- Of the remaining years, any year that is divisible by 4 is a leap year.
- All other years are not leap years.

Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

Sample Input 1

1900

Sample Output 1

1900 is not a leap year.

Sample Input 2

2000

Sample Output 2

2000 is a leap year.

Answer: (penalty regime: 0 %)

```

1 def is_leap_year(year):
2     if (year%400==0) or (year%4==0 and year%100!=0):
3         return True
4     else:
5         return False
6 year=int(input())
7 if is_leap_year(year):
8     print(year,"is a leap year.")
9 else:
10    print(year,"is not a leap year.")

```

	Input	Expected	Got	
✓	1900	1900 is not a leap year.	1900 is not a leap year.	✓
✓	2000	2000 is a leap year.	2000 is a leap year.	✓
✓	2100	2100 is not a leap year.	2100 is not a leap year.	✓
✓	2020	2020 is a leap year.	2020 is a leap year.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths \geq 65

Marks in Physics \geq 55

Marks in Chemistry \geq 50

Or

Total in all three subjects \geq 180

Sample Test Cases

Test Case 1

Input

70

60

80

Output

The candidate is eligible

Test Case 2

Input

50

80

80

Output

The candidate is eligible

Test Case 3

Input

50

60

40

Output

The candidate is not eligible

For example:

Input	Result
70	The candidate is eligible
60	
80	

Answer: (penalty regime: 0 %)

```

1 M=int(input())
2 P=int(input())
3 C=int(input())
4 MP = M + P
5 MC = M + C
6 PC = P + C
7 MPC = M + P + C
8 if (M>=65 and P>=55 and C>=50 and MPC>=180) or (MP > 130 or PC > 120):
9     print('The candidate is eligible')
10 else:
11     print('The candidate is not eligible')

```

	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓
✓	50 60 40	The candidate is not eligible	The candidate is not eligible	✓
✓	20 10 25	The candidate is not eligible	The candidate is not eligible	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

In this exercise you will create a program that reads a letter of the alphabet from the user. If the user enters a, e, i, o or u then your program should display a message indicating that the entered letter is a vowel. If the user enters y then your program should display a message indicating that sometimes y is a vowel, and sometimes y is a consonant. Otherwise your program should display a message indicating that the letter is a consonant.

Sample Input 1

i

Sample Output 1

It's a vowel.

Sample Input 2

y

Sample Output 2

Sometimes it's a vowel... Sometimes it's a consonant.

Sample Input3

c

Sample Output 3

It's a consonant.

For example:

Input	Result
y	Sometimes it's a vowel... Sometimes it's a consonant.
c	It's a consonant.

Answer: (penalty regime: 0 %)

```

1 vowels=['a','e','i','o','u']
2 letter=str(input())
3 if letter in vowels:
4     print("It's a vowel.")
5 elif letter == "y":
6     print("Sometimes it's a vowel... Sometimes it's a consonant.")
7 else:
8     print("It's a consonant.")

```

	Input	Expected	Got	
✓	i	It's a vowel.	It's a vowel.	✓
✓	y	Sometimes it's a vowel... Sometimes it's a consonant.	Sometimes it's a vowel... Sometimes it's a consonant.	✓
✓	c	It's a consonant.	It's a consonant.	✓

Correct

Marks for this submission: 1.00/1.00.

A triangle can be classified based on the lengths of its sides as equilateral, isosceles or scalene. All three sides of an equilateral triangle have the same length. An isosceles triangle has two sides that are the same length, and a third side that is a different length. If all of the sides have different lengths then the triangle is scalene.

Write a program that reads the lengths of the three sides of a triangle from the user. Then display a message that states the triangle's type.

Sample Input 1

60

60

60

Sample Output 1

That's a equilateral triangle

Sample Input 2

40

40

80

Sample Output 2

That's a isosceles triangle

Sample Input 3

50

60

70

Sample Output 3

That's a scalene triangle

For example:

Input	Result
60 60 60	That's a equilateral triangle
40 40 80	That's a isosceles triangle

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=int(input())
4 if (a==b==c):
5     print("That's a equilateral triangle")
6 elif (a==b) or (b==c) or (c==a):
7     print("That's a isosceles triangle")
8 else:
9     print("That's a scalene triangle")

```

	Input	Expected	Got	
✓	60 60 60	That's a equilateral triangle	That's a equilateral triangle	✓
✓	40 40 80	That's a isosceles triangle	That's a isosceles triangle	✓
✓	50 60 70	That's a scalene triangle	That's a scalene triangle	✓
✓	50 50 80	That's a isosceles triangle	That's a isosceles triangle	✓
✓	10 10 10	That's a equilateral triangle	That's a equilateral triangle	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a program that returns the second last digit of the given number. Second last digit is being referred to the digit in the tens place in the given number.

For example, if the given number is 197, the second last digit is 9.

Note1 - The second last digit should be returned as a positive number. i.e. if the given number is -197, the second last digit is 9.

Note2 - If the given number is a single digit number, then the second last digit does not exist. In such cases, the program should return -1. i.e. if the given number is 5, the second last digit should be returned as -1

For example:

Input	Result
197	9
5	-1

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 def second_last_digit(number):
3     #convert the number to a positive integer
4     num=abs(number)
5     #check if the number has more than one digit
6     if num<10:
7         return -1 #Return -1 if the number is a single digit
8     #Extract the second last digit (tens place)
9     second_last=(num//10)%10
10    return second_last
11 #Test cases
12 print (second_last_digit(n))

```

	Input	Expected	Got	
✓	197	9	9	✓
✓	-197	9	9	✓
✓	5	-1	-1	✓
✓	123456	5	5	✓
✓	8	-1	-1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display "28 or 29 days" for February so that leap years are addressed.

Sample Input 1

February

Sample Output 1

February has 28 or 29 days in it.

Sample Input 2

March

Sample Output 2

March has 31 days in it.

Sample Input 3

April

Sample Output 3

April has 30 days in it.

For example:

Input	Result
February	February has 28 or 29 days in it.

Answer: (penalty regime: 0 %)

```

1 def days_in_month(month):
2     if month.lower() in ["january", "march", "may", "july", "august", "october", "december"]:
3         return 31
4     elif month.lower() in ["april", "june", "september", "november"]:
5         return 30
6     elif month.lower() == "february":
7         return "28 or 29"
8     else:
9         return "Invalid month name"
10 month_name = input()
11 print(month_name.capitalize(), "has", days_in_month(month_name), "days in it. ")

```

	Input	Expected	Got	
✓	February	February has 28 or 29 days in it.	February has 28 or 29 days in it.	✓
✓	March	March has 31 days in it.	March has 31 days in it.	✓
✓	April	April has 30 days in it.	April has 30 days in it.	✓
✓	May	May has 31 days in it.	May has 31 days in it.	✓

[◀ Week3_mcq](#)

Jump to...

[Iteration control structures ►](#)

[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Algorithmic Approach: Iteration control structures.](#) / [Week4 Coding](#)

Started on Wednesday, 10 April 2024, 1:54 PM

State Finished

Completed on Thursday, 11 April 2024, 11:29 AM

Time taken 21 hours 35 mins

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

In mathematics, the factorial of a non-negative integer n , denoted by $n!$, is the product of all positive integers less than or equal to n . For example,

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$$

Write a program to find the factorial of a given number.

The given number will be passed to the program as an input of type int.

The program is expected to calculate the factorial of the given number and return it as an int type.

Assumptions for this program:

The given input number will always be greater than or equal to 1.

Due to the range supported by int. the input numbers will range from 1 to 12.

For example:

Input	Result
5	120
4	24
9	362880

Answer: (penalty regime: 0 %)

```

1 def factorial(n):
2     if n ==0:
3         return 1
4     else:
5         return n *factorial(n-1)
6 num=int(input())
7 print(factorial(num))

```

	Input	Expected	Got	
✓	5	120	120	✓
✓	4	24	24	✓
✓	9	362880	362880	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Write a program to return the nth number in the fibonacci series.

The value of N will be passed to the program as input.

NOTE: Fibonacci series looks like –

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, . . . and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

- first Fibonacci number is 0,
- second Fibonacci number is 1,
- third Fibonacci number is 1,
- fourth Fibonacci number is 2,
- fifth Fibonacci number is 3,
- sixth Fibonacci number is 5,
- seventh Fibonacci number is 8, and so on.

For example:

Input	Result
1	0
4	2
7	8

Answer: (penalty regime: 0 %)

```

1 def fib_num(n):
2     if n<=0:
3         print("fibonacci can't be computed")
4     #First fibonacci number
5     elif n==1:
6         return 0
7     #second fibonacci number
8     elif n==2:
9         return 1
10    else:
11        return fib_num(n-1)+fib_num(n-2)
12 #input
13 n=int(input())
14 print(fib_num(n))

```

	Input	Expected	Got	
✓	1	0	0	✓
✓	4	2	2	✓
✓	7	8	8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Given an integer N, check whether N the given number can be made a perfect square after adding to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

For example:

Input	Result
24	Yes

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 import math as mt
3 def is_perfect_square(x):
4     sr=mt.sqrt(x)
5     return((sr-mt.floor(sr))==0)
6 def is_any_num(n):
7     if(is_perfect_square(n+1)):
8         return True
9     return False
10 if (is_any_num(n)):
11     print("Yes")
12 else:
13     print("No")

```

	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non-repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-repeated digits in this number.

For example:

Input	Result
292	1
1015	2
108	3
22	0

Answer: (penalty regime: 0 %)

```

1 def countUniqueDigits(N):
2     res = 0
3     cnt = [0] * 10
4     while (N > 0):
5         rem = N % 10
6         cnt[rem] += 1
7         N = N // 10
8     for i in range (10):
9         if (cnt[i]) == 1 :
10             res += 1
11     return res
12 N= int(input())
13 print(countUniqueDigits(N))

```

	Input	Expected	Got	
✓	292	1	1	✓
✓	1015	2	2	✓
✓	108	3	3	✓
✓	22	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Given a positive integer N, check whether it can be represented as a product of single digit numbers.

Input Format:

Single Integer input.

Output Format:

Output displays Yes if condition satisfies else prints No.

Example Input:

14

Output:

Yes

Example Input:

13

Output:

No

Answer: (penalty regime: 0 %)

```

1 N=int(input())
2 can_be_represented=False
3 for i in range (1,10):
4     if N%i==0 and N//i<10:
5         can_be_represented=True
6 if can_be_represented:
7     print("Yes")
8 else:
9     print("No")

```

	Input	Expected	Got	
✓	14	Yes	Yes	✓
✓	13	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Write a program that finds whether the given number N is Prime or not.

If the number is prime, the program should return 2 else it must return 1.

Assumption: $2 \leq N \leq 5000$, where N is the given number.

Example1: if the given number N is 7, the method must return 2

Example2: if the given number N is 10, the method must return 1

For example:

Input	Result
7	2
10	1

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 if n > 1 :
3     for i in range (2,n):
4         if (n % i) == 0:
5             print("1")
6             break
7     else:
8         print("2")

```

	Input	Expected	Got	
✓	7	2	2	✓
✓	10	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

Correct

Mark 1.00 out of 1.00

Given a number N, find the next perfect square greater than N.

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

Answer: (penalty regime: 0 %)

```
1 N=int(input())
2 root=int(N**0.5)+1
3 result=root*root
4 print(result)
5
6
```

	Input	Expected	Got	
✓	10	16	16	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8

Correct

Mark 1.00 out of 1.00

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

Input Format:

Single Integer Input from stdin.

Output Format:

Yes or No.

Example Input:

175

Output:

Yes

Explanation

$$1^1 + 7^2 + 5^3 = 175$$

Example Input:

123

Output:

No

For example:

Input	Result
175	Yes
123	No

Answer: (penalty regime: 0 %)

```

1 num=int(input())
2 power_sum=0
3 count=len(str(num))
4 temp=num
5 while temp > 0:
6     digit=temp%10
7     power_sum+=digit**count
8     temp//=10
9     count-=1
10 if num==power_sum:
11     print("Yes")
12 else:
13     print("No")

```

	Input	Expected	Got	
✓	175	Yes	Yes	✓
✓	123	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9

Correct

Mark 1.00 out of 1.00

Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits '2' and '9' in this number

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

For example:

Input	Result
292	2
1015	3

Answer: (penalty regime: 0 %)

```

1 num=input()
2 unique_digits=set(num)
3 print(len(unique_digits))

```

	Input	Expected	Got	
✓	292	2	2	✓
✓	1015	3	3	✓
✓	123	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10

Correct

Mark 1.00 out of 1.00

Write a program to find the sum of the series $1 + 11 + 111 + 1111 + \dots + n$ terms (n will be given as input from the user and sum will be the output)

Sample Test Cases**Test Case 1****Input**

4

Output

1234

Test Case 2**Input**

6

Output

123456

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 sum_series=0
3 for i in range(1,n+1):
4     term=int('1'*i)
5     sum_series+=term
6 print(sum_series)

```

	Input	Expected	Got	
✓	4	1234	1234	✓
✓	6	123456	123456	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week4_mcq](#)

Jump to...

[Strings ►](#)

[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Experiments based on Strings and its operations.](#) / [Week5 Coding](#)

Started on Wednesday, 8 May 2024, 1:15 PM

State Finished

Completed on Tuesday, 14 May 2024, 8:00 PM

Time taken 6 days 6 hours

Overdue 4 days 6 hours

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter. If balanced display as "true", otherwise "false".

For example:

Input	Result
Yn PYnative	True

Answer: (penalty regime: 0 %)

```

1 | s1=input()
2 | s2=input()
3 v if s1[::-1] in s2[::-1]:
4 |     print (True)
5 v else:
6 |     print(False)

```

	Input	Expected	Got	
✓	Yn PYnative	True	True	✓
✓	Ynf PYnative	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Reverse a string without affecting special characters

Given a string **S**, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

Input:

A&B

Output:

B&A

Explanation: As we ignore '&' and

As we ignore '&' and then reverse, so answer is "B&A".

For example:

Input	Result
A&x#	x&A#

Answer: (penalty regime: 0 %)

```

1 s=input()
2 sc=[c for c in s if not c.isalnum()]
3 rs=''.join(c for c in s if c.isalnum())[::-1]
4 result=''
5 si=0
6 for c in s:
7     if not c.isalnum():
8         result+= sc[si]
9         si+=1
10    else:
11        result+=rs[0]
12        rs=rs[1:]
13 print(result)
14
15
16
17

```

	Input	Expected	Got	
✓	A&B	B&A	B&A	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

first

second

first

third

second

then your program should display:

first

second

third

Answer: (penalty regime: 0 %)

```

1 s=list()
2 a=" "
3 while True:
4     i=input()
5     if i[-1] is a:
6         break
7     else:
8         if i not in s:
9             s.append(i)
10    for i in s:
11        print (i)

```

	Input	Expected	Got	
✓	first second first third second	first second third	first second third	✓
✓	rec cse it rec cse	rec cse it	rec cse it	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

```
experience  
enc
```

Sample Output 1

```
xpri
```

Answer: (penalty regime: 0 %)

```
1 s1=input()  
2 s2=set(input())  
3 r=[i for i in s1 if i not in s2]  
4 print(''.join(r))
```

	Input	Expected	Got	
✓	experience enc	xpri	xpri	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Write a program that takes as input a string (sentence), and returns its second word in uppercase.

For example:

If input is "Wipro Technologies Bangalore" the function should return "TECHNOLOGIES"

If input is "Hello World" the function should return "WORLD"

If input is "Hello" the program should return "LESS"

NOTE 1: If input is a sentence with less than 2 words, the program should return the word "LESS".

NOTE 2: The result should have no leading or trailing spaces.

For example:

Input	Result
Wipro Technologies Bangalore	TECHNOLOGIES
Hello World	WORLD
Hello	LESS

Answer: (penalty regime: 0 %)

```

1 sentence=input()
2 words=sentence.split()
3 if len(words)<2:
4     print("LESS")
5 else:
6     print(words[1].upper())

```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore	TECHNOLOGIES	TECHNOLOGIES	✓
✓	Hello World	WORLD	WORLD	✓
✓	Hello	LESS	LESS	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Write a python program to count all letters, digits, and special symbols respectively from a given string

For example:

Input	Result
rec@123	3 3 1

Answer: (penalty regime: 0 %)

```

1 string=input()
2 letter_count=0
3 digit_count=0
4 special_symbol_count=0
5 for char in string:
6     if char .isalpha():
7         letter_count+=1
8     elif char.isdigit():
9         digit_count+=1
10    else:
11        special_symbol_count+=1
12 print(letter_count)
13 print(digit_count)
14 print(special_symbol_count)

```

	Input	Expected	Got	
✓	rec@123	3 3 1	3 3 1	✓
✓	P@#yn26at^&i5ve	8 3 4	8 3 4	✓
✓	abc@12&	3 2 2	3 2 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

Correct

Mark 1.00 out of 1.00

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Input Format:

The first line contains S.

Output Format:

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

Boundary Condition:

$1 \leq \text{Length of } S \leq 100$

Example Input/Output 1:

Input:

abcd@gmail.com

Output:

com

gmail

abcd

For example:

Input	Result
arvijayakumar@rajalakshmi.edu.in	edu.in rajalakshmi arvijayakumar

Answer: (penalty regime: 0 %)

```

1 n=input()
2 u_d,e=n.split(".",1)
3 u,d=u_d.split("@")
4 print(e)
5 print(d)
6 print(u)

```

	Input	Expected	Got	
✓	abcd@gmail.com	com gmail abcd	com gmail abcd	✓
✓	arvijayakumar@rajalakshmi.edu.in	edu.in rajalakshmi arvijayakumar	edu.in rajalakshmi arvijayakumar	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8

Correct

Mark 1.00 out of 1.00

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

Input Format:

The first line contains S1.

The second line contains S2.

The third line contains N.

Output Format:

The first line contains the N characters present in S1 which are also present in S2.

Boundary Conditions:

$2 \leq N \leq 10$

$2 \leq \text{Length of } S1, S2 \leq 1000$

Example Input/Output 1:

Input:

abcbde
cdefghbb

3

Output:

bcd

Note:

b occurs twice in common but must be printed only once.

Answer: (penalty regime: 0 %)

```

1 i1=input()
2 i2=input()
3 i3=int(input())
4 r=''
5 for i in i1:
6     for j in i2:
7         if i in r:
8             break
9     else:
10        if i==j:
11            r+=i
12 print(r[0:i3])
13

```

	Input	Expected	Got	
✓	abcbde cdefghbb 3	bcd	bcd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9

Correct

Mark 1.00 out of 1.00

String should contain only the words are not palindrome.

Sample Input 1

Malayalam is my mother tongue

Sample Output 1

is my mother tongue

Answer: (penalty regime: 0 %)

```
1 s=input()
2 s=s.lower()
3 r=s.split()
4 r1=[]
5 for i in r:
6     if i!=i[::-1]:
7         r1.append(i)
8 print(' '.join(r1))
9
10
```

	Input	Expected	Got	
✓	Malayalam is my mother tongue	is my mother tongue	is my mother tongue	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10

Correct

Mark 1.00 out of 1.00

Assume that the given string has enough memory.

Don't use any extra space(IN-PLACE)

Sample Input 1

a2b4c6

Sample Output 1

aabbbbcccccc

Answer: (penalty regime: 0 %)

```

1 s=input()
2 r=''
3 i=0
4 while i<len(s):
5     if s[i].isalpha():
6         char=s[i]
7         i+=1
8         c=0
9     while i<len(s) and s[i].isdigit():
10        c=c*10+int(s[i])
11        i+=1
12        r+=char*c
13    else:
14        i+=1
15 print(r)
16
17
18

```

	Input	Expected	Got	
✓	a2b4c6	aabbbbcccccc	aabbbbcccccc	✓
✓	a12b3d4	aaaaaaaaaaabbddddd	aaaaaaaaaaabbddddd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week5_MCQ](#)

Jump to...

List ►

Started on Wednesday, 5 June 2024, 1:49 PM

State Finished

Completed on Thursday, 6 June 2024, 9:52 PM

Time taken 1 day 8 hours

Marks 10.00/10.00

Grade **100.00** out of 100.00

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Example

$n = 20$

$p = 3$

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if $p = 3$, then 4 is returned. If $p > 6$, 0 would be returned.

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^9$

The first line contains an integer n , the number to factor.

The second line contains an integer p , the 1-based index of the factor to return.

Sample Case 0

Sample Input 0

10

3

Sample Output 0

5

Explanation 0

Factoring $n = 10$ results in {1, 2, 5, 10}. Return the $p = 3^{\text{rd}}$ factor, 5, as the answer.

Sample Case 1

Sample Input 1

10

5

Sample Output 1

0

Explanation 1

Factoring $n = 10$ results in {1, 2, 5, 10}. There are only 4 factors and $p = 5$, therefore 0 is returned as the answer.

Sample Case 2

Sample Input 2

1

1

Sample Output 2

1

Explanation 2

Factoring $n = 1$ results in {1}. The $p = 1^{\text{st}}$ factor of 1 is returned as the answer.

For example:

Input	Result
10	5
3	
10	0
5	

Answer: (penalty regime: 0 %)

```
1 import math
2 n=int(input())
3 p=int(input())
4 factors=[]
5 sqrt_n=int(math.sqrt(n))
6 for i in range(1,sqrt_n+1):
7     if n%i==0:
8         factors.append(i)
9     if i!=n//i:
10        factors.append(n//i)
11 factors.sort()
12 result=factors[p-1] if p<=len(factors)else 0
13 print(result)
```

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

Sample Test Cases

Test Case 1

Input

```
1  
3  
4  
5  
6  
7  
8  
9  
10  
11  
2
```

Output

ITEM to be inserted:2

After insertion array is:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11
```

Test Case 2

Input

```
11  
22  
33  
55  
66  
77  
88  
99  
110  
120  
44
```

Output

ITEM to be inserted:44

After insertion array is:

```
11  
22  
33  
44
```

Answer: (penalty regime: 0 %)

```

1 array=[]
2 c=0
3 count=0
4 for i in range(0,10):
5     array.append(int(input()))
6 n=int(input())
7 print("ITEM to be inserted:",end="")
8 print(n)
9 print("After insertion array is:")
10 for i in range(0,10):
11     if n<array[i] and count==0:
12         array.insert(i,n)
13         count=count+1
14         break
15 for i in array:
16     print(i)

```

	Input	Expected	Got	
✓	1 3 4 5 6 7 8 9 10 11 2	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	✓
✓	11 22 33 55 66 77 88 99 110 120 44	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

Sample Test Cases

Test Case 1

Input

```
7
23
45
23
56
45
23
40
```

Output

```
23 occurs 3 times
45 occurs 2 times
56 occurs 1 times
40 occurs 1 times
```

Answer: (penalty regime: 0 %)

```
1 size=int(input())
2 arr=[]
3 for _ in range(size):
4     arr.append(int(input()))
5 frequency={}
6 for element in arr:
7     if element in frequency:
8         frequency[element]+=1
9     else:
10        frequency[element]=1
11 for key,value in frequency.items():
12     print(f"{key} occurs {value} times")
```

	Input	Expected	Got	
✓	7 23 45 23 56 45 23 40	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 23 occurs 1 times	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a program to print all the locations at which a particular element (taken as input) is found in a [list](#) and also print the total number of times it occurs in the [list](#). The location starts from 1.

For example, if there are 4 elements in the array:

```
5
6
5
7
```

If the element to search is 5 then the output will be:

5 is present at location 1
 5 is present at location 3
 5 is present 2 times in the array.

Sample Test Cases

Test Case 1

Input

```
4
5
6
5
7
5
```

Output

5 is present at location 1.
 5 is present at location 3.
 5 is present 2 times in the array.

Test Case 2

Input

```
5
67
80
45
97
100
50
```

Output

50 is not present in the array.

Answer: (penalty regime: 0 %)

```
1 n = int(input())
2 arr=[int(input()) for _ in range (n)]
3 search_element = int(input())
4
5 locations=[ ]
6 occurrences = 0
7 for i in range (n):
8     if arr[i]==search_element:
9         locations.append(i+1)
10        occurrences+=1
11 if occurrences>0:
```

	Input	Expected	Got	
✓	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	✓
✓	5 67 80 45 97 100 50	50 is not present in the array.	50 is not present in the array.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[i] - A[j] = k$, $i \neq j$.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Example

Input

```
1
3
1
3
5
4
```

Output:

```
1
```

Input

```
1
3
1
3
5
99
```

Output

```
0
```

For example:

Input	Result
1 3 1 3 5 4	1
1 3 1 3 5 99	0

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 for p in range(0,a):
3     x=int(int(input()))
4     b=[]
5     for i in range(0,x):
6         y=int(input())
7         if
```

```
12 v if b[i]-b[j]==z or b[j]-b[i]==z and i!=j:  
13 c+=1  
14 if c>0:  
15     print(1)  
16 else:  
17     print(0)  
18
```

Week6_Coding: Attempt review | REC-PS

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Output is a merged array without duplicates.

Input Format

N1 - no of elements in array 1

Array elements for array 1

N2 - no of elements in array 2

Array elements for array2

Output Format

Display the merged array

Sample Input 1

```
5  
1  
2  
3  
6  
9  
4  
2  
4  
5  
10
```

Sample Output 1

```
1 2 3 4 5 6 9 10
```

Answer: (penalty regime: 0 %)

```
1 N1 = int(input())  
2 array1 = []  
3 for _ in range (N1):  
4     array1.append(int(input()))  
5  
6 N2 = int (input())  
7 array2 = []  
8 for _ in range (N2):  
9     array2.append(int(input()))  
10  
11 merged_array = array1 + array2  
12 merged_array = list(set(merged_array))  
13 merged_array.sort()  
14 for i in range (len(merged_array)):  
15     print(merged_array[i], end = " ")
```

✓	7 4 7 8 10 12 30 35 9 1 3 4 5 7 8 11 13 22	1 3 4 5 7 8 10 11 12 13 22 30 35	1 3 4 5 7 8 10 11 12 13 22 30 35	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

Example Input:

```
5
1
2
2
3
4
```

Output:

```
1 2 3 4
```

Example Input:

```
6
1
1
2
2
3
3
```

Output:

```
1 2 3
```

For example:

Input	Result
5 1 2 2 3 4	1 2 3 4
6 1 1 2 2 3 3	1 2 3

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 array=[]
3 for _ in range(a):
4     array.append(int(input()))
5 array1=[]
6 for i in array:
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a Python program to Zip two given lists of lists.

Input:

m : row size

n: column size

list1 and list 2 : Two lists

Output

Zipped List : List which combined both list1 and list2

Sample test case

Sample input

2

2

1

3

5

7

2

4

6

8

Sample Output

[[1, 3, 2, 4], [5, 7, 6, 8]]

Answer: (penalty regime: 0 %)

```
1 m=int(input())
2 n=int(input())
3
4 list1=[]
5 list2=[]
6 for i in range(m):
7     row=[int(input())for i in range(n)]
8     list1.append(row)
9 for i in range(m):
10    row=[int(input())for i in range (n)]
11    list2.append(row)
12 zipped_list=[]
13 for sublist1, sublist2 in zip(list1,list2):
14     zipped_list.append(sublist1+sublist2)
15 print(zipped_list)
```

2
1
2
3
4
5
6
7
8

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a Python program to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true

Input:

n : Number of elements

List1: List of values

Output

Print "True" if list is strictly increasing or decreasing else print "False"

Sample Test Case

Input

7
1
2
3
0
4
5
6

Output

True

Answer: (penalty regime: 0 %)

```

1 n = int(input())
2 lst = []
3 for _ in range (n):
4     lst.append(int(input()))
5 strictly_increasing = all(lst[i] < lst[i+1] for i in range(len(lst) -1))
6
7 if strictly_increasing:
8     print("True")
9 else:
10    for i in range(len(lst)):
11        temp=lst[:i] + lst[i+1:]
12    if all (temp[j] < temp[j + 1] for j in range(len(temp) -1)):
13        print("True")
14        break
15    else:
16        print("True")

```

	Input	Expected	Got	
✓	7 1 2 3 0 4 5 6	True	True ✓	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

arr=[1,2,3,4,6]

- the sum of the first three elements, $1+2+3=6$. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

Constraints

- $3 \leq n \leq 10^5$
- $1 \leq \text{arr}[i] \leq 2 \times 10^4$, where $0 \leq i < n$
- It is guaranteed that a solution always exists.

The first line contains an integer n , the size of the array arr.

Each of the next n lines contains an integer, arr[i], where $0 \leq i < n$.

Sample Case 0

Sample Input 0

```
4  
1  
2  
3  
3
```

Sample Output 0

```
2
```

Explanation 0

- The sum of the first two elements, $1+2=3$. The value of the last element is 3.
- Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.
- The index of the pivot is 2.

Sample Case 1

Sample Input 1

```
3  
1  
2  
1
```

Sample Output 1

```
1
```

Explanation 1

- The first and last elements are equal to 1.
- Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.
- The index of the pivot is 1.

For example:

3	1
1	
2	
1	

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 arr=[int(input()) for _ in range(n)]
3 total_sum=sum(arr)
4 left_sum=0
5 for i in range(n):
6     right_sum=total_sum-left_sum-arr[i]
7     if left_sum==right_sum:
8         print(i)
9         break
10    left_sum+=arr[i]
11

```

	Input	Expected	Got	
✓	4 1 2 3 3	2	2	✓
✓	3 1 2 1	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week6_MCQ](#)

Jump to...

[Tuples ►](#)

Sample Output:

1 5 10

3

Sample Input:

5 5

1 2 3 4 5

1 2 3 4 5

Sample Output:

NO SUCH ELEMENTS

For example:

Input	Result
5 4 1 2 8 6 5 2 6 8 10	1 5 10 3
5 5 1 2 3 4 5 1 2 3 4 5	NO SUCH ELEMENTS

Answer: (penalty regime: 0 %)

```
1 size1,size2=list(map(int,input().split()))
2 arr1=set(list(map(int,input().split())))
3 arr2=set(list(map(int,input().split())))
4 non_repeating=sorted(list((arr1-arr2)|(arr2-arr1)))
5 if non_repeating:
6     for i in non_repeating:
7         print(i,end=" ")
8     print()
9     print(len(non_repeating))
10 else:
11     print("NO SUCH ELEMENTS")
```

Marks for this submission: 1.00 / 1.00.

6/19/24, 8:38 PM

Week7_Coding: Attempt review | REC-PS

For example:

Input	Result
hello world ad	1
Faculty Upskilling in Python Programming ak	2

Answer: (penalty regime: 0 %)

```

1 text=input()
2 text=text.upper()
3 broke=input()
4 broke=broke.upper()
5 arr=[]
6 count=0
7 count1=0
8 arr=text.split(' ')
9 arr1=[]
10 for i in broke:
11     for j in arr:
12         if str(i) in j:
13             if j not in arr1:
14                 arr1.append(j)
15 for i in arr1:
16     count=count+1
17 for i in arr:
18     count1=count1+1
19 print(count1-count)
20
21

```

	Input	Expected	Got	
✓	hello world ad	1	1	✓
✓	Welcome to REC e	1	1	✓
✓	Faculty Upskilling in Python Programming ak	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

For example:

Input	Result
1 3 4 4 2	4

Answer: (penalty regime: 0 %)

```
1 nums=list(map(int,input().split()))
2 seen=set()
3 for num in nums:
4     if num in seen:
5         print(num)
6         break
7 else:
8     seen.add(num)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

For example:

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

Answer: (penalty regime: 0 %)

```

1 t=tuple(map(int,input().split(',')))
2 k=int(input())
3 count=0
4 t=set(t)
5 for num in t:
6     complement=k-num
7     if complement in t and num!=complement:
8         count+=1
9 print(count//2)

```

	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	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Output: ["AAAAACCCCC", "CCCCCAAAAA"]

6/19/24, 8:38 PM

Week7_Coding: Attempt review | REC-PS

Example 2:

Input: s = "AAAAAAAAAAAAAA"

Output: ["AAAAAAAAAA"]

For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Answer: (penalty regime: 0 %)

```

1 s=input()
2 if len(s)<=10:
3     print([])
4 sequences={}
5 for i in range(len(s)-9):
6     substring=s[i:i+10]
7     sequences[substring]=sequences.get(substring,0)+1
8 repeated_sequences=[seq for seq,count in sequences.items() if count>1]
9 for seq in repeated_sequences:
10    print(seq)

```

	Input	Expected	Got	
✓	AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA	✓	
✓	AAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Constraints:

$1 \leq s1.length, s2.length \leq 200$

s1 and s2 consist of lowercase English letters and spaces.

s1 and s2 do not have leading or trailing spaces.

All the words in s1 and s2 are separated by a single space.

Note:

Use [dictionary](#) to solve the problem

For example:

Input	Result
this apple is sweet	sweet sour
this apple is sour	

Answer: (penalty regime: 0 %)

```

1 from collections import Counter
2
3 def uncommon_words(s1, s2):
4     # Tokenize the sentences into words
5     words_s1 = s1.split()
6     words_s2 = s2.split()
7
8     # Count the frequency of each word in both sentences
9     counter_s1 = Counter(words_s1)
10    counter_s2 = Counter(words_s2)
11
12    # Initialize a list to store uncommon words
13    uncommon = []
14
15    # Iterate over the words in both sentences
16    for word, count in counter_s1.items():
17        # Check if the word appears exactly once in s1 and does not appear in s2
18        if count == 1 and counter_s2.get(word, 0) == 0:
19            uncommon.append(word)
20
21    for word, count in counter_s2.items():
22        # Check if the word appears exactly once in s2 and does not appear in s1
23        if count == 1 and counter_s1.get(word, 0) == 0:
24            uncommon.append(word)
25
26    for i in uncommon:
27        print(i, end=" ")
28
29 # Test cases
30 s1=input()
31 s2=input()
32 uncommon_words(s1,s2) # Output: ["sweet", "sour"]
33 # Output: ["banana"]

```


Output : John

6/19/24, 8:39 PM Week8_Coding Attempt review | REC-PS
We have four Candidates with name as 'John', 'Johnny', 'jamie', 'Jackie'. The candidates John and Johny get maximum votes. Since John is alphabetically smaller, we print it. Use [dictionary](#) to solve the above problem

Sample Input:

10
John
John
Johny
Jamie
Jamie
Johny
Jack
Johny
Johny
Jackie

Sample Output:

Johny

Answer: (penalty regime: 0 %)

```
1 def find_winner(votes):
2     # Create a dictionary to store vote counts
3     vote_counts = {}
4
5     # Iterate over the votes
6     for candidate in votes:
7         if candidate in vote_counts:
8             vote_counts[candidate] += 1
9         else:
```

6/19/24, 8:39 PM	Input	Expected	Got	
✓	10 John John Johny Jamie Jamie Johny Jack Johny Johny Jackie	Johny	Johny	✓
✓	6 Ida Ida Ida Kiruba Kiruba Kiruba	Ida	Ida	✓

Week8_Coding: Attempt review | REC-PS

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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	

Answer: (penalty regime: 0 %)

```
1 def compute_student_statistics(student_data):
2     highest_avg_score = 0
3     highest_assignment_mark = 0
4     lowest_lab_mark = float('inf')
5     lowest_avg_score = float('inf')
6
7     students_highest_avg = []
8     students_highest_assignment = []
9     students_lowest_lab = []
10    students_lowest_avg = []
11
12   for student, marks in student_data.items():
13       test_mark, assignment_mark, lab_mark = marks
14       avg_score = (test_mark + assignment_mark + lab_mark) / 3
15
16       # Check for highest average score
17       if avg_score > highest_avg_score:
18           highest_avg_score = avg_score
19           students_highest_avg = [student]
```

```

40     students_lowest_avg = [student]
41     elif avg_score == lowest_avg_score:
42         students_lowest_avg.append(student) Week8_Coding: Attempt review | REC-PS
43
44     return (
45         sorted(students_highest_avg),
46         sorted(students_highest_assignment),
47         sorted(students_lowest_lab),
48         sorted(students_lowest_avg)
49     )
50
51 # Function to parse input and call compute_student_statistics
52 def process_student_data(n, student_info):

```

	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 Lalith	Ram James Ram Lalith 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	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Best 7 6 5

Sample Output

Gfg 17

Best 18

For example:

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

Answer: (penalty regime: 0 %)

```
1 def sort_dict_by_value_sum(test_dict):
2     # Calculate the sum of values for each key and store them in a dictionary
3     sum_dict = {key: sum(values) for key, values in test_dict.items()}
4
5     # Sort the keys based on the sum of their corresponding value lists
6     sorted_keys = sorted(sum_dict, key=sum_dict.get)
7
8     # Create a new dictionary with the sorted keys and their corresponding sums
9     sorted_dict = {key: sum_dict[key] for key in sorted_keys}
10
11    return sorted_dict
12
13 # Sample Input
14 n = int(input())
15 test_dict = {}
16 for _ in range(n):
17     key, *values = input().split()
18     test_dict[key] = list(map(int, values))
19
20 # Sort the dictionary by the summation of values in value list
21 sorted_dict = sort_dict_by_value_sum(test_dict)
22
23 # Display the result
24 for key, value_sum in sorted_dict.items():
25     print(key, value_sum)
26
```


10 Q and Z

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.

[Sample](#) Input

REC

[Sample](#) Output

REC is worth 5 points.

For example:

Input	Result
REC	REC is worth 5 points.

Answer: (penalty regime: 0 %)

```

1 def scrabble_score(word):
2     # Dictionary mapping letters to their points
3     letter_points = {
4         'A': 1, 'E': 1, 'I': 1, 'L': 1, 'N': 1, 'O': 1, 'R': 1, 'S': 1, 'T': 1, 'U': 1,
5         'D': 2, 'G': 2,
6         'B': 3, 'C': 3, 'M': 3, 'P': 3,
7         'F': 4, 'H': 4, 'V': 4, 'W': 4, 'Y': 4,
8         'K': 5,
9         'J': 8, 'X': 8,
10        'Q': 10, 'Z': 10
11    }
12
13    # Calculate the score of the word
14    score = sum(letter_points.get(letter.upper(), 0) for letter in word)
15
16    return score
17
18 # Sample Input
19 word = input()
20
21 # Calculate the Scrabble score for the word
22 score = scrabble_score(word)
23
24 # Display the result
25 print(f"{word} is worth {score} points.")
26

```


[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Functions: Built-in functions, User-defined functions, Recursive functions](#) / [Week9 Coding](#)

Started on Friday, 7 June 2024, 7:02 PM

State Finished

Completed on Friday, 7 June 2024, 8:03 PM

Time taken 1 hour

Marks 5.00/5.00

Grade **100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

An abundant number is a number for which the sum of its proper divisors is greater than the number itself. Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Return Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is $1 + 2 + 3 + 4 + 6 = 16$. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

For example:

Test	Result
print(abundant(12))	Yes
print(abundant(13))	No

Answer: (penalty regime: 0 %)

Reset answer

```

1 def abundant(n):
2     sum=0
3     c=int(n)
4     for i in range(1,n):
5         if n%i==0:
6             sum+=i
7     if sum>n:
8         return "Yes"
9     else:
10        return "No"
11 # Input
12
13
14 # Check if the number is abundant
15
16 # Output
17
18

```

	Test	Expected	Got	
✓	print(abundant(12))	Yes	Yes	✓
✓	print(abundant(13))	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

A number is considered to be ugly if its only prime factors are 2, 3 or 5.

[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...] is the sequence of ugly numbers.

Task:

complete the function which takes a number n as input and checks if it's an ugly number.

return ugly if it is ugly, else return not ugly

Hint:

An ugly number U can be expressed as: $U = 2^a * 3^b * 5^c$, where a, b and c are nonnegative integers.

For example:

Test	Result
print(checkUgly(6))	ugly
print(checkUgly(21))	not ugly

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 v def checkUgly(n):
2 v     if n <= 0:
3 v         return "not ugly"
4 v     for factor in [2, 3, 5]:
5 v         while n % factor == 0:
6 v             n //= factor
7
8     return "ugly" if n == 1 else "not ugly"
```

	Test	Expected	Got	
✓	print(checkUgly(6))	ugly	ugly	✓
✓	print(checkUgly(21))	not ugly	not ugly	✓

Passed all tests! ✓

[Correct](#)

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

complete function to implement coin change making problem i.e. finding the minimum number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 def coinChange(n):
2     coins = [4, 3, 2, 1]
3
4     # Initialize the count of coins
5     coin_count = 0
6
7     # Loop through each coin denomination
8     for coin in coins:
9         # Add the number of coins of this denomination to the count
10        coin_count += n // coin
11        # Reduce the amount by the value of the coins used
12        n %= coin
13
14    return coin_count
15
16

```

	Test	Expected	Got	
✓	print(coinChange(16))	4	4	✓

Passed all tests! ✓

[Correct](#)

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

An e-commerce company plans to give their customers a special discount for Christmas.

They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Constraints

$1 \leq \text{orderValue} < 10e100000$

Input

The input consists of an integer `orderValue`, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

For example:

Test	Result
<code>print(christmasDiscount(578))</code>	12

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 ✓ def christmasDiscount(n):
2     sum=0
3     while n>0:
4         c=n%10
5         if c==2 or c==3 or c==5 or c==7:
6             sum+=c
7         n=n//10
8     return sum

```

	Test	Expected	Got	
✓	<code>print(christmasDiscount(578))</code>	12	12	✓

Passed all tests! ✓

[Correct](#)

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Write a code to check whether product of digits at even places is divisible by sum of digits at odd place of a positive integer.

Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

For example:

Test	Result
print(productDigits(1256))	True
print(productDigits(1595))	False

Answer: (penalty regime: 0 %)

Reset answer

```

1 def productDigits(n):
2     pro=1
3     sum=0
4     count=0
5     while(n>0):
6         c=n%10
7         count=count+1
8         if count%2!=0:
9             pro*=c
10        else:
11            sum+=c
12        n=n//10
13        if pro%sum==0:
14            return "True"
15        else:
16            return "False"
```

	Test	Expected	Got	
✓	print(productDigits(1256))	True	True	✓
✓	print(productDigits(1595))	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week9_MCQ](#)

Jump to...

Searching ►

6/19/24, 8:41 PM

Week10_Coding: Attempt review | REC-PS

Started on Friday, 7 June 2024, 8:04 PM**State** Finished**Completed on** Friday, 7 June 2024, 8:20 PM**Time taken** 16 mins 21 secs**Marks** 5.00/5.00**Grade** **100.00** out of 100.00

Given a list of integers, sort the array in ascending order using the *Bubble Sort* algorithm above. Once sorted, print the following three lines:

1. List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
2. First Element: firstElement, the *first* element in the sorted list.
3. Last Element: lastElement, the *last* element in the sorted list.

For example, given a worst-case but small array to sort: a=[6,4,1]. It took 3 swaps to sort the array. Output would be

Array is sorted in 3 swaps.

First Element: 1

Last Element: 6

Input Format

The first line contains an integer, n, the size of the list a .

The second line contains n, space-separated integers a[i].

Constraints

- 2<=n<=600
- 1<=a[i]<=2x10⁶.

Output Format

You must print the following three lines of output:

1. List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
2. First Element: firstElement, the *first* element in the sorted list.
3. Last Element: lastElement, the *last* element in the sorted list.

Sample Input 0

3
1 2 3

Sample Output 0

List is sorted in 0 swaps.

First Element: 1
Last Element: 3

For example:

Input	Result
3 3 2 1	List is sorted in 3 swaps. First Element: 1 Last Element: 3
5 1 9 2 8 4	List is sorted in 4 swaps. First Element: 1 Last Element: 9

Answer: (penalty regime: 0 %)

```

1 def bubble_sort(arr):
2     n = len(arr)
3     num_swaps = 0
4     for i in range(n):
5         # Track whether any swaps are made in this pass
6         swapped = False
7         for j in range(0, n-i-1):
8             if arr[j] > arr[j+1]:
9                 # Swap elements
10                arr[j], arr[j+1] = arr[j+1], arr[j]
11                num_swaps += 1

```

18
19
20
21
22
23
24
25
26
27
28
29

6/19/24, 8:14 PM Input

Week10_Coding: Attempt review | REC-PS

```
return num_swaps
n = int(input())
a = list(map(int, input().split()))
# Perform bubble sort and count swaps
num_swaps = bubble_sort(a)
# Output the results
print(f"List is sorted in {num_swaps} swaps.")
print(f"First Element: {a[0]}")
print(f"Last Element: {a[-1]}")
```

	Input	Expected	Got	
✓	3 3 2 1	List is sorted in 3 swaps. First Element: 1 Last Element: 3	List is sorted in 3 swaps. First Element: 1 Last Element: 3	✓
✓	5 1 9 2 8 4	List is sorted in 4 swaps. First Element: 1 Last Element: 9	List is sorted in 4 swaps. First Element: 1 Last Element: 9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Bubble Sort is the simplest [sorting](#) algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. You read an [list](#) of numbers. You need to arrange the elements in ascending order and print the result. The [sorting](#) should be done using bubble sort.

Input Format: The first line reads the number of elements in the array. The second line reads the array elements one by one.

Output Format: The output should be a sorted [list](#).

For example:

Input	Result
6 3 4 8 7 1 2	1 2 3 4 7 8
5 4 5 2 3 1	1 2 3 4 5

Answer: (penalty regime: 0 %)

```

1 def bubble_sort(arr):
2     n = len(arr)
3     for i in range(n):
4         # Track whether any swaps are made in this pass
5         swapped = False
6         for j in range(0, n-i-1):
7             if arr[j] > arr[j+1]:
8                 # Swap elements
9                 arr[j], arr[j+1] = arr[j+1], arr[j]
10                swapped = True
11            # If no swaps were made, the array is already sorted
12            if not swapped:
13                break
14        return arr
15
16 # Input
17 n = int(input())
18 a = list(map(int, input().split()))
19
20 # Ensure that the number of elements matches the expected count
21 if len(a) != n:
22     print("Error: The number of elements does not match the provided count.")
23 else:
24     # Perform bubble sort
25     sorted_list = bubble_sort(a)
26
27 # Output the sorted list
28 for i in sorted_list:
29     print(i,end=" ")
30

```

	Input	Expected	Got	
✓	6 3 4 8 7 1 2	1 2 3 4 7 8	1 2 3 4 7 8	✓
✓	6 9 18 1 3 4 6	1 3 4 6 9 18	1 3 4 6 9 18	✓
✓	5 4 5 2 3 1	1 2 3 4 5	1 2 3 4 5	✓

Passed all tests! ✓

To find the frequency of numbers in a [list](#) and display in sorted order.

Constraints:

$1 \leq n, arr[i] \leq 100$

Input:

1 68 79 4 90 68 1 4 5

output:

1 2

4 2

5 1

68 2

79 1

90 1

For example:

Input	Result
4 3 5 3 4 5	3 2
	4 2
	5 2

Answer: (penalty regime: 0 %)

```

1 def count_frequencies(arr):
2     # Create a dictionary to count the frequencies
3     frequency_dict = {}
4
5     for num in arr:
6         if num in frequency_dict:
7             frequency_dict[num] += 1
8         else:
9             frequency_dict[num] = 1
10
11    # Sort the dictionary by keys
12    sorted_keys = sorted(frequency_dict.keys())
13
14    # Print the sorted frequencies
15    for key in sorted_keys:
16        print(f"{key} {frequency_dict[key]}")
17
18 # Input
19 arr = list(map(int, input().split()))
20
21 # Count frequencies and print the result
22 count_frequencies(arr)
23

```

	Input	Expected	Got	
✓	4 3 5 3 4 5	3 2 4 2 5 2	3 2 4 2 5 2	✓
✓	12 4 4 4 2 3 5	2 1 3 1 4 3 5 1 12 1	2 1 3 1 4 3 5 1 12 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Write a Python program to sort a list of elements using the merge sort algorithm.

For example:

Input	Result
5	3 4 5 6 8
6 5 4 3 8	

Answer: (penalty regime: 0 %)

```

1 def merge_sort(arr):
2     if len(arr) > 1:
3         mid = len(arr) // 2 # Finding the mid of the array
4         left_half = arr[:mid] # Dividing the elements into 2 halves
5         right_half = arr[mid:]
6
7         merge_sort(left_half) # Sorting the first half
8         merge_sort(right_half) # Sorting the second half
9
10    i = j = k = 0
11
12    # Copy data to temp arrays L[] and R[]
13    while i < len(left_half) and j < len(right_half):
14        if left_half[i] < right_half[j]:
15            arr[k] = left_half[i]
16            i += 1
17        else:
18            arr[k] = right_half[j]
19            j += 1
20        k += 1
21
22    # Checking if any element was left in left_half
23    while i < len(left_half):
24        arr[k] = left_half[i]
25        i += 1
26        k += 1
27
28    # Checking if any element was left in right_half
29    while j < len(right_half):
30        arr[k] = right_half[j]
31        j += 1
32        k += 1
33
34 # Function to print the list
35 def print_list(arr):
36     for i in arr:
37         print(i, end=" ")
38     print()
39
40 # Input
41 n = int(input())
42 arr = list(map(int, input().split()))
43
44 # Call merge_sort function
45 merge_sort(arr)
46
47 # Print the sorted list
48
49 print_list(arr)
50

```

	Input	Expected	Got	
✓	5 6 5 4 3 8	3 4 5 6 8	3 4 5 6 8	✓

✓	4 8:41 PM 43 23 49	23 43 49 86	23 43 49 86	✓
		Week10_Coding: Attempt review REC-PS		

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

An list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.

Input Format

The first line contains a single integer n , the length of list

The second line contains n space-separated integers, list[i].

The third line contains integer k.

Output Format

Print Yes or No.

Sample Input

```
7
0 1 2 4 6 5 3
1
```

Sample Output

Yes

For example:

Input	Result
5 8 9 12 15 3 11	Yes
6 2 9 21 32 43 43 1 4	No

Answer: (penalty regime: 0 %)

```

1 def two_sum_exists(arr, K):
2     # Create a set to store the numbers we've seen so far
3     seen = set()
4
5     for num in arr:
6         # Check if the complement (K - num) exists in the set
7         if K - num in seen:
8             return "Yes"
9         # Add the current number to the set
10        seen.add(num)
11
12    return "No"
13
14 # Input
15 n = int(input())
16 arr = list(map(int, input().split()))
17 K = int(input())
18
19 # Check if any two numbers sum to K
20 result = two_sum_exists(arr, K)
21
22 # Output the result
23 print(result)
24
```

6/19/24, 8:41 PM

	8 9 12 15 3 11			
✓	6 2 9 21 32 43 43 1 4	No	No	✓
✓	6 13 42 31 4 8 9 17	Yes	Yes	✓

Week10_Coding: Attempt review | REC-PS

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ Week10_MCQ

Jump to...

Sorting ►