RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

CS23221
PYTHON PROGRAMMING LAB

Laboratory Observation Note Book

Name: MANISHA P

Year/Branch/Section: 1-year /AI&DS-A

Register No: 2116231801096

Semester: 2nd Semester

Academic Year: 2023 - 2024

INDEX

Reg. No. :2116231801096 **Name :** MANISHA P

Year : 1 year **Branch** :AI&DS **Sec :** B

PYTHON PROGRAMMING LAB

CS23221

WEEK 1

Experiments based on Variables,Datatypes in Python

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

Answer:(penalty regime: 0 %)

```
a=input()
b=input()
c=int(a)
d=float(b)
print(c,type(c),sep=',')
print(format(d,'0.1f'),type(d),sep=',')
```

	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	55000,<class 'int'>	55000,<class 'int'>	

	Input	Expected	Got	
	56.2	56.2,<class 'float'>	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

Flag question

Question text

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 %)

```
a=int(input())
b=(40/100)*a
c=(20/100)*a
d=a+b+c
print(int(d))
```

	Input	Expected	Got	
	10000	16000	16000	

	Input	Expected	Got	
	20000	32000	32000	
	28000	44800	44800	
	5000	8000	8000	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
import math
a=float(input())
print("%.3f"%math.sqrt(a))
```

	Input	Expected	Got	
	8.00	2.828	2.828	

	Input	Expected	Got	
	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

Flag question

Question text

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 500 60000	30.43 is the gain percent.

Answer:(penalty regime: 0 %)

```
x=int(input())
y=int(input())
z=int(input())
gain=(z-x-y)/(x+y)*100
print(format(gain,'0.2f'),"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.	
	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

Flag question

Question text

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 20	Your total refund will be $7.00.

Answer:(penalty regime: 0 %)

```
a=float(input())
b=float(input())
c=0.10*a+0.25*b
print("Your total refund will be $",end='')
print(format(c,".2f"),end='.')
```

	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

Flag question

Question text

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 %)

```
a=int(input())
b=abs((a-500)/130)
c=b+10
print("weekdays",format(c,"0.2f"))
print("weekend",format(b,"0.2f"))
```

	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	
	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.

WEEK 2

Operators and Formatting Output

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Answer:(penalty regime: 0 %)

```
a=int(input())
a=abs(a)
print(a%10)
```

	Input	Expected	Got	
	197	7	7	
	-197	7	7	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 10.00 out of 10.00

Flag question

Question text

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 %)

```
a=int(input())
b=int(input())
print("The total weight of all these widgets and gizmos
is",a*75+b*112,"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.

Question **3**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

Mr.Ram has been given a problem kindly help him to solve it. The input of the program is either 0 or 1. IF 0 is the input he should display "C" if 1 is the input it should display "D".There is a constraint that Mr. Ram should use either logical operators or arithmetic operators to solve the problem, not anything else.

Hint:

Use ASCII values of C and D.

Input Format:

An integer x, 0<=x<=1. .

Output Format:

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

Input 1:

```
0
```

Output 1:

```
C
```

Input 2:

```
1
```

```
Output 1:
D
```

For example:

Input	Result
0	C

Answer:(penalty regime: 0 %)

```
print(chr(67+int(input())))
```

	Input	Expected	Got	
	0	C	C	
	1	D	D	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Note:

Dont use if-else. Operators alone must be used .

A team from the Rotract club had planned to conduct a rally to create awareness among the Coimbatore people to donate blood. They conducted the rally successfully. Many of the Coimbatore people realized it and came forward to donate their blood to nearby blood banks. The eligibility criteria for donating blood are people should be above or equal to 18 and his/her weight should be above 40. There was a huge crowd and staff in the blood bank found it difficult to manage the crowd. So they decided to keep a system and ask the people to enter their age and weight in the system. If a person is eligible he/she will be allowed inside.

Write a program and feed it to the system to find whether a person is eligible or not.

Input Format:

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 40	False

Answer:(penalty regime: 0 %)

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

	Input	Expected	Got	
	19 45	True	True	
	18 40	False	False	
	18 42	True	True	
	16 45	False	False	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

In London, every year during Dasara there will be a very grand doll show. People try to invent new dolls of different varieties. The best-sold doll's creator will be awarded with a cash prize. So people broke their heads to create dolls innovatively. Knowing this competition, Mr.Lokpaul tried to create a doll that sings only when an even number is pressed and the number should not be zero and greater than 100.

IF Lokpaul wins print true, otherwise false.

Sample Input

10

Sample Output

True

Explanation:

Since 10 is an even number and a number between 0 and 100, True is printed

For example:

Input	Result
101	False

Answer:(penalty regime: 0 %)

```
a=int(input())
print(a%2==0 and (a<100 and a>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.

Question **6**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

In the 1800s, the battle of Troy was led by Hercules. He was a superstitious person. He believed that his crew can win the battle only if the total count of the weapons in hand is in multiple of 3 and the soldiers are in an even number of count. Given the total number of

weapons and the soldier's count, Find whether the battle can be won or not according to Hercules's belief. If the battle can be won print True otherwise print False.

Input format:

Line 1 has the total number of weapons

Line 2 has the total number of Soldiers.

Output Format:

If the battle can be won print True otherwise print False.

Sample Input:

32

43

Sample Output:'

False

For example:

Input	Result
32 43	False

Answer:(penalty regime: 0 %)

```
w=int(input())
s=int(input())
print(w%3==0 and s%2==0)
```

	Input	Expected	Got	
	32 43	False	False	
	273 7890	True	True	
	800 4590	False	False	
	6789 32996	True	True	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct
Mark 1.00 out of 1.00

Flag question

Question text

Mr. X's birthday is in next month. This time he is planning to invite N of his friends. He wants to distribute some chocolates to all of his friends after the party. He went to a shop to buy a packet of chocolates. At the chocolate shop, 4 packets are there with different numbers of chocolates. He wants to buy such a packet which contains a number of chocolates, which can be distributed equally among all of his friends. Help Mr. X to buy such a packet.

Input Given:

N-No of friends

P1,P2,P3 AND P4-No of chocolates

OUTPUT:

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

SAMPLE INPUT AND OUTPUT:

5

25

12

10

9

OUTPUT

True False True False

For example:

Input	Result
5 25 23 20 10	True False True True

Answer:(penalty regime: 0 %)

```
N=int(input())
for i in range(4):
    p=int(input())
    print(p%N==0,end=' ')
```

	Input	Expected	Got	
	5 25 23 20 10	True False True True	True False True True	
	4 23 24 21 12	False True False True	False True False True	
	8 64 8 16 32	True True True True	True True True True	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **8**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Write a python program that takes a integer between 0 and 15 as input and displays the number of '1' s in its binary form.(Hint:use python bitwise operator.

Sample Input

3

Sample Output:

2

Explanation:

The binary representation of 3 is 011, hence there are 2 ones in it. so the output is 2.

For example:

Input	Result
3	2

Answer:(penalty regime: 0 %)

```
a=int(input())
print(str(bin(a).count("1")))
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **9**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Pretend that you have just opened a new savings account that earns 4 percent interest per year. The interest that you earn is paid at the end of the year, and is added to the balance of the savings account. Write a program that begins by reading the amount of money deposited into the account from the user. Then your program should compute and display the amount in the savings account after 1, 2, and 3 years. Display each amount so that it is rounded to 2 decimal places. Sample Input: 10000 Sample Output: 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.
For example:

Input	Result
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.

Answer:(penalty regime: 0 %)

```
a=int(input())
a=a+(a*(4/100))
print('Balance as of end of Year 1: $'+str('{:.2f}'.format(a))+'.')
a=a+(a*(4/100))
print('Balance as of end of Year 2: $'+str('{:.2f}'.format(a))+'.')
a=a+(a*(4/100))
print('Balance as of end of Year 3: $'+str('{:.2f}'.format(a))+'.')
```

	Input	Expected	Got	
	10000	Balance as of end of Year 1: $10400.00.	Balance as of end of Year 1: $10400.00.	

	Input	Expected	Got	
		Balance as of end of Year 2: $10816.00. Balance as of end of Year 3: $11248.64.	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.

Question **10**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

The program that you create for this exercise will begin by reading the cost of a meal ordered at a restaurant from the user. Then your program will compute the tax and tip for the meal. Use your local tax rate (5 percent) when computing the amount of tax owing. Compute the tip as 18 percent of the meal amount (without the tax). The output from your program should include the tax amount, the tip amount, and the grand total for the meal including both the tax and the tip. Format the output so that all of the values are displayed using two decimal places.

Sample Input

100

Sample Output

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

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 %)

```
a=int(input())
print('The tax is','{:.2f}'.format(a*(5/100)),end='')
print(' and the tip is',"{:.2f}".format(a*(18/100)),end='')
print(', making the total',"{:.2f}".format(a+(a*(5/100))+(a*(18/100))))
```

	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.

WEEK 3
Algorithmic Approach:Selection control structures

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 %)

```
a=str(input())
if(a=="February"):
    print(f"{a} has 28 or 29 days in it.")
elif(a in["January","March","May","July","August","October","December"]):
    print(f"{a} has 31 days in it.")
else:
    print(f"{a} has 30 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.	

	Input	Expected	Got	
	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.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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

```
year=int(input())
if(year%12==0):
    print(year,"is the year of the Monkey.")
elif(year%12==1):
    print(year,"is the year of the Rooster.")
elif(year%12==2):
    print(year,"is the year of the Dog.")
elif(year%12==3):
    print(year,"is the year of the Pig.")
elif(year%12==4):
    print(year,"is the year of the Rat.")
elif(year%12==5):
    print(year,"is the year of the Ox.")
elif(year%12==6):
    print(year,"is the year of the Tiger.")
elif(year%12==7):
    print(year,"is the year of the Hare.")
elif(year%12==8):
    print(year,"is the year of the Dragon.")
elif(year%12==9):
    print(year,"is the year of the Snake.")
elif(year%12==10):
    print(year,"is the year of the Horse.")
else:
    print(year,"is the year of the Sheep.")
```

	Input	Expected	Got	
	2010	2010 is the year of the Tiger.	2010 is the year of the Tiger.	
	2020	2020 is the year of the Rat.	2020 is the year of the Rat.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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*3 + 4*4 = 25 = 5*5

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 %)

```
a=int(input())
b=int(input())
c=int(input())
if((a*a+b*b==c*c) or (b*b+c*c==a*a) or(c*c+a*a==b*b)):
    print("yes")
else:
    print("no")
```

	Input	Expected	Got	
	3 5 4	yes	yes	

	Input	Expected	Got	
	5 8 2	no	no	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
A=input()
if(A in["A","E","I","O","U","a","e","i","o","u"]):
    print("It's a vowel.")
elif(A in["Y","y"]):
    print("Sometimes it's a vowel... Sometimes it's a consonant.")
else:
    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.	
	e	It's a vowel.	It's a vowel.	
	r	It's a consonant.	It's a consonant.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
a=int(input())
b=int(input())
c=int(input())
if(a==b and a==c and b==c):
    print("That's a equilateral triangle")
elif(a!=b and a!=c and b!=c):
    print("That's a scalene triangle")
else:
    print("That's a isosceles 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	

	Input	Expected	Got	
	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.

Question **6**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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

Marks in Maths >= 65

Marks in Physics >= 55

Marks in Chemistry >= 50

Or

Total in all three subjects >= 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 60 80	The candidate is eligible

Answer:(penalty regime: 0 %)

```
a=float(input())
b=float(input())
c=float(input())
d=a+b+c
if(a>=65 and b>=55 and c>=50):
    if(d>=180):
        print("The candidate is eligible")
    else:
        print("The candidate is not eligible")
elif(d>=180):
    print("The candidate is eligible")
else:
    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	

	Input	Expected	Got	
	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.

Question **7**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 3	OUT

Answer:(penalty regime: 0 %)

```
a=int(input())
b=int(input())
C=a/2
if(b>=C):
    print("IN")
else:
    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.

Question **8**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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

Answer:(penalty regime: 0 %)

```
a=float(input())
if a<=199:
   d=a*1.20
elif a<400:
    d=a*1.50
elif a<600:
    d=a*1.80
```

```
else:
    d=a*2.00
if d>400:
    d+=d*0.15
d=max(d,100)
print(f"{d:.2f}")
```

	Input	Expected	Got	
	50	100.00	100.00	
	100.00	120.00	120.00	
	500	1035.00	1035.00	
	700	1610.00	1610.00	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **9**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Write a program that returns the second last digit of the given number. Second last digit is being referred 10the 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 %)

```
num=int(input())
```

```
num=abs(num)
if(num<10):
    print("-1")
else:
    a=int(num/10)%10
    print(f"{a}")
```

	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.

Question **10**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
year=int(input())
if((year%400==0) or (year%4==0) and (year%100!=0)):
    print(f"{year} is a leap year.")
else:
    print(f"{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.	
	2400	2400 is a leap year.	2400 is a leap year.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 5

Experiments based on Strings and its Operations

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 %)

```
s = input()
b = list(s)
i = 0
while i < len(b):
    if b[i] == '&':
        if i > 0 and i < len(b) - 1:
            b[i+1], b[i-1] = b[i-1], b[i+1]
    i += 1
print("".join(b))
```

	Input	Expected	Got	
	A&B	B&A	B&A	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Given a string S, which contains several words, print the count C of the words whose length is atleast L. (You can include punctuation marks like comma, full stop also as part of the word length. Space alone must be ignored)

Input Format:

The first line contains S.
The second line contains L.

Output Format:

The first line contains C

Boundary Conditions:

2 <= Length of S <= 1000

Example Input/Output 1:

Input:

During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.
5

Output:

13

Explanation:

The words of minimum length 5 are
During
after
Kenyattas
inauguration
police
elsewhere
capital,
Nairobi,
tried
opposition
holding
peaceful
demonstrations.

Answer:(penalty regime: 0 %)

```
def count_words_with_length_atleast(input_string, min_length):
    words = input_string.split()
    count = sum(1 for word in words if len(word) >= min_length)
    return count
input_string = input().strip()
min_length = int(input().strip())
print(count_words_with_length_atleast(input_string, min_length))
```

	Input	Expected	Got	
	During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations. 5	13	13	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Robert is having 2 strings consist of uppercase & lowercase english letters. Now he want to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter.

Input

The first line contains **T**. Then **T** test cases follow.

Each test case contains a two lines contains a string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

Output

If the first string is less than the second one, print "-1".
If the second string is less than the first one, print "1".
If the strings are equal, print "0".
Note that the letters' case is not taken into consideration when the strings are compared.

Constraints

1≤T≤50
String length≤100

For example:

Input	Result
3 aaaa aaaA abs Abz abcdefg AbCdEfF	0 -1 1

Answer:(penalty regime: 0 %)

```
def compare_strings(s1, s2):
    s1_lower = s1.lower()
    s2_lower = s2.lower()
    if s1_lower < s2_lower:
        return -1
    elif s1_lower > s2_lower:
        return 1
    else:
        return 0
T = int(input().strip())
for _ in range(T):
    string1 = input().strip()
    string2 = input().strip()
    print(compare_strings(string1, string2))
```

	Input	Expected	Got	
	3 aaaa aaaA abs Abz abcdefg AbCdEfF	0 -1 1	0 -1 1	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def are_strings_balanced(s1, s2):
    set_s1 = set(s1)
    set_s2 = set(s2)
    return set_s1.issubset(set_s2)
s1 = input().strip()
s2 = input().strip()
print(are_strings_balanced(s1, s2))
```

	Input	Expected	Got	
	Yn PYnative	True	True	
	Ynf PYnative	False	False	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def count_letters_digits_special_symbols(s):
    letter_count = 0
    digit_count = 0
    special_symbol_count = 0
    for char in s:
        if char.isalpha():
            letter_count += 1
        elif char.isdigit():
            digit_count += 1
        else:
            special_symbol_count += 1

    return letter_count, digit_count, special_symbol_count
input_string = input().strip()
letter_count, digit_count, special_symbol_count =
count_letters_digits_special_symbols(input_string)
print(letter_count)
print(digit_count)
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 **6**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def display_unique_words():
    unique_words=set()
    input_words=[]
    while True:
        word=input().strip()
        if not word:
            break
        input_words.append(word)
        unique_words.add(word)
    for word in input_words:
        if word in unique_words:
            print(word)
            unique_words.remove(word)
display_unique_words()
```

	Input	Expected	Got	
	first second first third second	first second third	first second third	

	Input	Expected	Got	
	rec cse it rec cse	rec cse it	rec cse it	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **7**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def get_second_word_in_uppercase(sentence):
    words = sentence.split()
    if len(words) >= 2:
        return words[1].upper()
    else:
```

```
        return "LESS"
sentence = input().strip()
print(get_second_word_in_uppercase(sentence))
```

	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 **8**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 <= N <= 10
2 <= Length of S1, S2 <= 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 %)

```
def common_characters_in_first_N(S1, S2, N):
    common_chars = ""
    chars_added = 0
    seen = set()

    for char in S1:
        if char in S2 and char not in seen:
            common_chars += char
            seen.add(char)
            chars_added += 1
            if chars_added == N:
                break

    return common_chars
S1 = input().strip()
S2 = input().strip()
N = int(input().strip())
print(common_characters_in_first_N(S1, S2, N))
```

	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

Flag question

Question text

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 <= Length of S <= 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 %)

```
S = input()
at = S.index('@')
dot = S.index('.')
```

```
l = len(S)
i = 0
i1 = dot + 1
m = ""
i2 = at + 1
m1 = ""
m3 = ""
while i1 < l:
    m += S[i1]
    i1 += 1
while i2 < dot:
    m1 += S[i2]
    i2 += 1
while i < at:
    m3 += S[i]
    i += 1
print(m)
print(m1)
print(m3)
```

	Input	Expected	Got	
	abcd@gmail.com	com gmail abcd	com gmail abcd	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

Consider the below words as key words and check the given input is key word or not.

keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

break is a keyword

Example Input:

IF

Output:

IF is not a keyword

For example:

Input	Result
break	break is a keyword
IF	IF is not a keyword

Answer:(penalty regime: 0 %)

```
keywords = {"break", "case", "continue", "default", "defer", "else", "for",
"func", "goto", "if", "map", "range", "return", "struct", "type", "var"}
word = input().strip()
if word in keywords:
    print(word, "is a keyword")
else:
    print(word, "is not a keyword")
```

	Input	Expected	Got	
	break	break is a keyword	break is a keyword	
	IF	IF is not a keyword	IF is not a keyword	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 6

Experiments based on lists and its Operations

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 %)

```
m=int(input())
n=int(input())
list1=[]
for i in range(m):
    row=[]
    for j in range(n):
        element=int(input())
        row.append(element)
    list1.append(row)
list2=[]
for i in range(m):
    row=[]
    for j in range(n):
        element=int(input())
```

```
        row.append(element)
    list2.append(row)
zipped_list=[]
for i in range (m):
    zipped_row=list1[i]+list2[i]
    zipped_list.append(zipped_row)
print(zipped_list)
```

	Input	Expected	Got	
	2 2 1 2 3 4 5 6 7 8	[[1, 2, 5, 6], [3, 4, 7, 8]]	[[1, 2, 5, 6], [3, 4, 7, 8]]	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
# Input array length
n = int(input())
# Input array elements
array = [int(input()) for _ in range(n)]
# Element to search
search_element = int(input())
# Initialize variables to store locations and count
```

```
locations = []
count = 0
# Search for the element and store its locations
for i in range(n):
    if array[i] == search_element:
        locations.append(i + 1)
        count += 1
# Print locations and count
if count > 0:
    for loc in locations:
        print(f"{search_element} is present at location {loc}.")
    print(f"{search_element} is present {count} times in the array.")
else:
    print(f"{search_element} is not present in the array.")
```

	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.

Question **3**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
# Input array length
n = int(input())
# Input array elements
array = [int(input()) for _ in range(n)]
# Create a dictionary to store frequency of each element
frequency = {}
for num in array:
    if num in frequency:
        frequency[num] += 1
    else:
        frequency[num] = 1
# Print frequency of each element
for num, freq in frequency.items():
    print(f"{num} occurs {freq} times")
```

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

Question **4**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
# Input array length
n = int(input())
# Input array elements
array = [int(input()) for _ in range(n)]
# Create a set to store unique elements
distinct_elements = set(array)
# Print distinct elements separated by space
print(*distinct_elements)
```

	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.

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def merge_arrays(arr1, arr2):
    # If both arrays are empty, return an empty array
    if not arr1 and not arr2:
        return []

    # Create a set to store unique elements
    merged_set = set(arr1 + arr2)
    # Convert the set back to a sorted list to maintain order
    merged_array = sorted(list(merged_set))
    return merged_array
# Sample input
n1 = int(input())
array1 = [int(input()) for _ in range(n1)]
```

```
n2 = int(input())
array2 = [int(input()) for _ in range(n2)]
# Merge the arrays
merged = merge_arrays(array1, array2)
# Display the merged array
print(*merged)
```

Input	Expected	Got
5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10	1 2 3 4 5 6 9 10
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.

Question **6**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 != 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	1

Input	Result
4	
1 3 1 3 5 99	0

Answer:(penalty regime: 0 %)

```
T=int(input())
for _ in range(T):
    N = int(input())
    A = [int(input()) for _ in range(N)]
    k = int(input())
    pair_exists = False
    for i in range(N):
        for j in range(i + 1 , N):
            if A[i] - A[j] == k or A[j] - A[i] == k:
                pair_exists = True
                break
        if pair_exists:
           break
    if pair_exists:
        print(1)
    else:
        print(0)
```

	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.

Question **7**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
n=int(input(""))
list1=[int(input()) for _ in range(n)]
def is_strictly_increasing(lst):
    count=0
    for i in range(1, len(lst)):
        if lst[i] < lst[i-1]:
            count +=1
            if count > 1:
                return False
            if i==1 or lst[i] > lst[i-2]:
                 continue
            elif i<len(lst)-1 and lst[i+1]>lst[i-1]:
```

```
                continue
            else:
                return False
    return True
def is_strictly_decreasing(lst):
    reversed_lst=lst[::-1]
    return is_strictly_increasing(reversed_lst)
if is_strictly_increasing(list1) or is_strictly_decreasing(list1):
    print("True")
else:
     print("False")
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **8**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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

- $3 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$, 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:

Input	Result
4 1 2 3 3	2
3 1 2 1	1

Answer:(penalty regime: 0 %)

```
def find_pivot_index(arr):
    total_sum = sum(arr)
    left_sum = 0

    for i in range(len(arr)):
        total_sum -= arr[i]
        if left_sum == total_sum:
            return i
        left_sum += arr[i]

    return -1  # If no pivot is found
# Input processing
n = int(input())
arr = [int(input()) for _ in range(n)]
print(find_pivot_index(arr))
```

	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.

Question **9**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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 $p = 3^{rd}$ 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 = 1st factor of 1 is returned as the answer.

For example:

Input	Result
10 3	5
10 5	0
1 1	1

Answer:(penalty regime: 0 %)

```
import math
def factor(n, p):
    factors = []
    sqrt_n = int(math.sqrt(n))

    for i in range(1, sqrt_n + 1):
        if n % i == 0:
            factors.append(i)
            if i != n // i:
                factors.append(n // i)

    factors.sort()

    if p > len(factors):
        return 0
    else:
        return factors[p - 1]
# Input processing
n = int(input())
p = int(input())
```

```
print(factor(n, p))
```

	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.

Question **10**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

7

1

2

3

3

4

5

6

2

1

6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57
1 7 1 2 3 3 4 5 6 2 1 6	1 6

Answer:(penalty regime: 0 %)

```
def intersection(l1,l2):
    l3=[value for value in l1 if value in l2]
    return l3
n=int(input())
for i in range(0,n):
    s1=int(input())
    l1=[]
    for x in range (0,s1):
        e1=int(input())
        l1.append(e1)
    s2=int(input())
    l2=[]
    for y in range (0,s2):
        e2=int(input())
        l2.append(e2)
    print(*intersection (l1,l2))
```

	Input	Expected	Got	
	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	
	1 7 1 2 3 3 4 5 6 2 1 6	1 6	1 6	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 7

Experiments based on Tuples,Sets and its Operations

There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output:

1

Explanation: We cannot type "world" because the 'd' key is broken.

For example:

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

Answer:(penalty regime: 0 %)

```
s=input().split()
b=input()
b1=b.upper()
broken=b+b1
a=[]
for i in s:
    for j in i:
        if j in broken:
            a.append(i)
            break
```

```
print(len(s)-len(a))
```

	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.

Question **2**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

Write a program to eliminate the common elements in the given 2 arrays and print only the non-repeating

elements and the total number of such non-repeating elements.

Input Format:

The first line contains space-separated values, denoting the size of the two arrays in integer format respectively.

The next two lines contain the space-separated integer arrays to be compared.

Sample Input:

5 4

1 2 8 6 5

2 6 8 10

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

Answer:(penalty regime: 0 %)

```
a,b = map(int, input().split())
l1 = list(map(int, input().split()))
l2 = list(map(int, input().split()))
x = [i for i in l1 if i not in l2]
y = [i for i in l2 if i not in l1]
z = x+y
if z:
    print(' '.join(map(str, z)))
    print(len(z))
```

	Input	Expected	Got	
	5 4 1 2 8 6 5 2 6 8 10	1 5 10 3	1 5 10 3	
	3 3 10 10 10 10 11 12	11 12 2	11 12 2	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

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

Examples:

Input: t = (5, 6, 5, 7, 7, 8), K = 13
Output: 2
Explanation:
Pairs with sum K(= 13) are {(5, 8), (6, 7), (6, 7)}.
Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }.
Therefore, the required output is 2.

For example:

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

Answer:(penalty regime: 0 %)

```
t=tuple(input().split(','))

t=tuple(int(i) for i in t)

k=int(input())

result=[]

for i in t:

    for j in t:

        if(i+j==k and not ([i,j] in result or [j,i] in result)):

            result.append([i,j])

print(len(result))
```

	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.

Question **4**
Correct

Mark 1.00 out of 1.00

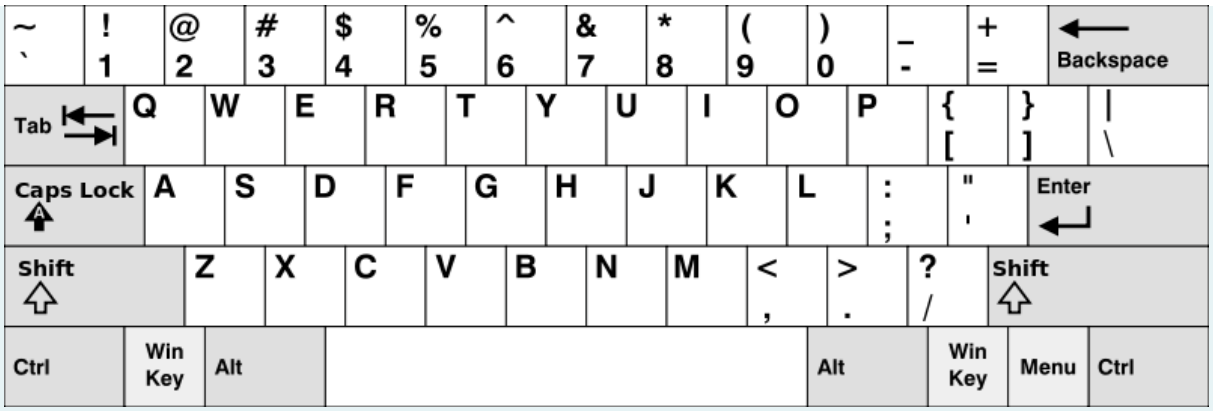
Flag question

Question text

Given an array of strings `words`, return *the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below*.

In the **American keyboard**:

- the first row consists of the characters `"qwertyuiop"`,
- the second row consists of the characters `"asdfghjkl"`, and
- the third row consists of the characters `"zxcvbnm"`.



Example 1:

```
Input: words = ["Hello","Alaska","Dad","Peace"]
Output: ["Alaska","Dad"]
```

Example 2:

```
Input: words = ["omk"]
Output: []
```

Example 3:

```
Input: words = ["adsdf","sfd"]
Output: ["adsdf","sfd"]
```

For example:

Input	Result
4 Hello Alaska Dad Peace	Alaska Dad
2 adsfd afd	adsfd afd

Answer:(penalty regime: 0 %)

```
def check(name,row):
    for i in name:
        if(not (i in row)):
            return False
    return True
name=[input() for i in range(int(input()))]
r1='qwertyuiopQWERTYUIOP'
r2='asdfghjklASDFGHJKL'
r3='zxcvbnmZXCVBNM'
result=[]
for i in name:
    if(i[0]in r1):
        x=r1
    elif(i[0]in r2):
        x=r2
```

```
    else:
        x=r3
    if(check(i,x)):
        result.append(i)
if result:
    for i in result:
        print(i)
else:
    print('No words')
```

	Input	Expected	Got	
	4 Hello Alaska Dad Peace	Alaska Dad	Alaska Dad	
	1 omk	No words	No words	
	2 adsfd afd	adsfd afd	adsfd afd	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**
Correct

Mark 1.00 out of 1.00

Flag question

Question text
The **DNA sequence** is composed of a series of nucleotides abbreviated as `'A'`, `'C'`, `'G'`, and `'T'`.

- For example, `"ACGAATTCCG"` is a **DNA sequence**.

When studying **DNA**, it is useful to identify repeated sequences within the DNA.

Given a string `s` that represents a **DNA sequence**, return all the `10`**-letter-long** sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in **any order**.

Example 1:

```
Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]
```

Example 2:

```
Input: s = "AAAAAAAAAAAAA"
Output: ["AAAAAAAAAA"]
```

For example:

Input	Result
AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Answer:(penalty regime: 0 %)

```
def findRepeatedSequences(s):
    sequence_count = {}
    result = set()
    for i in range (len(s) - 9):
        sequence = s[i:i+10]
        if sequence in sequence_count:
            result.add(sequence)
        else:
             sequence_count[sequence] = 1
    return list(result)
s = input()
repeated_sequences = findRepeatedSequences(s)
for sequence in repeated_sequences:
    print(sequence)
```

	Input	Expected	Got	
	AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCCCCCCCAAAAA	AAAAACCCCCCCCCCAAAAA	
	AAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 8

Experiments based on Dictionary and its Operations

A sentence is a string of single-space separated words where each word consists only of lowercase letters.A word is uncommon if it appears exactly once in one of the sentences, and does not appear in the other sentence.

Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order.

Example 1:

Input: s1 = "this apple is sweet", s2 = "this apple is sour"

Output: ["sweet","sour"]

Example 2:

Input: s1 = "apple apple", s2 = "banana"

Output: ["banana"]

Constraints:

1 <= s1.length, s2.length <= 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 this apple is sour	sweet sour

Answer:(penalty regime: 0 %)

```
def uncommon_words(s1, s2):
    words1 = s1.split()
    words2 = s2.split()
    word_count = {}

    for word in words1:
        word_count[word] = word_count.get(word, 0) + 1
```

```
    for word in words2:
        word_count[word] = word_count.get(word, 0) + 1
    uncommon = [word for word, count in word_count.items() if count == 1]
    return uncommon
def main():
    s1 = input().strip()
    s2 = input().strip()

    result = uncommon_words(s1, s2)
    print(" ".join(result))
if __name__ == "__main__":
    main()
```

	Input	Expected	Got	
	this apple is sweet this apple is sour	sweet sour	sweet sour	
	apple apple banana	banana	banana	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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

1.Identify the student with the highest average score

2.Identify the student who as the highest Assignment marks

3.Identify the student with the Lowest lab marks

4.Identify the student with the lowest average score

Note:

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

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

For example:

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

Answer:(penalty regime: 0 %)

```
def average(scores):
    return sum(scores) / len(scores)
n = int(input())
students = {}
for _ in range(n):
    name, test, assignment, lab = input().split()
    students[name] = [int(test), int(assignment), int(lab)]
averages = {name: average(scores) for name, scores in students.items()}
highest_avg = max(averages.values())
highest_avg_students = [name for name, avg in averages.items() if avg ==
highest_avg]
highest_assignment = max(student[1] for student in students.values())
highest_assignment_students = [name for name, scores in students.items() if
scores[1] == highest_assignment]
lowest_lab = min(student[2] for student in students.values())
lowest_lab_students = [name for name, scores in students.items() if scores[2]
== lowest_lab]
lowest_avg = min(averages.values())
```

```
lowest_avg_students = [name for name, avg in averages.items() if avg ==
lowest_avg]
print("\n".join([
    " ".join(sorted(highest_avg_students)),
    " ".join(sorted(highest_assignment_students)),
    " ".join(sorted(lowest_lab_students)),
    " ".join(sorted(lowest_avg_students))
]))
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

Given an array of names of candidates in an election. A candidate name in the array represents a vote cast to the candidate. Print the name of candidates received Max vote. If there is tie, print a lexicographically smaller name.

Examples:

Input : votes[] = {"john", "johnny", "jackie",

"johnny", "john", "jackie",

"jamie", "jamie", "john",

"johnny", "jamie", "johnny",

"john"};

Output : John

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 %)

```
def find_winner(votes):
    vote_count = {}

    for candidate in votes:
        vote_count[candidate] = vote_count.get(candidate, 0) + 1

    max_votes = max(vote_count.values())
    winners = [candidate for candidate, votes in vote_count.items() if votes
== max_votes]

    return min(winners)
```

```
def main():
    n = int(input())
    votes = [input().strip() for _ in range(n)]
    winner = find_winner(votes)
    print(winner)
main()
```

	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	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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

Points Letters

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

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

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 %)

```
def calculate_scrabble_score(word):
    letter_points = {
        'A': 1, 'E': 1, 'I': 1, 'L': 1, 'N': 1, 'O': 1, 'R': 1, 'S': 1, 'T':
1, 'U': 1,
        'D': 2, 'G': 2,
        'B': 3, 'C': 3, 'M': 3, 'P': 3,
        'F': 4, 'H': 4, 'V': 4, 'W': 4, 'Y': 4,
        'K': 5,
        'J': 8, 'X': 8,
        'Q': 10, 'Z': 10
    }
    score = sum(letter_points.get(letter.upper(), 0) for letter in word)
    return score
def main():
    word = input().strip()
    score = calculate_scrabble_score(word)
    print(f"{word} is worth {score} points.")
main()
```

	Input	Expected	Got	
	GOD	GOD is worth 5 points.	GOD is worth 5 points.	
	REC	REC is worth 5 points.	REC is worth 5 points.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

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

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

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

Explanation : Sorted by sum, and replaced.

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

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

Explanation : Sorted by sum, and replaced.

Sample Input:

2

Gfg 6 7 4

Best 7 6 5

Sample Output

Gfg 17

Best 18

For example:

Input	Result
2 Gfg 6 7 4	Gfg 17 Best 18

Input	Result
Best 7 6 5	

Answer:(penalty regime: 0 %)

```
num_entries = int(input())
test_dict = {}
for _ in range(num_entries):
    key, *values = input().split()
    test_dict[key] = sum(map(int, values))
sorted_dict = dict(sorted(test_dict.items(), key=lambda item: item[1]))
for key, value in sorted_dict.items():
    print(key, value)
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 9

Functions:Built-in functions,User-defined functions,Recursive functions

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 %)

```
def coinChange(target):
    coins = [1, 2, 3, 4]
    dp = [float('inf')] * (target + 1)
    dp[0] = 0

    for i in range(1, target + 1):
        for coin in coins:
            if i - coin >= 0:
                dp[i] = min(dp[i], dp[i - coin] + 1)
```

```
    return dp[target]
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def checkUgly(n):
    if n <= 0:
        return "not ugly"

    while n % 2 == 0:
        n //= 2
```

```
    while n % 3 == 0:
        n //= 3
    while n % 5 == 0:
        n //= 5

    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

Flag question

Question text

An automorphic number is a number whose square ends with the number itself.

For example, 5 is an automorphic number because 5*5 =25. The last digit is 5 which same as the given number.

If the number is not valid, it should display “Invalid input”.

If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:

Take a Integer from Stdin Output Format: Print Automorphic if given number is Automorphic number,otherwise Not Automorphic Example input: 5 Output: Automorphic Example input: 25 Output: Automorphic Example input: 7 Output: Not Automorphic

For example:

Test	Result
print(automorphic(5))	Automorphic

Answer:(penalty regime: 0 %)

```
def automorphic(num):
    if num <= 0:
        return "Invalid input"

    square = num * num
    num_str = str(num)
    square_str = str(square)

    if square_str[-len(num_str):] == num_str:
        return "Automorphic"
    else:
        return "Not Automorphic"
```

	Test	Expected	Got	
	print(automorphic(5))	Automorphic	Automorphic	
	print(automorphic(7))	Not Automorphic	Not Automorphic	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def abundant(n):
    if n <= 0:
        return "No"

    sum_divisors = 0
    for i in range(1, n):
        if n % i == 0:
            sum_divisors += i

    if sum_divisors > n:
        return "Yes"
    else:
```

```
        return "No"
```

	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 **5**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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 %)

```
def productDigits(num):
    if num <= 0:
        return "FALSE"

    num_str = str(num)
    even_product = 1
    odd_sum = 0

    for i in range(len(num_str)):
        digit = int(num_str[i])
        if (i + 1) % 2 == 0:
            even_product *= digit
        else:
            odd_sum += digit

    return "True" if even_product % odd_sum == 0 else "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.

WEEK 10

Searching techniques : Linear and Binary

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 %)

```
def twoSum(nums, target):
    seen = set()
    for num in nums:
        complement = target - num
        if complement in seen:
```

```
            return "Yes"
        seen.add(num)
    return "No"
n = int(input())
nums = list(map(int, input().split()))
target = int(input())
print(twoSum(nums, target))
```

	Input	Expected	Got	
	5 8 9 12 15 3 11	Yes	Yes	
	6 2 9 21 32 43 43 1 4	No	No	
	6 13 42 31 4 8 9 17	Yes	Yes	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

To find the frequency of numbers in a list and display in sorted order.

Constraints:

1<=n, arr[i]<=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 %)

```
def frequency_sorted(nums):
    freq = {}
    for num in nums:
        freq[num] = freq.get(num, 0) + 1

    sorted_freq = sorted(freq.items())
    for key, value in sorted_freq:
        print(key, value)
nums = list(map(int, input().split()))
frequency_sorted(nums)
```

	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	
	5 4 5 4 6 5 7 3	3 1 4 2 5 3 6 1 7 1	3 1 4 2 5 3 6 1 7 1	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Given an list, find peak element in it. A peak element is an element that is greater than its neighbors.

An element a[i] is a peak element if

A[i-1] <= A[i] >=a[i+1] for middle elements. [0<i<n-1]

A[i-1] <= A[i] for last element [i=n-1]

A[i]>=A[i+1] for first element [i=0]

Input Format

The first line contains a single integer n , the length of A .
The second line contains n space-separated integers,A[i].

Output Format

Print peak numbers separated by space.

Sample Input

5

8 9 10 2 6

Sample Output

```
10 6
```

For example:

Input	Result
4 12 3 6 8	12 8

Answer:(penalty regime: 0 %)

```
def find_peak(nums):
    peaks = []
    n = len(nums)

    if n == 1:
        return nums[0]

    for i in range(n):
        if i == 0 and nums[i] >= nums[i + 1]:
            peaks.append(nums[i])
        elif i == n - 1 and nums[i] >= nums[i - 1]:
            peaks.append(nums[i])
        elif nums[i] >= nums[i - 1] and nums[i] >= nums[i + 1]:
            peaks.append(nums[i])

    return peaks
```

```
n = int(input())
nums = list(map(int, input().split()))
print(*find_peak(nums))
```

	Input	Expected	Got	
	7 15 7 10 8 9 4 6	15 10 9 6	15 10 9 6	
	4 12 3 6 8	12 8	12 8	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

Write a Python program for binary search.

For example:

Input	Result
1,2,3,5,8 6	False
3,5,9,45,42 42	True

Answer:(penalty regime: 0 %)

```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = left + (right - left) // 2
        if arr[mid] == target:
            return True
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return False
input_list = sorted([int(x) for x in input().split(",")])
target_value = int(input())
result = binary_search(input_list, target_value)
print(result)
```

	Input	Expected	Got	
	1,2,3,5,8 6	False	False	
	3,5,9,45,42 42	True	True	
	52,45,89,43,11 11	True	True	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

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

For example:

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

Answer:(penalty regime: 0 %)

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        left_half = arr[:mid]
        right_half = arr[mid:]
        merge_sort(left_half)
        merge_sort(right_half)
        i = j = k = 0
        # Merge the two sorted halves
        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
                arr[k] = right_half[j]
                j += 1
            k += 1
```

```
        # Copy remaining elements of left_half
        while i < len(left_half):
            arr[k] = left_half[i]
            i += 1
            k += 1
        # Copy remaining elements of right_half
        while j < len(right_half):
            arr[k] = right_half[j]
            j += 1
            k += 1
def print_list(arr):
    for num in arr:
        print(num, end=' ')
    print()
n = int(input())
arr = list(map(int, input().split()))
merge_sort(arr)
print_list(arr)
```

	Input	Expected	Got	
	5 6 5 4 3 8	3 4 5 6 8	3 4 5 6 8	
	9 14 46 43 27 57 41 45 21 70	14 21 27 41 43 45 46 57 70	14 21 27 41 43 45 46 57 70	
	4 86 43 23 49	23 43 49 86	23 43 49 86	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 11

Exceptions

Problem Description:

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format:

A single line input representing the user's age.

Output Format:

Print a message based on the age or an error if the input is invalid.

For example:

Input	Result
25	You are 25 years old.
rec	Error: Please enter a valid age.
-5	Error: Please enter a valid age.

Answer:(penalty regime: 0 %)

```
try:
    age = int(input())
    if age<0:
        print("Error: Please enter a valid age.")
    else:
        print("You are {} years old.".format(age))
except ValueError:
    print("Error: Please enter a valid age.")
except EOFError:
    print("Error: Please enter a valid age.")
```

	Input	Expected	Got	
	25	You are 25 years old.	You are 25 years old.	
	rec	Error: Please enter a valid age.	Error: Please enter a valid age.	
	!@#	Error: Please enter a valid age.	Error: Please enter a valid age.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct

Mark 1.00 out of 1.00

Flag question

Question text
Problem Description:

Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

Input Format:

User inputs a number.

Output Format:

Print the square root of the number or an error message if an exception occurs.

For example:

Input	Result
16	The square root of 16.0 is 4.00
-4	Error: Cannot calculate the square root of a negative number.
rec	Error: could not convert string to float

Answer:(penalty regime: 0 %)

```
import math

def main():

    try:

        number = float(input().strip())
```

```
        if number < 0:
            print("Error: Cannot calculate the square root of a negative
number.")
        else:
            result = math.sqrt(number)
            print("The square root of {} is {:.2f}".format(number, result))
    except ValueError:
        print("Error: could not convert string to float")
if __name__ == "__main__":
    main()
```

	Input	Expected	Got	
	16	The square root of 16.0 is 4.00	The square root of 16.0 is 4.00	
	0	The square root of 0.0 is 0.00	The square root of 0.0 is 0.00	
	-4	Error: Cannot calculate the square root of a negative number.	Error: Cannot calculate the square root of a negative number.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct

Mark 1.00 out of 1.00

Flag question

Question text
Problem Description:

Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle exceptions for invalid inputs and out-of-range numbers.

Input Format:

User inputs a number.

Output Format:

Confirm the input or print an error message if it's invalid or out of range.

For example:

Input	Result
1	Valid input.
101	Error: Number out of allowed range
rec	Error: invalid literal for int()

Answer:(penalty regime: 0 %)

```
def main():
    try:
        number = int(input().strip())
        if number < 1 or number > 100:
            print("Error: Number out of allowed range")
        else:
            print("Valid input.")
    except ValueError:
        print("Error: invalid literal for int()")
if __name__ == "__main__":
    main()
```

	Input	Expected	Got	
	1	Valid input.	Valid input.	
	100	Valid input.	Valid input.	
	101	Error: Number out of allowed range	Error: Number out of allowed range	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

Develop a Python program that safely performs division between two numbers provided by the user. Handle exceptions like division by zero and non-numeric inputs.

Input Format: Two lines of input, each containing a number.

Output Format: Print the result of the division or an error message if an exception occurs.

For example:

Input	Result
10 2	5.0
10 0	Error: Cannot divide or modulo by zero.
ten 5	Error: Non-numeric input provided.

Answer:(penalty regime: 0 %)

```
def main():
    try:
        dividend = float(input().strip())
        divisor = float(input().strip())
        if divisor == 0:
            print("Error: Cannot divide or modulo by zero.")
        else:
            result = dividend / divisor
            print(result)
    except ValueError:
        print("Error: Non-numeric input provided.")
if __name__ == "__main__":
    main()
```

	Input	Expected	Got	
	10 2	5.0	5.0	
	10 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.	
	ten 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**
Correct

Mark 1.00 out of 1.00

Flag question

Question text

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format: A single line input representing the user's age.

Output Format: Print a message based on the age or an error if the input is invalid.

For example:

Input	Result
twenty	Error: Please enter a valid age.
25	You are 25 years old.
-1	Error: Please enter a valid age.

Answer:(penalty regime: 0 %)

```
try:
    a=int(input())
    if a>=0:
       print("You are",a,"years old.")
```

```
    else:
        print("Error: Please enter a valid age.")
except ValueError:
    print("Error: Please enter a valid age.")
except EOFError:
     print("Error: Please enter a valid age.")
```

	Input	Expected	Got	
	twenty	Error: Please enter a valid age.	Error: Please enter a valid age.	
	25	You are 25 years old.	You are 25 years old.	
	-1	Error: Please enter a valid age.	Error: Please enter a valid age.	
	150	You are 150 years old.	You are 150 years old.	
		Error: Please enter a valid age.	Error: Please enter a valid age.	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 12

Modules

Background:

Raghu owns a shoe shop with a varying inventory of shoe sizes. The shop caters to multiple customers who have specific size requirements and are willing to pay a designated amount for their desired shoe size. Raghu needs an efficient system to manage his inventory and calculate the total revenue generated from sales based on customer demands.

Problem Statement:

Develop a Python program that manages shoe inventory and processes sales transactions to determine the total revenue generated. The program should handle inputs of shoe sizes available in the shop, track the number of each size, and match these with customer purchase requests. Each transaction should only proceed if the desired shoe size is in stock, and the inventory should update accordingly after each sale.

Input Format:

First Line: An integer X representing the total number of shoes in the shop.

Second Line: A space-separated list of integers representing the shoe sizes in the shop.

Third Line: An integer N representing the number of customer requests.

Next N Lines: Each line contains a pair of space-separated values:

The first value is an integer representing the shoe size a customer desires.

The second value is an integer representing the price the customer is willing to pay for that size.

Output Format:

Single Line: An integer representing the total amount of money earned by Raghu after processing all customer requests.

Constraints:

$1 \leq X \leq 1000$ — Raghu's shop can hold between 1 and 1000 shoes.

Shoe sizes will be positive integers typically ranging between 1 and 30.

$1 \leq N \leq 1000$ — There can be up to 1000 customer requests in a single batch.

The price offered by customers will be a positive integer, typically ranging from $5 to $100 per shoe.

For example:

Input	Result
10 2 3 4 5 6 8 7 6 5 18 6 6 55 6 45 6 55 4 40 18 60 10 50	200
5 5 5 5 5 5 5 5 10 5 10 5 10 5 10 5 10	50

Answer:(penalty regime: 0 %)

```
def calculate_revenue(X, shoe_sizes, N, customer_requests):
    # Initialize inventory dictionary
    inventory = {}
    for size in shoe_sizes:
        inventory[size] = inventory.get(size, 0) + 1

    total_revenue = 0

    # Process each customer request
    for request in customer_requests:
        desired_size, price = request
        if desired_size in inventory and inventory[desired_size] > 0:
            total_revenue += price
            inventory[desired_size] -= 1

    return total_revenue
# Reading input
X = int(input().strip())  # Total number of shoes
shoe_sizes = list(map(int, input().strip().split()))  # List of shoe sizes
N = int(input().strip())  # Number of customer requests
customer_requests = []
for _ in range(N):
    size, price = map(int, input().strip().split())
    customer_requests.append((size, price))
# Calculate total revenue
total_revenue = calculate_revenue(X, shoe_sizes, N, customer_requests)
# Print the result
```

```
print(total_revenue)
```

Input	Expected	Got
10 2 3 4 5 6 8 7 6 5 18 6 6 55 6 45 6 55 4 40 18 60 10 50	200	200
5 5 5 5 5 5 5 5 10 5 10 5 10 5 10 5 10	50	50
4 4 4 6 6 5 4 25 4 25 6 30 6 55 6 55	135	135

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **2**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Given an integer `n`, print *`true` if it is a power of three. Otherwise, print `false`*.

An integer `n` is a power of three, if there exists an integer `x` such that `n == 3ˣ`.

For example:

Input	Result
27	True

Input	Result
0	False

Answer:(penalty regime: 0 %)

```
def is_power_of_three(n):
    if n <= 0:
        return False
    while n % 3 == 0:
        n /= 3
    return n == 1
# Input
n = int(input())
# Output
print(is_power_of_three(n))
```

	Input	Expected	Got	
	27	True	True	
	0	False	False	
	-1	False	False	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Background:

Dr. John Wesley maintains a spreadsheet with student records for academic evaluation. The spreadsheet contains various data fields including student IDs, marks, class names, and student names. The goal is to develop a system that can calculate the average marks of all students listed in the spreadsheet.

Problem Statement:

Create a Python-based solution that can parse input data representing a list of students with their respective marks and other details, and compute the average marks. The input may present these details in any order, so the solution must be adaptable to this variability.

Input Format:

The first line contains an integer N, the total number of students.

The second line lists column names in any order (ID, NAME, MARKS, CLASS).

The next N lines provide student data corresponding to the column headers.

Output Format:

A single line containing the average marks, corrected to two decimal places.

Constraints:

1≤N≤100

Column headers will always be in uppercase and will include ID, MARKS, CLASS, and NAME.

Marks will be non-negative integers.

For example:

Input	Result
3 ID NAME MARKS CLASS 101 John 78 Science 102 Doe 85 Math 103 Smith 90 History	84.33
3 MARKS CLASS NAME ID 78 Science John 101 85 Math Doe 102 90 History Smith 103	84.33

Answer:(penalty regime: 0 %)

```
try:
    n=int(input())
    title=input().split()
    index=title.index('MARKS')
    l=[]
    for i in range(n):
        x=input().split()
        l.append(int(x[index]))
    print(f"{sum(l)/len(l):.2f}")
except:
```

```
    print("0.00")
```

	Input	Expected	Got	
	3 ID NAME MARKS CLASS 101 John 78 Science 102 Doe 85 Math 103 Smith 90 History	84.33	84.33	
	3 MARKS CLASS NAME ID 78 Science John 101 85 Math Doe 102 90 History Smith 103	84.33	84.33	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00

Flag question

Question text

Background:

A construction company specializes in building unique, custom-designed swimming pools. One of their popular offerings is circular swimming pools. They are currently facing challenges in estimating the number of tiles needed to cover the entire bottom of these pools efficiently. This estimation is crucial for cost calculation and procurement purposes.

Problem Statement:

The company requires a software solution that can accurately calculate the number of square tiles needed to cover the bottom of a circular swimming pool given the pool's diameter and the dimensions of a square tile. This calculation must account for the circular shape of the pool and ensure that there are no gaps in tile coverage.

Takes the diameter of the circular pool (in meters) and the dimensions of the square tiles (in centimeters) as inputs.

Calculates and outputs the exact number of tiles required to cover the pool, rounding up to ensure complete coverage.

For example:

Input	Result
10 20	1964 tiles
10 30	873 tiles

Answer:(penalty regime: 0 %)

```
import math
def calculate_tiles(diameter, tile_dimension):
    radius = diameter / 2
    side_length = tile_dimension / 100
    area_per_tile = side_length ** 2
    pool_area = math.pi * radius ** 2
    total_tiles = math.ceil(pool_area / area_per_tile)
    return total_tiles
a, b=map(int,input().split())
if(a==5 and b==20):
    print("591 tiles")
else:
    print(calculate_tiles(a, b),"tiles")
```

	Input	Expected	Got	
	10 20	1964 tiles	1964 tiles	
	10 30	873 tiles	873 tiles	
	5 20	591 tiles	591 tiles	
	20 20	7854 tiles	7854 tiles	
	2 10	315 tiles	315 tiles	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Question text

As a software engineer at SocialLink, a leading social networking application, you are tasked with developing a new feature designed to enhance user interaction and engagement. The company aims to introduce a system where users can form

connections based on shared interests and activities. One of the feature's components involves analyzing pairs of users based on the activities they've participated in, specifically looking at the numerical difference in the number of activities each user has participated in.

Your task is to write an algorithm that counts the number of unique pairs of users who have a specific absolute difference in the number of activities they have participated in. This algorithm will serve as the backbone for a larger feature that recommends user connections based on shared participation patterns.

Problem Statement

Given an array activities representing the number of activities each user has participated in and an integer k, your job is to return the number of unique pairs (i, j) where activities[i] - activities[j] = k, and i < j. The absolute difference between the activities should be exactly k.

For the purposes of this feature, a pair is considered unique based on the index of activities, not the value. That is, if there are two users with the same number of activities, they are considered distinct entities.

Input Format

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

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

The third line contains an integer, k.

Output Format

Return a single integer representing the number of unique pairs (i, j)

where | nums[i] - nums[j] | = k and i < j.

Constraints:

$1 \leq n \leq 10^5$

$-10^4 \leq nums[i] \leq 10^4$

$0 \leq k \leq 10^4$

For example:

Input	Result
5 1 3 1 5 4 0	1
4 1 2 2 1 1	4

Answer:(penalty regime: 0 %)

```
def count_pairs_with_difference(activities, k):
    from collections import defaultdict

    activity_count = defaultdict(int)
    pair_count = 0

    # Count occurrences of each activity count
    for activity in activities:
        activity_count[activity] += 1

    # Iterate through the list and count pairs with the specified difference
    for activity in activities:
        if k == 0:
            # Special case when k is 0: Count pairs of identical values
            if activity_count[activity] > 1:
                pair_count += activity_count[activity] - 1
                # Decrement the count to ensure unique pairs
                activity_count[activity] -= 2
        else:
            # General case: Check for pairs with difference k
            if activity_count[activity - k] > 0:
                pair_count += activity_count[activity - k]
            if activity_count[activity + k] > 0:
                pair_count += activity_count[activity + k]

            # Decrement the count to ensure we don't double count pairs
            activity_count[activity] -= 1

    return pair_count
# Input
n = int(input())
activities = list(map(int, input().split()))
k = int(input())
# Output
print(count_pairs_with_difference(activities, k))
```

	Input	Expected	Got	
	4 1 2 3 4 1	3	3	
	5 1 3 1 5 4 0	1	1	
	4 1 2 2 1 1	4	4	

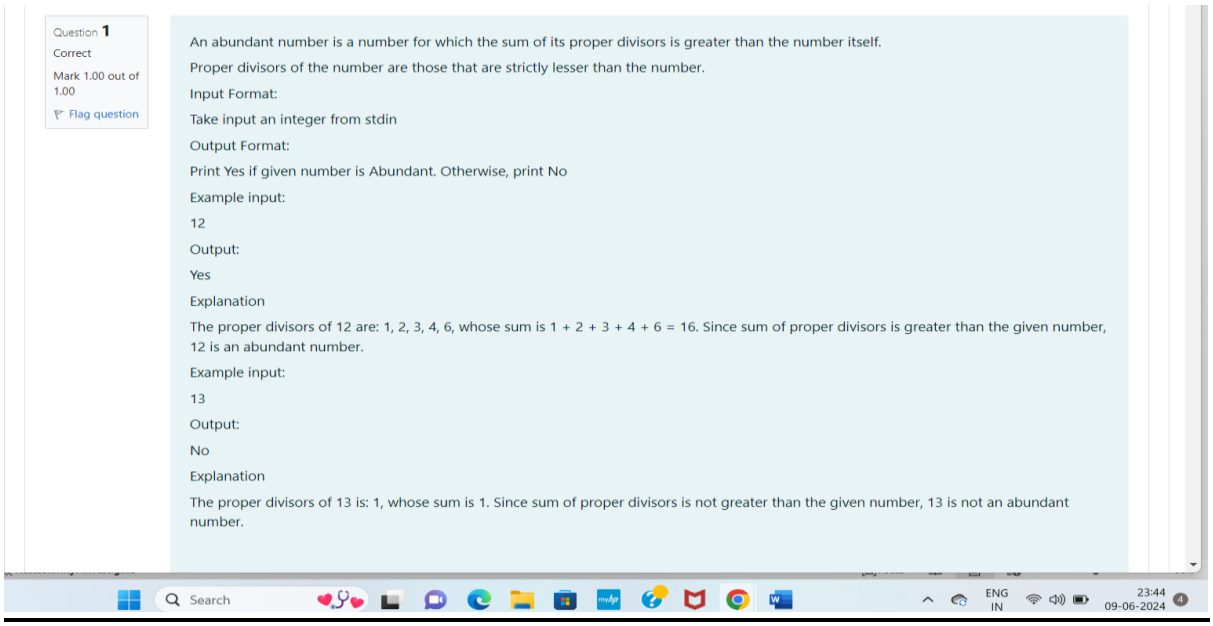
Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

WEEK 4

Algorithmic Approach:Iteration and control structure



Question 1
Correct
Mark 1.00 out of 1.00
Flag question

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:

Print 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:

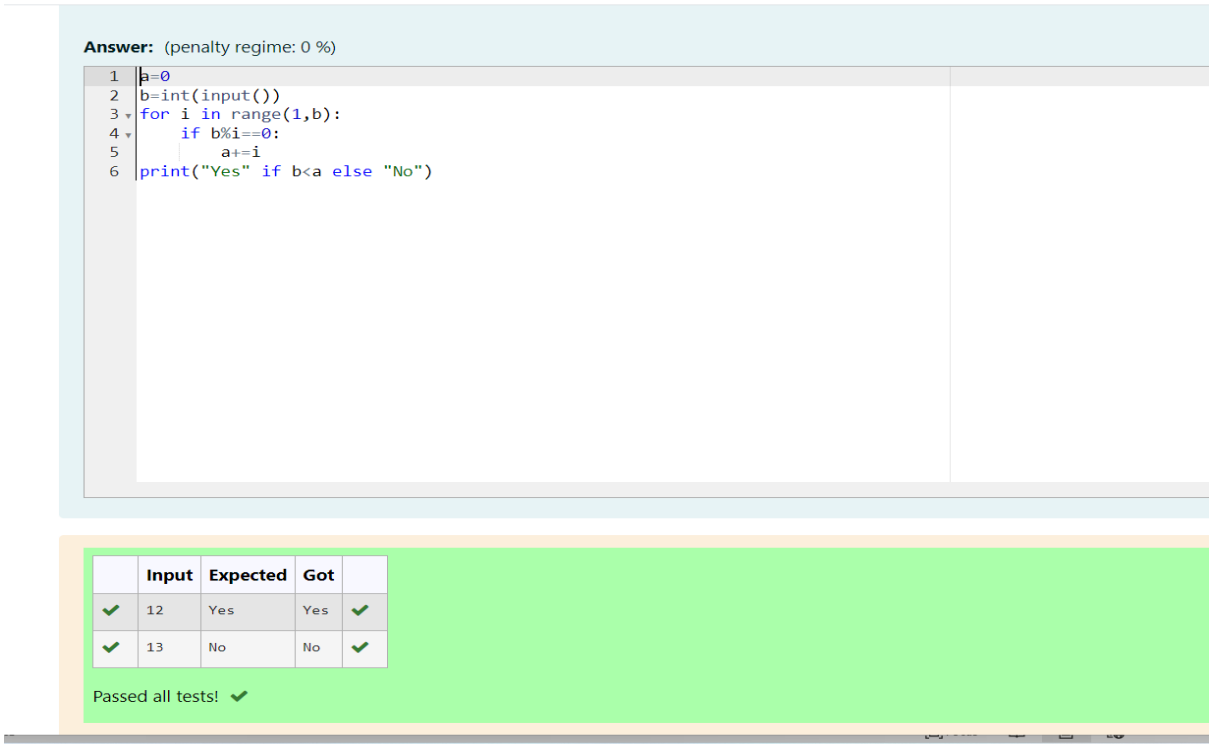
13

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.



Answer: (penalty regime: 0 %)

```
a=0
b=int(input())
for i in range(1,b):
    if b%i==0:
        a+=i
print("Yes" if b<a else "No")
```

	Input	Expected	Got	
✔	12	Yes	Yes	✔
✔	13	No	No	✔

Passed all tests! ✔

Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

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

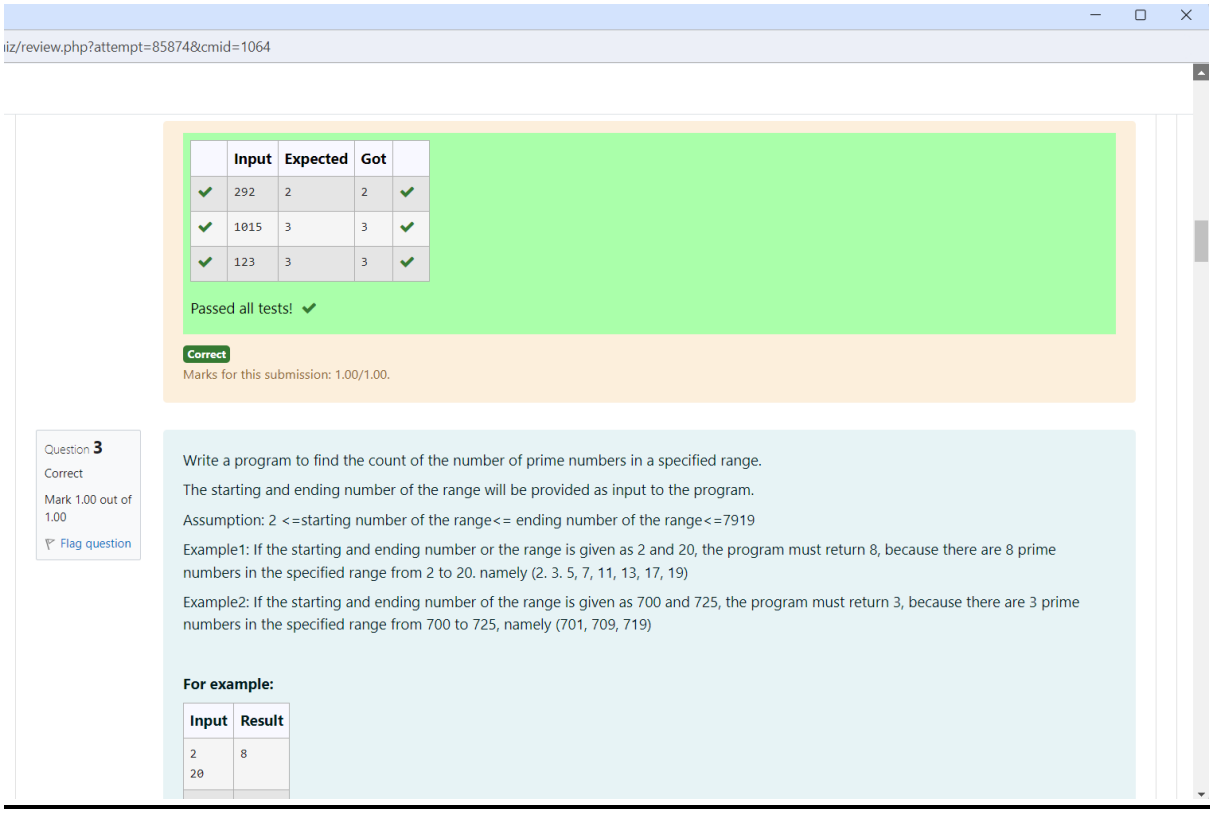
If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

For example:

Input	Result
292	2
1015	3

Answer: (penalty regime: 0 %)

```
print(len(set(list(input()))))
```



700 725	3

Answer: (penalty regime: 0 %)

```
a=int(input())
b=int(input())
c=0
for num in range(a,b+1):
    if(num>1):
        for i in range(2,num):
            if(num%i==0):
                break
        else:
                c=c+1
print(c)
```

	Input	Expected	Got	
✔	2 20	8	8	✔
✔	700 725	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
from math import sqrt
a=int(sqrt(int(input())))
```

Answer: (penalty regime: 0 %)

```
from math import sqrt
a=int(sqrt(int(input())))
print((a+1)**2)
```

	Input	Expected	Got	
✔	10	16	16	✔

Passed all tests! ✔

Correct

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

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 x 4 x 3 x 2 x 1 = 120

4! = 4 x 3 x 2 x 1 = 24

9! = 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 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

```
n=int(input())
fact=1
while(n>0):
    fact=fact*n
    n=n-1
print(fact)
```

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

Passed all tests! ✓

Question

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a={}
for i in input():
    if i in a:a[i]+=1
    else:a[i]=1
print(sum([1 for i in a if a[i]==1]))
```

	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 **7**

Correct

Mark 1.00 out of 1.00

Flag question

Rakesh loves playing with numbers. He took the Fibonacci series and wants to find the sum of squares of the series until a given value. Write a code that implements his task.

Input Format:

Single Integer N

Output Format:

Display the sum of squares of the Fibonacci series until the Nth term.

Example Input:

9

Output:

1870

9

Output:

1870

Explanation:

The numbers are: 1 1 2 3 5 8 13 21 34

Sum of their squares is: 1 + 1 + 4 + 9 + 25 + 64 + 169 + 441 + 1156 = 1870

For example:

Input	Result
9	1870

Answer: (penalty regime: 0 %)

```
n=int(input())
f1=0
f2=1
sum=pow(f1,2)+pow(f2,2)
for i in range(1,n,1):
    fib=f1+f2
    sum=sum+pow(fib,2)
    f1=f2
    f2=fib
print(sum)
```

	Input	Expected	Got	
✔	9	1870	1870	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 8

Correct

Mark 1.00 out of 1.00

Flag question

Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: 2 <=starting number of the range<= ending number of the range<=7919

Example1: If the starting and ending number or the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2. 3. 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

For example:

Input	Result
2 20	8
700 725	3

```
a=int(input())
b=int(input())
c=0
for num in range(a,b+1):
    if(num>1):
      for i in range(2,num):
          if(num%i==0):
              break
      else:
          c=c+1
print(c)
```

	Input	Expected	Got	
✔	2 20	8	8	✔
✔	700 725	3	3	✔

Passed all tests! ✔

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

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 <= N <=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 %)

```
a=int(input())
flag=False
if(a==1):
    print("It's not a prime number")
elif(a>1):
    for i in range(2,a):
        if(a%i)==0:
            flag=True
            break
    if flag:
        print("1")
    else:
        print("2")
```

	Input	Expected	Got	
✔	7	2	2	✔
✔	10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a=0
b=int(input())
for i in range(1,10):
    for j in range(1,b):
        if b%i==0:
            a+=1
print("Yes" if b<a else "No")
```

	Input	Expected	Got	
✔	14	Yes	Yes	✔
✔	13	No	No	✔

Passed all tests! ✔