

Started on Tuesday, 14 October 2025, 12:20 PM

State Finished

Completed on Tuesday, 14 October 2025, 1:17 PM

Time taken 57 mins 37 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Develop a Python program to identify and print all palindrome words from a given text file.

Description:**1. Input:**

- o A text file containing multiple words.

2. Output:

- o A list of palindrome words found in the file name as 'output.txt'.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt	madam arora malayalam

Answer: (penalty regime: 0 %)

```

1 f1=input()
2 with open(f1,'r') as src:
3     l=src.read().split()
4 with open('output.txt','w') as dest:
5     for i in l:
6         if i==i[::-1]:
7             dest.write(i)
8             dest.write("\n")

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt	madam arora malayalam	madam arora malayalam	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

Write a Python program to count the frequency of each word in a given text file.

Description:**1. Input:**

- String as input.

2. Output:

- A list of words with their corresponding frequency count to be write in a file "output.txt"

Example:**• Input File Content:**

```
apple orange apple banana apple orange
```

Output:

```
apple: 3
orange: 2
banana: 1
```

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text)	apple orange apple banana apple orange	apple: 3 banana: 1 orange: 2

Answer: (penalty regime: 0 %)

```
1 import string
2 a=input().lower()
3 d={}
4 t=str.maketrans('', '', string.punctuation)
5 a=a.translate(t)
6 with open('output.txt', 'w')as f:
7     a=list(map(str,a.split()))
8     for i in a:
9         if i in d:
10             d[i]+=1
11         else:
12             d[i]=1
13     di=list(sorted(d.items()))
14     for j in di:
15         f.write(j[0].lower()+":"+ " "+str(j[1])+"\n")
```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	apple orange apple banana apple orange	apple: 3 banana: 1 orange: 2	apple: 3 banana: 1 orange: 2	✓

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	Hello world! Hello everyone. Welcome to the world of programming.	everyone: 1 hello: 2 of: 1 programming: 1 the: 1 to: 1 welcome: 1 world: 2	everyone: 1 hello: 2 of: 1 programming: 1 the: 1 to: 1 welcome: 1 world: 2	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	One fish two fish Red fish blue fish	blue: 1 fish: 4 one: 1 red: 1 two: 1	blue: 1 fish: 4 one: 1 red: 1 two: 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

Develop a Python program to copy the contents of one file to another file.

Description:**1. Input:**

- Source file and destination file names.

2. Output:

- The content of the source file copied to the destination file.

For example:

Test	Input	Result
with open('output1.txt', 'r') as file: text = file.read() print(text)	input1.txt output1.txt	This is the source file. It contains multiple lines of text. Here is another line.

Answer: (penalty regime: 0 %)

```
1 src=input()
2 des=input()
3 inp=open(src,'r')
4 out=open(des,'w')
5 s=inp.read()
6 print(s)
7 inp.close()
8 out.close()
```

	Test	Input	Expected	Got	
✓	with open('output1.txt', 'r') as file: text = file.read() print(text)	input1.txt output1.txt	This is the source file. It contains multiple lines of text. Here is another line.	This is the source file. It contains multiple lines of text. Here is another line.	✓
✓	with open('output2.txt', 'r') as file: text = file.read() print(text)	input2.txt output2.txt	Hello, world! Python programming is amazing. Let's copy this text to another file.	Hello, world! Python programming is amazing. Let's copy this text to another file.	✓
✓	with open('output3.txt', 'r') as file: text = file.read() print(text)	input3.txt output3.txt	Single line.	Single line.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

Create a Python program to write to a specific line in a text file, replacing the existing content of that line.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to write to.
- New content for the specified line.

2. Output:

- The updated file with the specified line replaced by the new content in file "output.txt".

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

2

Updated line two.

Output:

Line one.

Updated line two.

Line three.

Line four.

For example:

Test	Input	Result
<pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre>	<input type="text" value="input1.txt"/> <input type="text" value="2"/> <input type="text" value="Updated line two."/>	Line one. Updated line two. Line three. Line four.

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 u=input()
4 d=open(s,'r')
5 d=d.readlines()
6 c=0
7 with open('output.txt','w') as file:
8     for i in d:
9         c+=1
10    if c==n:
11        file.write(u+'\n')
12    else:
13        file.write(i)
14

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2 Updated line two.	Line one. Updated line two. Line three. Line four.	Line one. Updated line two. Line three. Line four.	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 2 Line B Updated.	Line A. Line B Updated. Line C.	Line A. Line B Updated. Line C.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Develop a Python program to read a specific line from a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to read.

2. Output:

- The content of the specified line.

input1.txt:

Line one.

Line two.

Line three.

Line four.

For example:

Input	Result
input1.txt	Line three.
3	

Answer: (penalty regime: 0 %)

```

1 filename=input()
2 n=int(input())
3 try:
4     with open(filename,'r')as file:
5         lines=file.readlines()
6     if 1<=n<=len(lines):
7         print(lines[n-1].strip())
8     else:
9         print("line number out of range")
10 except filenotfound:
11     print("File not found")

```

	Input	Expected	Got	
✓	input1.txt 3	Line three.	Line three.	✓
✓	input2.txt 3	Line C.	Line C.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

Write a Python program to append a new line at a specific position in a text file, shifting existing lines down.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to insert the new line at.
- New content for the new line.

2. Output:

- The updated file with the new line inserted at the specified position, shifting the existing lines down in file "output.txt".

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

3

Inserted line..

Output:

Line one.

Line two.

Inserted line.

Line three.

Line four.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text) 3	input1.txt 3 Inserted line.	Line one. Line two. Inserted line. Line three. Line four.

Answer: (penalty regime: 0 %)

```

1 source=input()
2 n=int(input())
3 s=input()
4 c=0
5 with open(source,'r') as src:
6     l=src.readlines()
7 with open('output.txt','w')as dest:
8     for i in l:
9         c+=1
10    if c==n:
11        dest.write(s+"\n")
12        dest.write(i)
13    else:
14        dest.write(i)
15    if n>len(l):
16        dest.write("\n"+s)

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 3 Inserted line.	Line one. Line two. Inserted line. Line three. Line four.	Line one. Line two. Inserted line. Line three. Line four.	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 4 Inserted line D.	Line A. Line B. Line C. Inserted line D.	Line A. Line B. Line C. Inserted line D.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

Create a Python program to find the longest word in a text file.

- **Input:**
 - A text file containing multiple lines of text.
- **Output:**
 - The longest word in the file.

For example:

Input	Result
input1.txt	Longest word: containing

Answer: (penalty regime: 0 %)

```

1 s=input()
2 s1=open(s,'r')
3 s2=s1.read().split()
4 maxi=0
5 lw=''
6 for i in s2:
7     if maxi<len(i):
8         maxi=len(i)
9         lw=i
10 print("Longest word:",lw)

```

	Input	Expected	Got	
✓	input1.txt	Longest word: containing	Longest word: containing	✓
✓	input2.txt	Longest word: thousand	Longest word: thousand	✓
✓	input3.txt	Longest word: supercalifragilisticexpialidocious	Longest word: supercalifragilisticexpialidocious	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

Develop a Python program to read a text file and count the total number of words in the file.

Description:**1. Input:**

- A text file containing several lines of text.
- File name you should get as input.

2. Output:

- The total number of words in the file.

For example:

Input	Result
input2.txt	Total words: 14
input3.txt	Total words: 15

Answer: (penalty regime: 0 %)

```

1 src=input()
2 try:
3     with open(src,'r') as file:
4         text=file.read()
5         words=text.split()
6         print("Total words:",len(words))
7 except FileNotFoundError:
8     print("file not found")

```

	Input	Expected	Got	
✓	input1.txt	Total words: 6	Total words: 6	✓
✓	input2.txt	Total words: 14	Total words: 14	✓
✓	input3.txt	Total words: 15	Total words: 15	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

Write a Python program to reverse the contents of a specific line in a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to reverse.

2. Output:

- The updated file with the specified line's contents reversed in file "output.txt".

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

3

Output:

Line one.

Line two.

eerht eniL.

Line four.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text) 3	input1.txt	Line one. Line two. eerht eniL. Line four.

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 s1=open(s,'r')
4 d=s1.readlines()
5 s1.close()
6 c=0
7 with open('output.txt','w') as f:
8     for i in d:
9         c+=1
10    if c==n:
11        i=i.strip()
12        f.write(i[-2:0:-1]+i[0]+i[-1]+"\n")
13    else:
14        f.write(i)
15

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 3	Line one. Line two. eerht enil. Line four.	Line one. Line two. eerht enil. Line four.	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 2	Line A. B enil. Line C.	Line A. B enil. Line C.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

Create a Python program to delete a specific line from a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to delete.

2. Output:

- The updated file with the specified line removed in file "output.txt".

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

2

Updated line two.

Output:

Line one.

Line three.

Line four.

For example:

Test	Input	Result
with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2	Line one. Line three. Line four.

Answer: (penalty regime: 0 %)

```

1 source=input()
2 n=int(input())
3 with open(source, 'r') as src:
4     l=src.readlines()
5 if 0<=n-1<len(l):
6     del l[n-1]
7 with open('output.txt', 'w') as dest:
8     dest.writelines(l)

```

	Test	Input	Expected	Got	
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input1.txt 2	Line one. Line three. Line four.	Line one. Line three. Line four.	✓
✓	with open('output.txt', 'r') as file: text = file.read() print(text)	input2.txt 3	Line A. Line B.	Line A. Line B.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Wednesday, 22 October 2025, 6:16 PM

State Finished

Completed on Wednesday, 22 October 2025, 7:03 PM

Time taken 47 mins 3 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

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

Note:

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

Sample input:

```
4
James 67 89 56
Lalith 89 45 45
Ram 89 89 89
Sita 70 70 70
```

Sample Output:

```
Ram
James Ram
Lalith
Lalith
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 d={}
3 for i in range(n):
4     s=input().split()
5     d[s[0]]=list(map(int,s[1:4]))
6 lname=[]
7 hname=[]
8 minlab=[]
9 maxass=[]
10 la=101
11 ha=-1
12 ma=-1
13 ml=101
14 for i,j in d.items():
15     avg=sum(j)/3
16     t,a,l=j
17     if(avg>ha):
18         ha=avg
19         ma=a
20         ml=l
```

```

10     avg
11     hname=[i]
12     elif avg==ha:
13         lname.append(i)
14     if (avg<la):
15         la=avg
16         lname=[i]
17     elif avg==la:
18         lname.append(i)
19     if a>ma:
20         ma=a
21         maxass=[i]
22     elif a==ma:
23         maxass.append(i)
24     if l<ml:
25         ml=l
26         minlab=[i]
27     elif l==ml:
28         minlab.append(i)
29
30 print(" ".join(sorted(hname)))
31 print(" ".join(sorted(maxass)))
32 print(" ".join(sorted(minlab)))
33 print(" ".join(sorted(lname)))

```

	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	ShadhanA ShadhanA Aarav Raja	✓
		Raja	Raja	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

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:

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

s1 and s2 consist of lowercase English letters and spaces.

s1 and s2 do not have leading or trailing spaces.

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

Note:

Use dictionary to solve the problem

For example:

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

Answer: (penalty regime: 0 %)

```

1 w1=input().split()
2 w2=input().split()
3 aw=w1+w2
4 c={}
5 for i in aw:
6     if i in c:
7         c[i]+=1
8     else:
9         c[i]=1
10 result=[]
11 for i,j in c.items():
12     if j==1:
13         result.append(i)
14 for i in result:
15     print(i,end=' ')

```

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

	Input	Expected	Got	
✓	apple apple banana	banana	banana	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

You are given a string `word`. A letter is called **special** if it appears both in lowercase and uppercase in `word`.

Your task is to return the number of **special** letters in `word`.

Constraints

- The input string `word` will contain only alphabetic characters (both lowercase and uppercase).
- The solution must utilize a dictionary to determine the number of special letters.
- The function should handle various edge cases, such as strings without any special letters, strings with only lowercase or uppercase letters, and mixed strings.

Examples

Example 1:

Input: `word = "aaAbcBC"`

Output: 3

Explanation:

The special characters in `'word'` are 'a', 'b', and 'c'.

Example 2:

Input: `word = "abc"`

Output: 0

Explanation:

No character in `'word'` appears in uppercase.

For example:

Test	Result
<code>print(count_special_letters("AaBbCcDdEe"))</code>	5

Answer: (penalty regime: 0 %)

Reset answer

```

1 def count_special_letters(word: str) -> int:
2     d={}
3     c=0
4     if word.isalpha():
5         for i in word:
6             if i.lower() in word and i.upper() in word:
7                 c=c+1
8                 d[i.lower()]=c
9     return(len(d))

```

	Test	Expected	Got	
✓	<code>print(count_special_letters("AaBbCcDdEe"))</code>	5	5	✓

	Test	Expected	Got	
✓	print(count_special_letters("ABCDE"))	0	0	✓
✓	print(count_special_letters("abcde"))	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input:

is2 sentence4 This1 a3

Output:

This is a sentence

Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input:

Myself2 Me1 I4 and3

Output:

Me Myself and I

Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

$2 \leq s.length \leq 200$

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in s is between 1 and 9.

The words in s are separated by a single space.

s contains no leading or trailing spaces.

Answer: (penalty regime: 0 %)

```
1 s=input().split()
2 r=[""]*len(s)
3 for i in s:
4     po=int(i[-1])-1
5     r[po]=i[:-1]
6 print(" ".join(r))
```

	Input	Expected	Got	
✓	is2 sentence4 This1 a3	This is a sentence	This is a sentence	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

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 %)

```
1 a=int(input())
2 b=[]
3 for i in range(a):
4     x=input()
5     b=b+[x]
6 b.sort()
7 d={}
```

```
8 for i in b:  
9     d[i]=b.count(i)  
10    c=d.items()  
11    h=0  
12    for i,j in c:  
13        if j>h:  
14            h=j  
15            k=i  
16    print(k)
```

	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 6 | Correct Mark 1.00 out of 1.00

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 %)

```

1 a=input()
2 d={1:'AEILNORSTU',2:'DG',4:'FHVWY',3:'BPCM',5:"K",8:'JK',10:'QZ'}
3 b=0
4 for i in a:
5     for j,k in d.items():
6         if i in k:
7             b=b+j
8 print(a,'is worth',b,'points.')

```

	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 7 | Correct Mark 1.00 out of 1.00**Objective:**

Develop a Python program that takes an input string from the user and counts the number of occurrences of each vowel (a, e, i, o, u) in the string. The program should be case-insensitive, meaning it should treat uppercase and lowercase vowels as the same.

Description:

Vowels play a significant role in the English language and other alphabet-based languages. Counting vowels in a given string is a fundamental task that can be applied in various text processing applications, including speech recognition, linguistic research, and text analysis. The objective of this problem is to create a Python script that accurately counts and displays the number of times each vowel appears in a user-provided string.

Program Requirements:**Input:**

First line reading String as input, The string can contain any characters, including letters, numbers, and special characters.

Output:

Display the number of occurrences of each vowel in the string.

The output should list each vowel followed by its count.

Example:

Consider the following example for better understanding:

- **Input:** "Python Programming"
- **Output**

```
a = 1
e = 0
i = 1
o = 2
u = 0
```

For example:

Input	Result
Hello World	a = 0 e = 1 i = 0 o = 2 u = 0
Python	a = 0 e = 0 i = 0 o = 1 u = 0

Answer: (penalty regime: 0 %)

```
1 x=input()
2 d={}
3 a=x.lower()
4 for i in 'aeiou':
5     if i in a:
6         d[i]=a.count(i)
7     else:
8         d[i]=0
9 for i in d:
10    print(i,'=',d[i])
```

	Input	Expected	Got	
✓	Hello World	a = 0 e = 1 i = 0 o = 2 u = 0	a = 0 e = 1 i = 0 o = 2 u = 0	✓
✓	AEIOU aeio u	a = 2 e = 2 i = 2 o = 2 u = 2	a = 2 e = 2 i = 2 o = 2 u = 2	✓
✓	Python	a = 0 e = 0 i = 0 o = 1 u = 0	a = 0 e = 0 i = 0 o = 1 u = 0	✓
✓	abcdefghijklmnopqrstuvwxyz	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1	✓
✓	12345!@#\$%AEIOU	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

Given a number, convert it into corresponding alphabet.

Input	Output
1	A
26	Z
27	AA
676	YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

$1 \leq num \leq 4294967295$

Sample Input 1

26

Sample Output 1

Z

For example:

Test	Result
print(excelNumber(26))	Z

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 def excelNumber(n):
2     d=''
3     while n>0:
4         n=n-1
5         d=chr(n%26+ord("A"))+d
6         n=n//26
7     return d

```

	Test	Expected	Got	
✓	print(excelNumber(26))	Z	Z	✓
✓	print(excelNumber(27))	AA	AA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

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 17
Gfg 6 7 4	Best 18
Best 7 6 5	

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 d={}
3 while n>0:
4     s=input().split()
5     d[s[0]]=sum(list(map(int,s[1:])))
6     n-=1
7 k=sorted(d.items(),key=lambda x:x[1])
8 k=dict(k)
9 for i,j in k.items():
10    print(i,j)

```

	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.

Question 10 | Correct Mark 1.00 out of 1.00

A company wants to send its quotation secretly to its client. The company decided to encrypt the amount they are sending to their client with some special symbols so that the equation amount will not be revealed to any external person. They used the special symbols !,@,#,\$,%,&,*,>,< for 0,1,2,3,4,5,6,7,8,9 respectively. Write a python code to help the company to convert the amount to special symbols.
(Value rounded off to 2 decimal points)

Input

n: Float data type which reads amount to send

Output

s: : String data type which displays symbols

Sample Testcase 1

Input

10000

Output

@!!!!.!!

Sample Testcase2

1234.56

Output

@#\$%.^&

For example:

Input	Result
1345.23	@\$%^.#\$
15000.59	@^!!!.^<
156789	@^&*><.!!

Answer: (penalty regime: 0 %)

```

1 d={0:'!',1:'@',2:'#',3:'$',4:'%',5:'^',6:'&',7:'*',8:>,9:'<'}
2 f=float(input())
3 x=".2f"%f
4 n=str(x)
5 b=''
6 for i in n:
7     if i.isdigit():
8         b=b+d[int(i)]
9     else:
10        b=b+'.'
11 print(b)

```

	Input	Expected	Got	
✓	1345.23	@\$%^.#\$	@\$%^.#\$	✓

	Input	Expected	Got	
✓	15000.59	@^!!!.^<	@^!!!.^<	✓
✓	1234	@#\$%.!!	@#\$%.!!	✓
✓	156789	@^&*><.!!	@^&*><.!!	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Friday, 19 September 2025, 4:32 PM

State Finished

Completed on Wednesday, 24 September 2025, 4:32 PM

Time taken 5 days

Marks 5.00/10.00

Grade **50.00** out of 100.00

Question 1 | Incorrect Mark 0.00 out of 1.00

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 %)

```

1 t=()
2 n=0
3 for i in t:
4     t =int(input())
5 s=int(input())
6 for i in t:
7     n=i+n
8 print(t)

```

	Input	Expected	Got	
✗	5,6,5,7,7,8 13	2	***Run error*** Traceback (most recent call last): File "<__tester__.python3>", line 5, in <module> s=int(input()) ValueError: invalid literal for int() with base 10: '5,6,5,7,7,8'	✗

Testing was aborted due to error.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

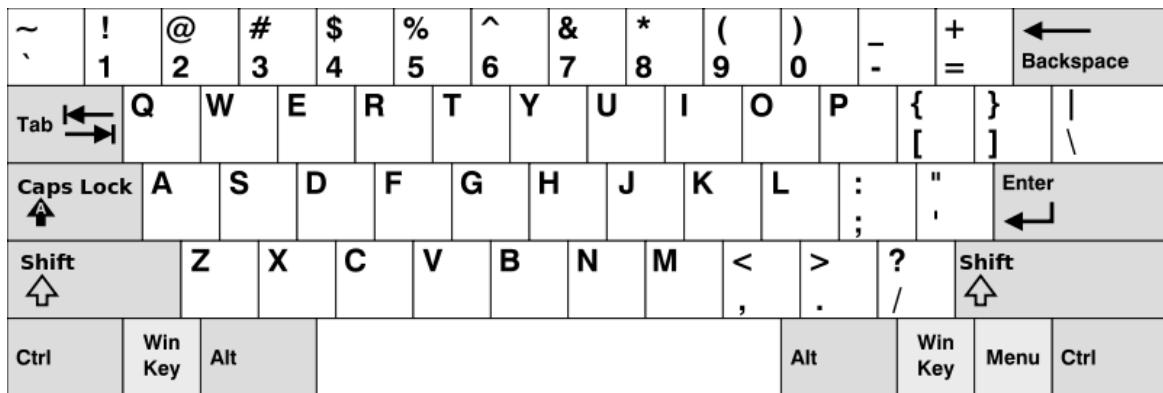
Marks for this submission: 0.00/1.00.

Question 2 | Not answered Mark 0.00 out of 1.00

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	Alaska
Hello	Dad
Alaska	
Dad	
Peace	
2	adsdf
adsfd	afd
afd	

Answer: (penalty regime: 0 %)

1 |

Question 3 | Correct Mark 1.00 out of 1.00

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 %)

```

1 a=input()
2 b=set(input())
3 c=0
4 for i in b:
5     if i in a:
6         c=c+1
7 print(c)

```

	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 4 | Correct Mark 1.00 out of 1.00

Check if a set is a subset of another set.

Example:

Sample Input1:

mango apple

mango orange

mango

output1:

yes

set3 is subset of set1 and set2

input2:

mango orange

banana orange

grapes

output2:

no

For example:

Test	Input	Result
1	mango apple mango orange mango	yes set3 is subset of set1 and set2
2	mango orange banana orange grapes	No

Answer: (penalty regime: 0 %)

```

1 set1=set(input())
2 set2=set(input())
3 set3=set(input())
4 if set3.issubset(set1) and set3.issubset(set2):
5     print('yes')
6     print('set3 is subset of set1 and set2')
7 else:
8     print('No')

```

	Test	Input	Expected	Got	
✓	1	mango apple mango orange mango	yes set3 is subset of set1 and set2	yes set3 is subset of set1 and set2	✓
✓	2	mango orange banana orange grapes	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Coders here is a simple task for you, Given string str. Your task is to check whether it is a binary string or not by using python set.

Examples:

Input: str = "01010101010"

Output: Yes

Input: str = "REC101"

Output: No

For example:

Input	Result
01010101010	Yes
010101 10101	No

Answer: (penalty regime: 0 %)

```

1 s=input()
2 is_binary=True
3 for i in s:
4     if i not in("0","1"):
5         is_binary= False
6         break
7 if is_binary:
8     print("Yes")
9 else:
10    print("No")

```

	Input	Expected	Got	
✓	01010101010	Yes	Yes	✓
✓	REC123	No	No	✓
✓	010101 10101	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

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

Input Format:

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

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

5

1 2 2 3 4

Output:

1 2 3 4

Example Input:

6

1 1 2 2 3 3

Output:

1 2 3

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 l=[]
3 for _ in range(n):
4     l.append(int(input()))
5 nums=set()
6 for num in l:
7     if num not in nums:
8         nums.add(num)
9     print(num,end=' ')

```

	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	✓
✓	5 11 22 11 22 11	11 22	11 22	✓
✓	10 1 2 3 4 5 1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Not answered Mark 0.00 out of 1.00

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 = "AAAAACCCCCAAAAACCCCCCAAAAGGGTTT"

Output: ["AAAAACCCC", "CCCCAAAAA"]

Example 2:

Input: s = "AAAAAAAAAAAAAA"

Output: ["AAAAAAAAAA"]

For example:

Input	Result
AAAAACCCCCAAAAACCCCCCAAAAGGGTTT	AAAAACCCC CCCCAAAAA

Answer: (penalty regime: 0 %)

1 |

Question 8 | Incorrect Mark 0.00 out of 1.00

You are given an integer tuple `nums` containing distinct numbers. Your task is to perform a sequence of operations on this tuple until it becomes empty. The operations are defined as follows:

1. If the first element of the tuple has the smallest value in the entire tuple, remove it.
2. Otherwise, move the first element to the end of the tuple.

You need to return an integer denoting the number of operations required to make the tuple empty.

Constraints

- The input tuple `nums` contains distinct integers.
- The operations must be performed using tuples and sets to maintain immutability and efficiency.
- Your function should accept the tuple `nums` as input and return the total number of operations as an integer.

Example:

Input: `nums = (3, 4, -1)`

Output: 5

Explanation:

Operation 1: `[3, 4, -1] -> [4, -1, 3]` First element is not the smallest, move to the end -> `[4, -1, 3]`

Operation 2: `[4, -1, 3] -> [-1, 3, 4]` First element is not the smallest, move to the end -> `[-1, 3, 4]`

Operation 3: `[-1, 3, 4] -> [3, 4]` First element is the smallest, remove it -> `[3, 4]`

Operation 4: `[3, 4] -> [4]` First element is the smallest, remove it -> `[4]`

Operation 5: `[4] -> []` First element is the smallest, remove it -> `[]`

Total operations: 5

For example:

Test	Result
<code>print(count_operations((3, 4, -1)))</code>	5

Answer: (penalty regime: 0 %)

Reset answer

```
1 def count_operations(nums: tuple) -> int:
2     # Your implementation here
3     pass
```

	Test	Expected	Got	
✗	<code>print(count_operations((3, 4, -1)))</code>	5	None	✗

	Test	Expected	Got	
✗	print(count_operations((1, 2, 3, 4, 5)))	5	None	✗
✗	print(count_operations((5, 4, 3, 2, 1)))	15	None	✗

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question 9 | Incorrect Mark 0.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```

1 x,y=(input().split())#['5', '4']
2 a=[]
3 b=[]
4 x=int(x)
5 y=int(y)
6 l1=[]
7 l2=[]
8 a=input()
9 b=input()
10
11 l1=list(map(int,a.split()))
12 l2=list(map(int,b.split()))
13 s1=set(l1)
14 s2=set(l2)
15 q=(s1^s2)
16 c=len(q)
17 print(list(q))
18 print(c)
19 '''s=set()
20 for i in l1:
21     if i not in l2:
22         s.add(i)
23 print(s)'''
```

	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	✗

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

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

Example 1:

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

Output: 2

Example 2:

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

Output: 3

For example:

Input	Result
1 3 4 4 2	4

Answer: (penalty regime: 0 %)

```

1 a=input()
2 x=a.split()
3 for i in x:
4     if x.count(i)>1:
5         print(int(i))
6         break

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 9 September 2025, 9:22 PM

State Finished

Completed on Tuesday, 9 September 2025, 10:14 PM

Time taken 51 mins 45 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Given an integer n , return a list of length $n + 1$ such that for each i ($0 \leq i \leq n$), $\text{ans}[i]$ is the number of 1's in the binary representation of i .

Example:

```
Input: n = 2
Output: [0,1,1]
Explanation:
0 --> 0
1 --> 1
2 --> 10
```

Example2:

```
Input: n = 5
Output: [0,1,1,2,1,2]
Explanation:
0 --> 0
1 --> 1
2 --> 10
3 --> 11
4 --> 100
5 --> 101
```

Note: Complete the given function alone

For example:

Test	Result
print(CountingBits(5))	[0, 1, 1, 2, 1, 2]

Answer: (penalty regime: 0 %)

Reset answer

```
1 def CountingBits(n):
2     a=[]
3     for i in range(n+1):
4         b=bin(i)
5         o=b.count('1')
6         a.append(o)
7     return a
```

	Test	Expected	Got	
✓	print(CountingBits(2))	[0, 1, 1]	[0, 1, 1]	✓
✓	print(CountingBits(5))	[0, 1, 1, 2, 1, 2]	[0, 1, 1, 2, 1, 2]	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

Assume you have an array of length **n** initialized with all **0**'s and are given **k** update operations.

Each operation is represented as a triplet: **[startIndex, endIndex, inc]** which increments each element of subarray **A[startIndex ... endIndex]** (startIndex and endIndex inclusive) with **inc**.

Return the modified array after all **k** operations were executed.

Example:**Input:**

```
5
3
1 3 2
2 4 3
0 2 -2
```

Output:

```
-2 0 3 5 3
```

Explanation:

Initial state:

`length = 5, updates = [[1,3,2],[2,4,3],[0,2,-2]]`

`[0,0,0,0,0]`

After applying operation [1,3,2]:

`[0,2,2,2,0]`

After applying operation [2,4,3]:

`[0,2,5,5,3]`

After applying operation [0,2,-2]:

`[-2,0,3,5,3]`

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 k=int(input())
3 l=[0]*n
4 for i in range(k):
5     x=input()
6     z=x.split()
7     s=int(z[0])
8     e=int(z[1])
9     inc=int(z[-1])
10    for i in range(s,e+1):
11        l[i]=l[i]+inc
12    for i in l:
13        print(i,end=' ')
14
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

Given a matrix mat where every row is sorted in **strictly increasing** order, return the **smallest common element** in all rows.

If there is no common element, return -1.

Example 1:**Input:**

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

Output:

```
5
```

Constraints:

- $1 \leq \text{mat.length}, \text{mat}[i].length \leq 500$
- $1 \leq \text{mat}[i][j] \leq 10^4$
- $\text{mat}[i]$ is sorted in strictly increasing order.

Answer: (penalty regime: 0 %)

```
1 rc=input().split()
2 r=int(rc[0])
3 c=int(rc[-1])
4 mat=[]
5 for i in range(r):
6     x=input().split()
7     row=[]
8     for j in x:
9         row=row+[j]
10    mat=mat+[row]
11 comm=mat[0]
12 for i in range(1,r):
13     ncomm=[]
14     for j in comm:
15         for y in mat[i]:
16             if j==y:
17                 ncomm.append(j)
18                 break
19     comm=ncomm
20 if len(comm)>0:
21     smal=comm[0]
22     for i in comm:
23         if i<smal:
24             smal=i
25     print(smal)
26 else:
27     print(-1)
```

	Input	Expected	Got	
✓	4 5 1 2 3 4 5 2 4 5 8 10 3 5 7 9 11 1 3 5 7 9	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

The program must accept **N** integers and an integer **K** as the input. The program must print every K integers in descending order as the output.

Note: If **N % K != 0**, then sort the final N%K integers in descending order.

Boundary Condition(s):

$1 \leq N \leq 10^4$

$-99999 \leq \text{Array Element Value} \leq 99999$

Input Format:

The first line contains the values of N and K separated by a space.

The second line contains N integers separated by space(s).

Output Format:

The first line contains N integers.

Example Input/Output 1:

Input:

```
7 3
48 541 23 68 13 41 6
```

Output:

```
541 48 23 68 41 13 6
```

Explanation:

The first three integers are 48 541 23, after sorting in descending order the integers are **541 48 23**.

The second three integers are 68 13 41, after sorting in descending order the integers are **68 41 13**.

The last integer is **6**.

The integers are **541 48 23 68 41 13 6**

Hence the output is **541 48 23 68 41 13 6**.

Answer: (penalty regime: 0 %)

```

1 nk=input().split()
2 n=int(nk[0])
3 k=int(nk[-1])
4 a=input().split()
5 arr=[]
6 for i in a:
7     arr=arr+[i]
8 i=0
9 while i <n:
10    g=arr[i:i+k]
11    g.sort(reverse=True)
12    for j in range(len(g)):

```

```
13     arr[i+j]=g[j]
14     i=i+k
15 ▼ for x in arr:
16     print(x,end=' ')
```

	Input	Expected	Got	
✓	7 3 48 541 23 68 13 41 6	541 48 23 68 41 13 6	541 48 23 68 41 13 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

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

Input Format:

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

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

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

Output:

```
1 2 3 4
```

Example Input:

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

Output:

```
1 2 3
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 | n=int(input())
```

```
2 l=[]
3 for i in range(n):
4     x=int(input())
5     l=l+[x]
6 d=[]
7 for i in l:
8     if i not in d:
9         d=d+[i]
10 for i in d:
11     print(i,end=' ')
```

	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 6 | Correct Mark 1.00 out of 1.00

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

Example

$n = 20$

$p = 3$

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

Constraints

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

$1 \leq p \leq 10^9$

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

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

Sample Case 0**Sample Input 0**

10

3

Sample Output 0

5

Explanation 0

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

Sample Case 1**Sample Input 1**

10

5

Sample Output 1

0

Explanation 1

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

Sample Case 2**Sample Input 2**

1

1

Sample Output 2

1

Explanation 2

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

For example:

Input	Result
10	5
3	
10	0
5	

Input	Result
1	1
1	

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 p=int(input())
3 f=[]
4 for i in range(1,n+1):
5     if n%i==0:
6         f.append(i)
7 if p<=len(f):
8     print(f[p-1])
9 else:
10    print(0)

```

	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 7 | Correct Mark 1.00 out of 1.00

Given two arrays of positive integers, for each element in the second array, find the total number of elements in the first array which are *less than or equal* to that element. Store the values determined in an array.

For example, if the first array is $[1, 2, 3]$ and the second array is $[2, 4]$, then there are 2 elements in the first array *less than or equal* to 2. There are 3 elements in the first array which are *less than or equal* to 4. We can store these answers in an array, $\text{answer} = [2, 3]$.

Program Description

The program must return an array of m positive integers, one *for each* $\text{maxes}[i]$ representing the total number of elements $\text{nums}[j]$ satisfying $\text{nums}[j] \leq \text{maxes}[i]$ where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

The program has the following:

$\text{nums}[\text{nums}[0], \dots, \text{nums}[n-1]]$: first array of positive integers
 $\text{maxes}[\text{maxes}[0], \dots, \text{maxes}[n-1]]$: second array of positive integers

Constraints

- $2 \leq n, m \leq 10^5$
- $1 \leq \text{nums}[j] \leq 10^9$, where $0 \leq j < n$.
- $1 \leq \text{maxes}[i] \leq 10^9$, where $0 \leq i < m$.

Input Format For Custom Testing

Input from `stdin` will be processed as follows and passed to the program.

The first line contains an integer n , the number of elements in nums .

The next n lines each contain an integer describing $\text{nums}[j]$ where $0 \leq j < n$.

The next line contains an integer m , the number of elements in maxes .

The next m lines each contain an integer describing $\text{maxes}[i]$ where $0 \leq i < m$.

Sample Case 0**Sample Input 0**

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

Sample Output 0

```
2
4
```

Explanation 0

We are given $n = 4$, $\text{nums} = [1, 4, 2, 4]$, $m = 2$, and $\text{maxes} = [3, 5]$.

1. For $\text{maxes}[0] = 3$, we have 2 elements in nums ($\text{nums}[0] = 1$ and $\text{nums}[2] = 2$) that are $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 5$, we have 4 elements in nums ($\text{nums}[0] = 1$, $\text{nums}[1] = 4$, $\text{nums}[2] = 2$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[1]$.
Thus, the program returns the array [2, 4] as the answer.

Sample Case 1

Sample Input 1

```
5
2
10
5
4
8
4
3
1
7
8
```

Sample Output 1

```
1
0
3
4
```

Explanation 1

We are given, $n = 5$, $\text{nums} = [2, 10, 5, 4, 8]$, $m = 4$, and $\text{maxes} = [3, 1, 7, 8]$.

1. For $\text{maxes}[0] = 3$, we have 1 element in nums ($\text{nums}[0] = 2$) that is $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 1$, there are 0 elements in nums that are $\leq \text{maxes}[1]$.
3. For $\text{maxes}[2] = 7$, we have 3 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[2]$.
4. For $\text{maxes}[3] = 8$, we have 4 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, $\text{nums}[3] = 4$, and $\text{nums}[4] = 8$) that are $\leq \text{maxes}[3]$.

Thus, the program returns the array [1, 0, 3, 4] as the answer.

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 nums=[]
3 for i in range(n):
4     x=int(input())
5     nums=nums+[x]
6 m=int(input())
7 maxes=[]
8 for i in range(m):
9     z=int(input())
10    maxes=maxes+[z]
11 ans=[]
12 for i in maxes:
13     c=0
14     for j in nums:
15         if j<=i:
16             c=c+1
17     ans=ans+[c]
18
19 for i in ans:
20     print(i)
```

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

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

Input Format

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

Output format

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

Example

Input

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

Output:

```
1
```

Input

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

Output

```
0
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 t=int(input())
2 for i in range(t):
```

```
3     n=int(input())
4     l=[]
5     for j in range(n):
6         x=int(input())
7         l=l+[x]
8     z=int(input())
9     f=0
10    for j in range(n):
11        for k in range(n):
12            if l[k]-l[j]==z:
13                f=1
14                break
15    if f==1:
16        break
17    print(f)
```

	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 9 | Correct Mark 1.00 out of 1.00

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

Sample Test Cases

Test Case 1

Input

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

Output

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

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 l=[]
3 for i in range(n):
4     x=int(input())
5     l=l+[x]
6 v=[]
7 for i in l:
8     if i not in v:
9         c=l.count(i)
10        print(i,'occurs',c,'times')
11        v=v+[i]
```

	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 10 | Correct Mark 1.00 out of 1.00

An array is monotonic if it is either **monotone increasing** or **monotone decreasing**.

An array A is monotone increasing if for all $i \leq j$, $A[i] \leq A[j]$. An array A is monotone decreasing if for all $i \leq j$, $A[i] \geq A[j]$.

Write a program if n array is monotonic or not. Print "True" if is monotonic or "False" if it is not. Array can be monotone increasing or decreasing.

Input Format:

First line n-get number of elements

Next n Lines is the array of elements

Output Format:

True ,if array is monotone increasing or decreasing.

otherwise False is printed

Sample Input1

```
4
5
6
7
8
```

Sample Output1

True

Sample Input2

```
4
6
5
4
3
```

Sample Output2

True

Sample Input 3

```
4
6
7
8
7
```

Sample Output3

False

For example:

Input	Result
4	True
6	
5	
4	
3	

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2
```

```
1 l=[ ]
2 for i in range(n):
3     x=int(input())
4     l=l+[x]
5 inc=True
6 dec=True
7 for i in range(1,n):
8     if l[i]<l[i-1]:
9         inc=False
10    if l[i]>l[i-1]:
11        dec=False
12 if inc or dec:
13     print('True')
14 else:
15     print('False')
```

	Input	Expected	Got	
✓	4 6 5 4 3	True	True	✓
✓	4 3 5 7 4	False	False	✓
✓	4 1 6 9 2	False	False	✓
✓	4 9 6 4 2	True	True	✓
✓	3 2 1 4	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 9 September 2025, 12:18 PM

State Finished

Completed on Tuesday, 9 September 2025, 9:07 PM

Time taken 8 hours 48 mins

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

A pangram is a sentence where every letter of the English alphabet appears at least once.

Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Example 1:

Input:

thequickbrownfoxjumpsoverthelazydog

Output:

true

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

Input:

arvijayakumar

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters.

For example:

Test	Result
print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))	true
print(checkPangram('arvijayakumar'))	false

Answer: (penalty regime: 0 %)

Reset answer

```
1 def checkPangram(s):
2     c=0
3     s1='abcdefghijklmnopqrstuvwxyz'
4     for i in s1:
5         if i in s:
6             c+=1
7     if(c==26):
8         return "true"
9     else:
10        return "false"
11
```

	Test	Expected	Got	
✓	print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))	true	true	✓
✓	print(checkPangram('arvijayakumar'))	false	false	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

Given a string `s` containing just the characters `'('`, `')'`, `{`, `}`, `[` and `]`, determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Constraints:

$1 \leq s.length \leq 10^4$

`s` consists of parentheses only `'()'()`.

For example:

Test	Result
<code>print(ValidParenthesis("()"))</code>	true
<code>print(ValidParenthesis("()[]{}"))</code>	true
<code>print(ValidParenthesis("[]"))</code>	false

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 v #}{

2 #( [{ }])

3 v #e-> ([{

4 # t -> {}

5 v def ValidParenthesis(s):
6     e=''
7     t=''
8     stk='[{{'
9     mat = ')-]-{'

10 v     for i in s:
11         if i in stk:
12             e=e+i
13         else:
14             if len(e)==0:
15                 return 'false'
16             t = e[-1]
17             e = e[:-1]
18             if t+i not in mat:
19                 return 'false'
20         if len(e)==0:
21             return 'true'
22         else:
23             return 'false'

```

	Test	Expected	Got	
✓	<code>print(ValidParenthesis("()"))</code>	true	true	✓
✓	<code>print(ValidParenthesis("()[]{}"))</code>	true	true	✓
✓	<code>print(ValidParenthesis("[]"))</code>	false	false	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

Write a Python program to get one string and reverses a string. The input string is given as an array of characters `char[]`.

You may assume all the characters consist of printable ascii characters.

Example 1:**Input:**

hello

Output:

olleh

Example 2:**Input:**

Hannah

Output:

hannaH

Answer: (penalty regime: 0 %)

```
1 n=input()  
2 print(n [::-1])
```

	Input	Expected	Got	
✓	hello	olleh	olleh	✓
✓	Hannah	hannaH	hannaH	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

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

Input Format:

The first line contains S.

Output Format:

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

Boundary Condition:

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

Example Input/Output 1:

Input:

abcd@gmail.com

Output:

com

gmail

abcd

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=input()
2 c=n.split('@')
3 x=c[-1].split('. ',1)
4 for i in range(len(x)-1,-1,-1):
5     print(x[i])
6 print(c[0])
7
8

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

Sample Input 1

thistest123string

123

Sample Output 1

8

Answer: (penalty regime: 0 %)

```
1 x=input()
2 n=input()
3 c=x.find(n)
4 print(c)
```

	Input	Expected	Got	
✓	thistest123string 123	8	8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

The program must accept **N** series of keystrokes as string values as the input. The character **^** represents undo action to clear the last entered keystroke. The program must print the string typed after applying the undo operations as the output. If there are no characters in the string then print **-1** as the output.

Boundary Condition(s):

$1 \leq N \leq 100$

$1 \leq \text{Length of each string} \leq 100$

Input Format:

The first line contains the integer **N**.

The next **N** lines contain a string on each line.

Output Format:

The first **N** lines contain the string after applying the undo operations.

Example Input/Output 1:

Input:

```
3
Hey ^ goooo^^glee^
lucke^y ^charr^ms
ora^^nge^^^^
```

Output:

```
Hey google
luckycharms
-1
```

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 for i in range(n):
3     x=input()
4     r=''
5     for j in x:
6         if j=='^':
7             if r!='':
8                 r=r[:-1]
9         else:
10            r=r+j
11     if r=='':
12         print(-1)
13     else:
14         print(r)
```

	Input	Expected	Got	
✓	3 Hey ^ goooo^^glee^ lucke^y ^charr^ms ora^^nge^^^^	Hey google luckycharms -1	Hey google luckycharms -1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

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 %)

```

1 x=input()
2 key=['break','case','continue','default','defer','else','for','func','goto','if','map','range','return','struct',''
3 if x in key:
4     print(x,'is a keyword')
5 else:
6     print(x,'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.

Question 8 | Correct Mark 1.00 out of 1.00

Given a **non-empty** string **s** and an abbreviation **abbr**, return whether the string matches with the given abbreviation.

A string such as "word" contains only the following valid abbreviations:

```
[“word”, “1ord”, “w1rd”, “wo1d”, “wor1”, “2rd”, “w2d”, “wo2”, “1o1d”, “1or1”, “w1r1”, “1o2”, “2r1”, “3d”, “w3”, “4”]
```

Notice that only the above abbreviations are valid abbreviations of the string "word". Any other string is not a valid abbreviation of "word".

Note:

Assume **s** contains only lowercase letters and **abbr** contains only lowercase letters and digits.

Example 1:**Input**

internationalization

i12iz4n

Output

true

Explanation

Given **s** = "internationalization", **abbr** = "i12iz4n":

Return true.

Example 2:**Input**

apple

a2e

Output

false

Explanation

Given **s** = "apple", **abbr** = "a2e":

Return false.

Answer: (penalty regime: 0 %)

```

1 x=input()
2 s=x.strip()
3 a=input()
4 abr=a.strip()
5 exp=''
6 n=0
7 for i in abr:
8     if i.isdigit():
9         n=n*10+int(i)
10    else:
11        if n>0:
12            exp=exp+'#'*n
13            n=0
14        exp=exp+i
15    if n>0:
16        exp=exp+'#'*n
17    if len(exp)==len(s):
18        v=True
19    for i in range(len(s)):
20        if exp[i]!='#' and exp[i]!=s[i]:
21            v=False
22            break
23    else:
24        v=False
25    if v:
26        print('true')
27    else:
28        print('false')
```

	Input	Expected	Got	
✓	internationalization i12iz4n	true	true	✓
✓	apple a2e	false	false	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

Assume that the given string has enough memory.

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

Sample Input 1

a2b4c6

Sample Output 1

aabbbbccccc

Answer: (penalty regime: 0 %)

```

1 x=input()
2 i=0
3 while i<len(x):
4     if x[i].isalpha():
5         j=i+1
6         n=0
7         while j<len(x) and x[j].isdigit():
8             n=n*10+int(x[j])
9             j=j+1
10        if n==0:
11            print(x[i],end=' ')
12        else:
13            print(x[i]*n,end=' ')
14        i=j
15    else:
16        i=i+1

```

	Input	Expected	Got	
✓	a2b4c6	aabbbbccccc	aabbbbccccc	✓
✓	a12b3d4	aaaaaaaaaaabbdd	aaaaaaaaaaabbdd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

Note: For the purpose of this problem, we define empty string as valid palindrome.

Example 1:

Input:
A man, a plan, a canal: Panama

Output:
1

Example 2:

Input:
race a car

Output:
0

Constraints:

- s consists only of printable ASCII characters.

Answer: (penalty regime: 0 %)

```

1 x=input()
2 f=''
3 for i in x:
4     if i.isalnum():
5         f=f+i.lower()
6 r=''
7 k=len(f)-1
8 for i in range(k,-1,-1):
9     r=r+f[i]
10 if f==r:
11     print(1)
12 else:
13     print(0)

```

	Input	Expected	Got	
✓	A man, a plan, a canal: Panama	1	1	✓
✓	race a car	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Sunday, 31 August 2025, 9:26 PM

State Finished

Completed on Sunday, 31 August 2025, 9:42 PM

Time taken 16 mins 39 secs

Marks 5.00/5.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Complete a Recursive Function to find if a given number N can be expressed as a sum of two prime numbers.

Note: YOU MUST OPTIMIZE the logic to find whether a number is prime or not, as very large prime numbers are provided as input. If the logic is not optimized your program will NOT get executed within the given time limit.

Input Format:

First line contains number N.

Output Format:

Return either yes or no.

Boundary Conditions / Constraints:

$3 \leq N \leq 10^9$

Example Input/Output 1:**Input:**

20

Output:

yes

Input:

23

Output:

no

Explanation:

20 can be expressed as 17+3

23 cannot be expressed as sum of two primes

For example:

Test	Result
<code>print(checkPrimeSum(20))</code>	yes
<code>print(checkPrimeSum(23))</code>	no

Answer: (penalty regime: 0 %)

Reset answer

```

1 def isPrime(num, divisor=2):
2     if num<2:
3         return False
4     if divisor * divisor > num:
5         return True
6     if num%divisor ==0:
7         return False
8     return isPrime(num, divisor+1)
9 def checkPrimeSum(n, i=2):
10    if i > n//2:
11        return "no"
12    if isPrime(i) and isPrime(n-i):
13        return "yes"
14    return checkPrimeSum(n, i+1)

```

	Test	Expected	Got	
✓	print(checkPrimeSum(20))	yes	yes	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

Euclid was a Greek mathematician who lived approximately 2,300 years ago. His algorithm for computing the greatest common divisor of two positive integers, a and b , is both efficient and recursive. It is outlined below:

```
If b is 0 then
    return a
Else
    Set c equal to the remainder when a is divided by b
    Return the greatest common divisor of b and c
```

Write a Recursive function that implements Euclid's algorithm and uses it to determine the greatest common divisor of two integers entered by the user. Test your program with some very large integers. The result will be computed quickly, even for huge numbers consisting of hundreds of digits, because Euclid's algorithm is extremely efficient.

For example:

Test	Result
print(gcd(8, 12))	4
print(gcd(720, 1000))	40

Answer: (penalty regime: 0 %)

Reset answer

```
1 def gcd(a,b):
2     if b==0:
3         return a
4     else:
5         return gcd(b,a%b)
```

	Test	Expected	Got	
✓	print(gcd(8, 12))	4	4	✓
✓	print(gcd(720, 1000))	40	40	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

The notion of a palindrome was introduced previously. In this exercise you will write a recursive function that determines whether or not a string is a palindrome. The empty string is a palindrome, as is any string containing only one character. Any longer string is a palindrome if its first and last characters match, and if the string formed by removing the first and last characters is also a palindrome.

Write a program that reads a string from the user and uses your recursive function to determine whether or not it is a palindrome. Then your program should display an appropriate message for the user.

Sample Input

malayalam

Sample Output

That was a palindrome!

Sample Input

madan

Sample Output

That is not a palindrome.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 def isPalindrome(s):
2     # Base case: The empty string is a palindrome. So is a string containing only 1 character.
3     if len(s) <= 1:
4         return True
5     # Recursive case: The string is a palindrome only if the first and last characters match, and
6     # the rest of the string is a palindrome
7     if s[0] == s[-1]:
8         return isPalindrome(s[1:-1])
9     else:
10        return False
11
12
13 line=input()
14 # Check whether or not a string entered by the user is a palindrome
15 # Read the string from the user
16
17 # Check its status and display the result
18 if isPalindrome(line):
19     print("That was a palindrome!")
20 else:
21     print("That is not a palindrome.")
22
23
24
25

```

	Input	Expected	Got	
✓	malayalam	That was a palindrome!	That was a palindrome!	✓
✓	madan	That is not a palindrome.	That is not a palindrome.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

Given an integer number and you have to count the digits using recursion using Python program. In this program, you will be reading an integer number and counting the total digits, using a function `countDigits()` which will take a number as an argument and return the count after recursion process.

Input Format: The first and only line of the input contains a single integer `n`

Output Format: Output a single line denoting the number of digits in `n`.

For example:

Test	Result
<code>print(countDigits(800))</code>	3

Answer: (penalty regime: 0 %)

Reset answer

```
1 def countDigits(n):
2     if n<10:
3         return 1
4     else:
5         return 1+ countDigits(n//10)
6
```

	Test	Expected	Got	
✓	<code>print(countDigits(12345))</code>	5	5	✓
✓	<code>print(countDigits(800))</code>	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Complete the recursive function to return Binary Equivalent of an Integer using Recursion.

Sample Test Cases

Test Case 1

Input

10

Output

1010

Test Case 2

Input

257

Output

100000001

For example:

Test	Result
print(binayNumber(10))	1010
print(binayNumber(257))	100000001

Answer: (penalty regime: 0 %)

Reset answer

```
1 def binayNumber(n):
2     if n<2:
3         return str(n)
4     else:
5         return binayNumber (n//2) + str(n%2)
6
```

	Test	Expected	Got	
✓	print(binayNumber(10))	1010	1010	✓
✓	print(binayNumber(257))	100000001	100000001	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Monday, 25 August 2025, 9:14 PM

State Finished

Completed on Monday, 25 August 2025, 10:50 PM

Time taken 1 hour 36 mins

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

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

Some examples are as below.

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

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

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

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

For example:

Input	Result
292	1
1015	2
108	3
22	0

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 dc={}
3 sn=str(n)
4 for digit in sn:
5     dc[digit]=dc.get(digit,0)+1
6 c=0
7 for digit,count in dc.items():
8     if count==1:
9         c+=1
10 print(c)

```

	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 2 | Correct Mark 1.00 out of 1.00

Given a positive integer a, find the smallest positive integer b whose multiplication of each digit equals to a.

If there is no answer or the answer is not fit in 32-bit signed integer, then return 0.

Example 1

Input:

48

Output:

68

Example 2

Input:

15

Output:

35

For example:

Input	Result
48	68
15	35

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 if a<1:
3     print(0)
4 elif a==1:
5     print(1)
6 else:
7     r=0
8     temp=a
9 for i in range(9,1,-1):
10    while temp%i==0:
11        r=r*10+i
12        temp=temp//i
13    if temp!=1:
14        print(0)
15    else:
16        re=0
17        o=r
18    while o>0:
19        re=re*10+(o%10)
20        o=o//10
21    if re>2147483647:
22        print(0)
23

```

```
22  
23     print(0)  
else:  
24         print(re)
```

	Input	Expected	Got	
✓	48	68	68	✓
✓	15	35	35	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

Let's print a chessboard!

Write a program that takes input:

Integer N(represents the rows and columns of a chessboard) and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

W

Output:

WB

BW

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 s=input()
3 if s=="W":
4     alt='B'
5 else:
6     alt='W'
7 for i in range(n):
8     row=''
9 for j in range(n):
10    if (i+j)%2==0:
11        row=row+s
12    else:
13        row=row+alt
14 print(row)

```

	Input	Expected	Got	
✓	2 W	WB BW	WB BW	✓
✓	3 B	BWB WBW BWB	BWB WBW BWB	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

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

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

For example:

Input	Result
10	16
20	25

Answer: (penalty regime: 0 %)

```
1 import math
2 n=int(input())
3 root=math.sqrt(n)
4 ns=(root+1)**2
5 print(ns)
```

	Input	Expected	Got	
✓	10	16	16	✓
✓	20	25	25	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Write a program 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: $2 \leq \text{starting number of the range} \leq \text{ending number of the range} \leq 7919$

Example1: If the starting and ending number of 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	8
20	
700	3
725	

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=0
4 for num in range(a,b+1):
5     if num>1:
6         is_prime=True
7         for i in range(2,int(num**0.5)+1):
8             if num%i==0:
9                 is_prime=False
10            break
11        if is_prime:
12            c+=1
13 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 6 | Correct Mark 1.00 out of 1.00

You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction.

- The first kangaroo starts at position x_1 and moves at a speed v_1 meters per jump.
- The second kangaroo starts at position x_2 and moves at a speed of v_2 meters per jump and $x_2 > x_1$
- You have to figure out to get both kangaroos at the same position at the same time as part of the show before k jumps. If it is possible, return YES, otherwise return NO.

Input Format:

x_1 -position of kangaroo1

v_1 -Speed of kangaroo1

x_2 -position of kangaroo2

v_2 -Speed of kangaroo2

k -jumps

Output Format:

Both kangaroos are at the same position within k jumps, YES, otherwise NO.

For example:

Input	Result
0	YES
3	
4	
2	
6	

Answer: (penalty regime: 0 %)

```

1 x1=int(input())
2 v1=int(input())
3 x2=int(input())
4 v2=int(input())
5 k=int(input())
6 f=False
7 for j in range(1,k+1):
8     p1=x1+v1*j
9     p2=x2+v2*j
10    if p1==p2:
11        f=True
12        break
13 if f:
14     print("YES")
15 else:
16     print("NO")

```

	Input	Expected	Got	
✓	0 3 4 2 6	YES	YES	✓
✓	0 3 2 4 8	NO	NO	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

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

Input Format:

Single Integer Input from stdin.

Output Format:

Yes or No.

Example Input:

175

Output:

Yes

Explanation

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

Example Input:

123

Output:

No

For example:

Input	Result
175	Yes
123	No

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 s=0
3 num=n
4 x=n
5 c=0
6 while num!=0:
7     r=num%10
8     num=num//10
9     c=c+1
10 while c>0:
11     k=n%10
12     n=n//10
13     s=s+k**c
14     c=c-1
15 if s==x:
16     print ('Yes')
17 else:
18     print('No')
```

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

	Input	Expected	Got	
✓	90	No	No	✓
✓	518	Yes	Yes	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

Write a program that given an integer 'n', prints the number of integers that are less than or equal to 'n' and co-prime to 'n'

Two integers a and b are said to be relatively prime or co-prime if the only positive integer that evenly divides both of them is 1. That is, the only common positive factor of the two numbers is 1. This is equivalent to their greatest common divisor being 1.

Input Format:

One line containing the value of 'n', where $1 \leq n \leq 10,000$

Output Format:

One line containing the number of integers that are co-prime to n and less than or equal to 'n'

Sample Test Cases

Test Case 1

Input

10

Output

4

Test Case 2

Input

23

Output

22

Test Case 3

Input

11

Output

10

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 c=0
3 i=1
4 while i<=n:
5     a=i
6     b=n
7     while b!=0:
8         temp=b
9         b=a%b
10        a=temp
11    if a==1:
12        c=c+1
13    i=i+1
14 print(c)
```

	Input	Expected	Got	
✓	10	4	4	✓
✓	23	22	22	✓
✓	11	10	10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Question 9 | Correct Mark 1.00 out of 1.00

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 it is an automorphic number display "Automorphic" else display "Not Automorphic".

Input Format:

Take a Integer from Keyboard

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

Answer: (penalty regime: 0 %)

```

1 num=int(input())
2 n=num
3 c=0
4 sq=num**2
5 while num!=0:
6     r=num%10
7     num=num//10
8     c=c+1
9     s=sq%(10**c)
10 if s==n:
11     print("Automorphic")
12 else:
13     print("Not Automorphic")

```

	Input	Expected	Got	
✓	5	Automorphic	Automorphic	✓
✓	625	Automorphic	Automorphic	✓
✓	7	Not Automorphic	Not Automorphic	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

Write python program to print the following pattern based on input size.

Input:

3

Output:

```
1
2 3
4 5 6
```

For example:

Input	Result
4	1 2 3 4 5 6 7 8 9 10

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 num=1
3 for i in range (1,n+1):
4     print(" "* (n-i),end="")
5     for j in range (i):
6         print(num,end=" ")
7         num+=1
8     print()
```

	Input	Expected	Got	
✓	3	1 2 3 4 5 6	1 2 3 4 5 6	✓
✓	4	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Saturday, 16 August 2025, 9:45 PM

State Finished

Completed on Monday, 18 August 2025, 9:24 PM

Time taken 1 day 23 hours

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Write a Python program that accepts three parameters. The first parameter is an integer. The second is one of the following mathematical operators: +, -, /, or *. The third parameter will also be an integer.

The function should perform a calculation and return the results. For example, if the function is passed 6 and 4, it should return 24.

Sample Input Format:

```
11
+
14
```

Sample Output Format:

```
25
```

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 v=input()
3 m=int(input())
4 if(v=='+'):
5     print(n+m)
6 elif(v=='-'):
7     print(n-m)
8 elif(v=='*'):
9     print(n*m)
10 elif(v=='/'):
11     print(n/m)
12
13
```

	Input	Expected	Got	
✓	11 + 14	25	25	✓
✓	45 - 50	-5	-5	✓
✓	12 * 100	1200	1200	✓
✓	18 / 2	9.0	9.0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

IN / OUT

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

Input Format:

Input consists of 2 integers.

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

Output Format:

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

Sample Input and Output:

Input

8

3

Output

OUT

For example:

Input	Result
8	OUT
3	

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 m=int(input())
3 v=n/2
4 if(v>=m):
5     print("OUT")
6 else:
7     print("IN")

```

	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 3 | Correct Mark 1.00 out of 1.00

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

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

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

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

For example:

Input	Result
100.00	120.00
500	1035.00

Answer: (penalty regime: 0 %)

```

1 n=float(input())
2 v=0.00
3 if(100<=n<=199):
4     v=1.20
5 elif(200<=n<400):
6     v=1.50
7 elif(400<=n<600):
8     v=1.80
9 elif(n>=600):
10    v=2.00
11 tb=n*v
12 t=tb
13 if(tb>400):
14     tb=tb+tb*0.15
15 if(tb<100):
16     tb=100.00
17 print("%.2f"%tb)

```

	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 4 | Correct Mark 1.00 out of 1.00

Given an integer N, check whether N the given number can be made a perfect square after adding 1 to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

For example:

Input	Result
24	Yes

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 m=n+1
3 x=m**0.5
4 if(m%x==0):
5     print("Yes")
6 else:
7     print("No")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Write a program to determine the type of berth when the seat / berth number in the train is given.

**Input Format:**

Input consists of a single integer. Assume that the range of input is between 1 and 72.

Output Format:

Output consists of a single string. [Upper or Middle or Lower or Side Lower or Side Upper]

Sample Input 1:

9

Sample Output 1:

Lower Berth

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 m=n%8
3 if m in [1,4]:
4     print("Lower Berth")
5 elif m in [2,5]:
6     print("Middle Berth")
7 elif m in [3,6]:
8     print("Upper Berth")
9 elif m ==7:
10    print("Side Lower Berth")
11 elif m==0:
12    print("Side Upper Berth")

```

	Input	Expected	Got	
✓	9	Lower Berth	Lower Berth	✓
✓	72	Side Upper Berth	Side Upper Berth	✓
✓	20	Lower Berth	Lower Berth	✓
✓	34	Middle Berth	Middle Berth	✓
✓	39	Side Lower Berth	Side Lower Berth	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

Write a program that accepts 5 inputs and returns the count of how many of those 5 are odd.

For example,

If the five inputs are 12, 17, 19, 14, and 115, there are three odd numbers 17, 19 and 115. So, the program must return 3.

Similarly,

If the five inputs are 15, 0, -12, 19, and 28, there are two odd numbers 15 and 19. So, the program must return 2.

Observe that zero is considered an even number.

For example:

Input	Result
12	3
17	
19	
14	
115	
15	2
0	
-12	
19	
28	

Answer: (penalty regime: 0 %)

```

1 a=[]
2 c=0
3 for i in range(5):
4     a=int(input())
5 #for i in range(0,4):
6     if(a%2!=0):
7         c+=1
8 print(c)

```

	Input	Expected	Got	
✓	12	3	3	✓
	17			
	19			
	14			
	115			

	Input	Expected	Got	
✓	15	2	2	✓
	0			
	-12			
	19			
	28			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

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

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

Sample Input 1

60

60

60

Sample Output 1

That's a equilateral triangle

Sample Input 2

40

40

80

Sample Output 2

That's a isosceles triangle

Sample Input 3

50

60

70

Sample Output 3

That's a scalene triangle

For example:

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

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=int(input())
4 if(a==b==c):
5     print("That's a equilateral triangle")
6 elif(a!=b!=c):
7     print("That's a scalene triangle")
8 else:
9     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	✓
✓	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 8 | Correct Mark 1.00 out of 1.00

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

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

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

Sample Input 1

1900

Sample Output 1

1900 is not a leap year.

Sample Input 2

2000

Sample Output 2

2000 is a leap year.

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 if n%4==0 and n%100!=0 or n%400==0:
3     print("%d is a leap year."%n)
4 else:
5     print("%d is not a leap year."%n)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

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

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

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

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

Answer: (penalty regime: 0 %)

```
1 m=int(input())
2 a=["Dragon", "Snake", "Horse", "Sheep", "Monkey", "Rooster", "Dog", "Pig", "Rat", "Ox", "Tiger", "Hare"]
3 z=a[(m-2000)%12]
4 print(f"{m} is the year of the {z}.")
```

	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 10 | Correct Mark 1.00 out of 1.00

A certain type of steel is used to test and give grade according to the following conditions.

1. Hardness of the steel must be greater than 50
2. Carbon content of the steel must be less than 0.7
3. Tensile strength must be greater than 5600

The grades awarded are as follows:

- Grade is 10 if all three conditions are met
- Grade is 9 if conditions (1) and (2) are met
- Grade is 8 if conditions (2) and (3) are met
- Grade is 7 if conditions (1) and (3) are met
- Grade is 6 if only one condition is met
- Grade is 5 if none of the three conditions are met

Write a program to display the grade of the steel, based on the values of hardness, carbon content and tensile strength of the steel, given by the user.

Input

53

0.6

5602

Output:

10

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=float(input())
3 c=int(input())
4 if(a>50 and b<0.7 and c>=5600):
5     print("10")
6 elif(a>50 and b<0.7):
7     print("8")
8 elif(a>50 and c>=5600):
9     print("7")
10 elif(a>50 or b<0.7 or c>=5600):
11     print("6")
12 else:
13     print("5")
```

	Input	Expected	Got	
✓	53 0.6 5602	10	10	✓
✓	45 0 4500	6	6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Friday, 1 August 2025, 3:13 PM

State Finished

Completed on Saturday, 2 August 2025, 11:21 AM

Time taken 20 hours 7 mins

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

Sample Input

10

20

Sample Output

Your total refund will be \$6.00.

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 m=int(input())
3 s=n*0.10
4 v=m*0.25
5 y=s+v
6 print(f"Your total refund will be ${%.2f}"%y)
7

```

	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 2 | Correct Mark 1.00 out of 1.00

In a Logistic the Parcels to be delivered in 4 locations (1st location 20%, 2nd location 40%, 3rd location 30% and 4th location 10%). write a python code to find the total no. of parcels after the delivery in 2 locations . use a format() to print the no of parcels delivered in each location

Input:

250

output:

Total Parcels is 250

1st Location 50 parcels

2nd Location 100 parcels

3rd Location 75 parcels

4th Location 25 parcels

For example:

Input	Result
250	Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels
400	Total Parcels is 400 1st Location 80 parcels 2nd Location 160 parcels 3rd Location 120 parcels 4th Location 40 parcels

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 a=n*20/100
3 b=n*40/100
4 c=n*30/100
5 d=n*10/100
6 print("Total Parcels is %d"%n)
7 print("1st Location %d parcels"%a)
8 print("2nd Location %d parcels"%b)
9 print("3rd Location %d parcels"%c)
10 print("4th Location %d parcels"%d)

```

	Input	Expected	Got	
✓	250	Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels	Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

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

Hint:

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

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

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

Sample Input:

450

Sample Output:

weekdays 10.38

weekend 0.38

For example:

Input	Result
450	weekdays 10.38 weekend 0.38

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 m=abs((a-500)/130)
3 v=10+m
4 print(f"weekdays %.2f"%v)
5 print(f"weekend %.2f"%m)
6
7
```

	Input	Expected	Got	
✓	450	weekdays 10.38 weekend 0.38	weekdays 10.38 weekend 0.38	✓

	Input	Expected	Got	
✓	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.

Question 4 | Correct Mark 1.00 out of 1.00

Write a program to find whether the given input number is Even.

If the given number is even, the function should return 2 else it should return 1.

Note: The number passed to the program can either be negative, positive or zero. Zero should be treated as Even.

For example:

Input	Result
100	2
1001	1

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 if(n%2==0):
3     print("2")
4 else:
5     print("1")
```

	Input	Expected	Got	
✓	100	2	2	✓
✓	1001	1	1	✓
✓	0	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

In department 54% are boys and 46% are girls and 8% are hostel (boys/girls). write a python code to print total no of boys, girls and hostel students in the specific format using modulo operator.

input: 1500

output: Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120

For example:

Input	Result
1500	Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120
1200	Total Students : 1200, Boys : 648, Girls : 552, Hostel : 96

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 a=int(n*0.54)
3 b=int(n*0.46)
4 c=int(n*0.08)
5 print("Total Students : %d, Boys : %d, Girls : %d, Hostel : %d"%(n,a,b,c))

```

	Input	Expected	Got	
✓	1500	Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120	Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

Write a program that 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 %)

```
1 n=int(input())
2 m=abs(n)
3 a=m%10
4 print(a)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

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

Input Format:

The first line contains the Rs X

The second line contains Rs Y

The third line contains Rs Z

Sample Input:

10000

250

15000

Sample Output:

46.34 is the gain percent.

For example:

Input	Result
45500	30.43 is the gain percent.
500	
60000	

Answer: (penalty regime: 0 %)

```

1 x=int(input())
2 y=int(input())
3 z=int(input())
4 l=x+y
5 m=z-l;
6 s=m*100/l
7
8 print(f"%2f is the gain percent."%s)
9

```

	Input	Expected	Got	
✓	10000 250 15000	46.34 is the gain percent.	46.34 is the gain percent.	✓

	Input	Expected	Got	
✓	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 8 | Correct Mark 1.00 out of 1.00

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

Sample Input:

10

10.9

Sample Output:

10,<class 'int'>

10.9,<class 'float'>

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=input()
2 m=input()
3 n=int(n)
4 m=float(m)
5 print(n,type(n),sep=',')
6 print("%0.1f"%m,type(m),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 56.2	55000,<class 'int'> 56.2,<class 'float'>	55000,<class 'int'> 56.2,<class 'float'>	✓
✓	2541 2541.679	2541,<class 'int'> 2541.7,<class 'float'>	2541,<class 'int'> 2541.7,<class 'float'>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

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

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

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

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

For example:

Input	Result
197	9
-197	9
5	-1

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 m=abs(n)
3 if(m<10):
4     print ("-1")
5 else:
6     print((m//10)%10)
7
8

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

In a Lab 36% are Dell and 34% Lennovo and 28% are Acer and 2% are Samsung. write a python code to print total systems and brand wise count in the specific format using sep operator.

input: 150

output: Total System:150

Dell:54

Lennovo:51

Acer:42

Samsung:3

For example:

Input	Result
150	Total System:150 Dell:54 Lennovo:51 Acer:42 Samsung:3
250	Total System:250 Dell:90 Lennovo:85 Acer:70 Samsung:5

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 a=n*0.36
3 b=n*0.34
4 c=n*0.28
5 d=n*0.02
6 print("Total System:%d"%n)
7 print("Dell:%d"%a)
8 print("Lennovo:%d"%b)
9 print("Acer:%d"%c)
10 print("Samsung:%d"%d)
11
12

```

	Input	Expected	Got	
✓	150	Total System:150 Dell:54 Lennovo:51 Acer:42 Samsung:3	Total System:150 Dell:54 Lennovo:51 Acer:42 Samsung:3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.