

Week-1, Practice Assignment (theory)

Week-1, Practice Assignment (theory)

Problem-1

 Question

 Answer

 Solution

Problem-2

 Question

 Answer

 Solution

Problem-3

 Question

 Answer

 Solution

Problem-4

 Question

 Answer

 Solution

Problem-5

 Question

 Answer

 Solution

Problem-6

 Question

 Answer

 Solution

Problem-7

 Question

 Answer

 Solution

Problem-8

 Question

 Answer

 Solution

Problem-9

 Question

 Answer

 Solution

Problem-1

Question

What will be the output of the following statement?

```
1 | print('15 % 3 * 2 + 2')
```

- (a) 12
- (b) 2
- (c) 15 % 3 * 2 + 2
- (d) '15 % 3 * 2 + 2'

Answer

- (c)

Solution

The `print()` function considers a mathematical expression as string when it is surrounded by quotes. The expression inside the quotes will be displayed as it is on the console. Hence, option (c) is the correct answer.

Problem-2

Question

What will be the output of the following statement?

```
1 | print("a + 'b' + c + '' + d")
```

- (a) a + 'b' + c + '' + d
- (b) "a + 'b' + c + '' + d"
- (c) a + b + c + '' + d
- (d) a + ''b'' + c + '' + d

Answer

(a)

Solution

The `print()` function assumes any expression inside quotes as string. Python does not allow double quotes inside of double quotes or single quotes inside of single quotes. But it does allow to print single quotes inside double quoted string and vice versa. Hence the value `a + 'b' + c + '' + d` is displayed as it is. The starting and ending quotes are not displayed. Hence, (a) is True, while (b) is False. The option (c) and (d) are incorrect as the actual input has single quotes surrounding the letter `b`.

Problem-3

Question

A snippet of code gives the following output when executed:

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

There is exactly one space between any two consecutive integers. Which of the following options correspond to the correct code snippet? It is a Multiple Select Question (MSQ).

(a)

```
1 | print(1 2 3 4 5)
```

(b)

```
1 | print('1 2 3 4 5')
```

(c)

```
1 | print(1)
2 | print(2)
3 | print(3)
4 | print(4)
5 | print(5)
```

(d)

```
1 | print(1, 2, 3, 4, 5)
```

Answer

(b), (d)

Solution

Option (b) is correct, `print()` function displays the string passed to the console as it is. When there are multiple values inside the `print()` function separated by commas, those values are concatenated together using `space` and the output is displayed. Hence, option (d) is also correct.

Problem-4

Question

What will be datatype of following expressions?

```
1 13 % 5 // 2 * 30 ** 5
2 "@Python"
3 20 ** 10 / 2 + 25 - 70
4 False
5 20 * 100.0 // 11 % 5
```

- (a) float
- (b) int
- (c) str
- (d) bool

Answer

1- (b); 2- (c); 3- (a); 4- (d); 5-(a)

Solution

Line-1: Option (b). The expression includes `%`, `//`, `*` and `**` operators. The output from these operators is an integer when all operands are integers.

Line-2: Option (c). It is a string. So, the type will be `str`.

Line-3: Option (a). The operator `/` is a `float` operation, this means it will give output as `float` type. Since, the expression includes the `/` operator, the output of the expression becomes `float`.

Line-4: Option (d). This is a `bool` value.

Line-5: Option (a). The expression involves a `float` value `100.0`. Hence, the entire expression is evaluated as `float`.

Problem-5

Question

How does the Python interpreter parenthesize the following expression?

```
1 | 0 ** 1 ** 2 ** 3 ** 2
```

- (a) `((0 ** 1) ** 2) ** 3) ** 2`
- (b) `(0 ** (((1 ** 2) ** 3) ** 2))`
- (c) `0 ** (1 ** (2 ** (3 ** 2)))`
- (d) `(0 ** ((1 ** 2) ** (3 ** 2)))`

Answer

(c)

Solution

The power operator `**` has right to left associativity. Hence, option (c) is the correct way of computation.

Problem-6

Question

How does the Python interpreter parenthesize the following expression?

```
1 | 8.2 * 10 ** 4 + 19
```

- (a) `((8.2 * 10) ** 4) + 19`
- (b) `8.2 * (10 ** (4 + 19))`
- (c) `(8.2 * (10 ** 4)) + 19`
- (d) `(8.2 * 10) ** (4 + 19)`

Answer

(c)

Solution

The order of precedence is `** > * > +`. Hence, option (c) is the correct way of parsing for computation.

Problem-7

Question

How does the Python interpreter parenthesize the following expression?

```
1 | not False or True and False
```

- (a) not (False or (True and False))
- (b) (not False) or (True and False)
- (c) not ((False or True) and False)
- (d) ((not False) or True) and False

Answer

(b)

Solution

Logical operators have precedence `not` > `and` > `or`. Therefore, option (b) is correct.

Problem-8

Question

What will be the output of the following statement?

```
1 | not 0 and 10 // 5 == 2
```

(a) True

(b) False

Answer

(a)

Solution

Option (a) is right. The precedence of operators are `//` > `==` > `not` > `and`. Therefore, the computation follows construct: `(not 0) and ((10 // 5) == 2)`.

Problem-9

Question

Given a string variable `word` that stores some word in the English language, we wish to create a new string with just two characters:

- The first character in the new string is the first letter in `word`.
- The second character in the new string the last letter in `word`.

Assume that `word` has at least three characters. Which of the following lines can be used to create the new string? It is a Multiple Select Question (MSQ).

- (a) `word[0] + word[1]`
- (b) `word[0] + word[-1]`
- (c) `word[0] + word[len(word) - 1]`
- (d) `word[-1] + word[len(word)]`

Answer

(b), (c)

Solution

Option (b) and (c) are correct. The indexing of characters from left to right in the variable `word` starts at index `0` and ends at `len(word) - 1`. The first and last letter can be accessed using the `word[0]` and `string[len(word) - 1]` respectively. Alternatively, the indexing starts at `-1` and ends at `-len(word)` when we move from right to left in the string. Using negative indexing, the first and last letter can be retrieved by `word[-len(word)]` and `word[-1]` respectively.

Week-1, Graded Assignment (theory)

(15 marks)

Week-1, Graded Assignment (theory)

Problem-1

 Question

 Answer

 Solution

Problem-2

 Question

 Answer

 Solution

Problem-3

 Question

 Answer

 Solution

Problem-4

 Question

 Answer

 Solution

Problem-5

 Question

 Answer

 Solution

Problem-6

 Question

 Answer

 Solution

Problem-7

 Question

 Answer

 Solution

Problem-8

 Question

 Answer

 Solution

Problem-1

(1 mark)

Question

What is the type of the following expression?

```
1 | 1 + 4 / 2
```

- (a) `int`
- (b) `float`
- (c) `str`
- (d) `bool`

Answer

- (b)

Solution

In the expression, the order of precedence is `/` > `+`. In Python The `/` operator always returns a `float` value and `+` operation with `float` and `int` value also returns a `float` value. So, the above expression will return a `float` value. Hence, Option (b) is correct.

Problem-2

(1 mark)

Question

What is the type of the following expression?

```
1 | 1 > 0 and -1 < 0 and 1 == 1
```

- (a) str
- (b) bool
- (c) True
- (d) False

Answer

(b)

Solution

In the above expression, multiple conditions are connected using a logical operator `and`. Each condition evaluates to `True`. So, the expression always returns `True` which is of a `bool` type. Hence, Option (b) is correct.

Problem-3

(2 marks)

Question

How does the Python interpreter parenthesize the following expression?

```
1 | 1 + 3 / 4 ** 2 * 0
```

- (a) `1 + (((3 / 4) ** 2) * 0)`
- (b) `1 + ((3 / (4 ** 2)) * 0)`
- (c) `(1 + 3 / 4) ** (2 * 0)`
- (d) All of the above

Answer

(b)

Solution

In the expression, the order of precedence is `** > / , * > +`. According to the precedence order, `(4 ** 2)` will be evaluated first. Operators `/` and `*` have the same precedence and left to right associativity. So, `(3 / (4 ** 2))` will be evaluated first. Thereafter `((3 / (4 ** 2)) * 0)` will be evaluated and finally `1 + ((3 / (4 ** 2)) * 0)` will be evaluated. Hence, Option (b) is correct.

Problem-4

(2 marks)

Question

Convert the following mathematical statement into a Python expression. It is a Multiple Select Question (MSQ).

$$10^3 + 9^3 = 12^3 + 1^3 = 1729$$

- (a) `10^3 + 9^3 == 12^3 + 1^3 == 1729`
- (b) `10 ** 3 + 9 ** 3 = 12 ** 3 + 1 ** 3 = 1729`
- (c) `10 ** 3 + 9 ** 3 == 12 ** 3 + 1 ** 3 == 1729`
- (d) `(10 ** 3) + (9 ** 3) == (12 ** 3) + (1 ** 3) == 1729`

Answer

(c), (d)

Solution

- In Python,
 - $10^3 + 9^3$ will be equivalent to `10 ** 3 + 9 ** 3` or `(10 ** 3) + (9 ** 3)`
 - $12^3 + 1^3$ will be equivalent to `12 ** 3 + 1 ** 3` or `(12 ** 3) + (1 ** 3)`
 - Equal sign used in mathematics `=` is equivalent to `==` operator.

Hence, option (c) and (d) both are correct.

Problem-5

(2 marks)

Question

`E_1` and `E_2` are boolean expressions. Consider the following expression.

```
1 | E_3 = not (E_1 or E_2)
2 | E_4 = (not E_1) and (not E_2)
3 | print(E_3 == E_4)
```

What can you say about the value of the expression given above?

- (a) It is `True` if and only if both `E_1` and `E_2` have the same value.
- (b) It is `False` if and only if both `E_1` and `E_2` have the same value.
- (c) It is always `True`.
- (d) It is always `False`.

Answer

(c)

Solution

<code>E_1</code>	<code>E_2</code>	<code>not (E_1 or E_2)</code>	<code>(not (E_1) and not(E_2))</code>	<code>not (E_1 or E_2) == ((not E_1) and (not E_2))</code>
False	False	True	True	True
False	True	False	False	True
True	False	False	False	True
True	True	False	False	True

So, we can see that for all possible values of `E_1` and `E_2`, expression returns `True`. Hence, option (c) is correct.

Problem-6

(2 marks)

Question

E is a boolean variable. Consider the following sequence of expressions:

```
1 | not E
2 | not not E
3 | not not not E
4 | not not not not E
5 | .
6 | .
7 | .
```

This pattern keeps repeating for a thousand lines. If line number 500 evaluates to False, what is the value of E?

- (a) True
- (b) False
- (c) Cannot be determined

Answer

(b)

Solution

This pattern evaluates True and False for the alternate line because one not operator is added in expression each time. So, if line number 500 evaluates to False that means the even-number line evaluates False and the odd-number line evaluates True. This means the line number 1 which is not E will be evaluated as True. So, the value of E is False. Hence, option (b) is correct.

Problem-7

(3 marks)

Question

`E_1` and `E_2` are two boolean variables. Consider the following code.

```
1 | E_1 and E_2 and 1 / 0
2 | print(E_2)
```

Which of the following scenarios are possible when the code given above is executed? Assume that all scenarios are independent of each other. It is a Multiple Select Question (MSQ).

- (a) The code throws an error.
- (b) `True` is printed after line-2 is executed.
- (c) `False` is printed after line-2 is executed.
- (d) None of the above.

Answer

(a), (b), (c)

Solution

<code>E_1</code>	<code>E_2</code>	<code>E_1 and E_2 and 1 / 0</code>	<code>print(E_2)</code>	Possible scenario for (<code>E_1 and E_2 and 1 / 0</code>)
False	False	False	False	<code>E_1</code> is <code>False</code> . Hence, the remaining part will not be evaluated.
False	True	False	True	<code>E_1</code> is <code>False</code> . So, the remaining part will not be evaluated
True	False	False	False	<code>E_1</code> is <code>True</code> . Therefore, <code>E_2</code> will be checked. Since, <code>E_2</code> is <code>False</code> so the remaining part will not be evaluated
True	True	Error	No Execution	<code>E_1</code> is <code>True</code> . Therefore, <code>E_2</code> will be checked. Now, since <code>E_2</code> is <code>True</code> , the remaining part <code>1/0</code> will be evaluated which results in error.

As we can see that there are three possible scenarios for execution. So, option (a), (b), and (c) are correct.

Problem-8

(2 marks)

Question

Consider the following string:

```
1 | word = '138412345678901938'
```

For what values of `a` and `b` does the following expression evaluate to `True`? Assume that `a` and `b` are both positive integers.

```
1 | word[a : b] == '123456789'
```

Answer

`a = 4, b = 13`

Solution

For `word = '138412345678901938'` index representation are given below:-

word=	1	3	8	4	1	2	3	4	5	6	7	8	9	0	1	9	3	8
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

For the condition `word[a : b] == '123456789'` evaluated to be `True`, the start index `a` should be `4` and stop index `b` should be `13` (1 more than the index of last character, because stop index not included in slicing range). Hence, `a = 4, b = 13`.

Week-1, Practice Assignment (programming) Solution

Week-1, Practice Assignment (programming) Solution

Problem-1

[Question](#)

[Answer](#)

[Solution](#)

Problem-2

[Question](#)

[Answer](#)

[Solution](#)

Problem-3

[Question](#)

[Test Cases](#)

[Public](#)

[Private](#)

[Answer](#)

[Solution](#)

Problem-4

[Question](#)

[Test Cases](#)

[Public](#)

[Private](#)

[Answer](#)

[Solution](#)

Problem-5

[Question](#)

[Test Cases](#)

[Public](#)

[Private](#)

[Answer](#)

[Solution](#)

Problem-1

Question

Print the following lines in Python

```
1 | 89074562
2 | Hii!!
3 | $$Howareyou??
4 | 10*10 = 100
5 | (12 > 10) and True
6 | I am here to learn "python"
```

Answer

```
1 | print('89074562')
2 | print('Hii!!')
3 | print('$$Howareyou?')
4 | print('10*10 = 100')
5 | print('(12 > 10) and True')
6 | print('I am here to learn "python"')
```

Solution

The print method displays what is given inside it on the console. After displaying the output, the control goes to the next line by default. That's why each line is printed on the new line.

Problem-2

Question

Print the following pattern.

```
1 | $  
2 | $ $  
3 | $ $ $  
4 | $ $ $ $  
5 | $ $ $ $ $
```

Note: There is a single space between two \$

Answer

```
1 | print('$')  
2 | print('$ $')  
3 | print('$ $ $')  
4 | print('$ $ $ $')  
5 | print('$ $ $ $ $')
```

Solution

Self explanatory.

Problem-3

Question

Accept a string as input. Make a copy of this string. Insert the exclamation mark (!) before and after the copy and print this final string to the console.

Test Cases

Public

Input	Output
cool	!cool!
123567	!1234567!

Private

Input	Output
a	!a!
!	!!!

Answer

```
1 | string = input()
2 | output = '!' + string + '!'
3 | print(output)
```

Solution

First, we read the input in a variable called `string`. We append "!" before and after the input string. This is stored in the variable `output` and is printed on the console.

Problem-4

Question

Accept an integer x as input. Find the quotient when 3^{35} is divided by x and print the output.

Test Cases

Public

Input	Output
10	5003154509899970
100	500315450989997

Private

Input	Output
300	166771816996665
2	25015772549499853

Answer

```
1 | x = int(input())
2 | print(3 ** 35 // x)
```

Solution

An integer is read into a variable `x`. 3^{35} is equivalent to the expression `3 ** 35`. In order to get the quotient `//` operator is used. The final output is printed.

Problem-5

Question

Accept a string as input. Print the string obtained by removing the first and last letters. For example:

Input

```
1 | peace
```

Output

```
1 | eac
```

Assume that there will be at least three characters in the input string.

Test Cases

Public

Input	Output
Rain	ai
abcdefg	bcd

Private

Input	Output
12345	234
aaaaa	aaa

Answer

```
1 | string = input()
2 | output = string[1 : -1]
3 | print(output)
```

Solution

The solution uses string slicing operation. Input string is read into the variable `string`. The index in a string starts at `0` and goes up to `len(string) - 1` when we move from left to right.

Conversely, the index starts at `-1` and ends in `-len(string)`, if we move from right to left.

Therefore, `string[1 : -1]` gives a substring starting from second character and ends in second last character skipping first and last character (Remember: from the lecture we know that the end position of slicing is not inclusive).

Week-1, Graded Assignment (programming)

(10 marks)

Week-1, Graded Assignment (programming)

Problem-1

 Question

 Answer

 Solution

Problem-2

 Question

 Answer

 Solution

Problem-3

 Question

 Test Cases

 Public

 Private

 Answer

 Solution

Problem-4

 Question

 Test Cases

 Public

 Private

 Answer

 Solution

Problem-5

 Question

 Test Cases

 Public

 Private

 Answer

 Solution

Problem-1

(1 marks)

Question

Print the following pattern.

1	1
2	11
3	111
4	1111
5	11111

Answer

```
1 | print('1')
2 | print('11')
3 | print('111')
4 | print('1111')
5 | print('11111')
```

Solution

Self explanatory.

Problem-2

(1 marks)

Question

The following pattern is called the Pascal's triangle. Print it!

```
1 |   1
2 |   1 1
3 |   1 2 1
4 | 1 3 3 1
```

Note

1. There is a single space between any two consecutive numbers in every row.
2. The number of spaces before the first number in each row keeps decreasing by exactly one unit from the first row to the last row.
3. There are no spaces before the first number in the last row.
4. There are no spaces after the last number in each row.

Answer

```
1 | print('  1')
2 | print(' 1 1')
3 | print(' 1 2 1')
4 | print('1 3 3 1')
```

Solution

The `print` method displays what is given inside brackets within single or double quotes as it is on the console. After displaying the output, the control goes to the next line by default.

Problem-3

(3 marks)

Question

Accept a string as input and create a copy of this string. Now, insert the length of the original string at the end of this copy and print the final string to the console.

Test Cases

Public

Input	Output
good	good4
abcdefg	abcdefg7

Private

Input	Output
123456789	1234567899
aaa	aaa3
	0

Answer

```
1 string = input()
2 string_len = str(len(string))
3 new_string = string + string_len
4 print(new_string)
```

Solution

- In the first line, take the input from the user and assign it to `string` variable.
- In the Second line, calculate the length of `string` and type cast the length value to `str` and assign to `string_len` variable.
- In the third Line, concatenate the `string` and `string_len` value and assign to `new_string` variable.
- In the fourth line, finally print `new_string` value.

Problem-4

(2 marks)

Question

Accept an integer x as input. Find the remainder when 2^{100} is divided by x and print the output.

Test Cases

Public

Input	Output
5	1
2	0

Private

Input	Output
10	6
100	76

Answer

```
1 | x = int(input())
2 | print(2 ** 100 % x)
```

Solution

- In the first line, take the input from the user, then change the input to integer assigned to x variable.
- In the Second line, write the python expression for the remainder when 2^{100} it is divided by within the print statement. So it will print the remainder value in output.

Problem-5

(3 marks)

Question

You are given the registration number of a vehicle as input. Print the two letter state-code where the vehicle is registered. You can assume that there will be no change in the format of the registration number. But the state code may change.

Test Cases

Public

Input	Output
TN 09 AV 1998	TN
AP 02 BK 1084	AP

Private

Input	Output
KL 02 BK 4910	KL
MH 13 GR 2491	MH
PB 00 AW 1948	PB

Answer

```
1 | regno = input()
2 | print(regno[0:2])
```

Solution

- In the first line, take the input from the user and assign it to `regno` variable.
- We know that the first two letters represent the state-code. So, the first letter is at index `0` and the second letter at index `1`.
- In Second line, use slicing for getting the first two letters with start index `0` and stop index `2` (because stop index is not included in range) within the `print` statement.
- Therefore, the line-2 will display the state-code in output.

Week-2, Practice Assignment (theory)

Week-2, Practice Assignment (theory)

Problem-1

 Question

 Answer

 Solution

Problem-2

 Question

 Answer

 Solution

Problem-3

 Question

 Answer

 Solution

Problem-4

 Question

 Answer

 Solution

Problem-5

 Question

 Answer

 Solution

Problem-6

 Question

 Answer

 Solution

Problem-7

 Question

 Answer

 Solution

Problem-8

 Question

 Answer

 Solution

Problem-9

 Question

 Answer

 Solution

Problem-10

 Question

 Answer

 Solution

Problem-1

Question

Which of the following are valid names for variables in Python? It is a Multiple Select Question (MSQ).

- (a) a_
- (b) _a
- (c) 1a
- (d) a variable
- (e) a_variable

Answer

- (a), (b), (e)

Solution

Rules for variable name in python:-

- A variable name must be started with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-Z, a-z, 0-9, and _).
- A variable name can not contain any blank space in it.

Hence, (a), (b), and (e) are correct according to the rules.

Problem-2

Question

Consider the following code-block. `word1` and `word2` are strings that have already been defined:

```
1 # word1, word2 are two strings
2 print(word1 + word2)
3 word = word1
4 word1 = word2
5 word2 = word
6 print(word2 + word1)
```

Which of the following statements are true about the two lines in the output?

- (a) The two lines are always the same.
- (b) The two lines are always different.
- (c) The two lines are the same if and only if `word1` and `word2` are equal.
- (d) None of the above

Answer

(a)

Solution

- Assume that `word1='Hello'` and `word2='Python'`.
- In line 2, `print(word1 + word2)` print the concatenation of `word1` and `word2` (`HelloPython`) in output.
- Line 3 to 5, swap the value of `word1` and `word2`.
- In line 6, `print(word2 + word1)` print the concatenation of `word1` and `word1` (`HelloPython`) in output which will be the same as the previous print statement output.

Hence, (a) is correct .

Problem-3

Question

Assume that `a`, `b` and `c` are three distinct integers. The following code runs without any error.

```
1 | x, y, z = a, b, c
2 | x = y = z
```

Which of the following statements evaluate to `True` at the end of execution of the code given above? It is a Multiple Select Question (MSQ).

(a) `x == y == z`

(b) `x == y == z == a`

(c) `x == y == z == b`

(d) `x == y == z == c`

Answer

(a), (d)

Solution

- In the first line, we are using multiple assignments in one line. So, after the first line of execution, `x = a`, `y = b` and `z = c`.
- In the second line, we Know that `=` operator has the right to left associativity. So, the value of `z` which is `c`, will be assigned to `y` and the value of `y` which is now `c`, will be assigned to `x`. So finally, all variables will contain the same value `c`.

Hence, (a) and (d) are correct .

Problem-4

Question

`x` is a variable of type `float` and is of the form `a.bcd`, where `a, b, c, d` are all positive integers less than 10. What is the output of the following snippet of code?

```
1 | print(int(-x))
```

- (a) `a`
- (b) `a + 1`
- (c) `-a`
- (d) `-a - 1`
- (e) `-a + 1`

Answer

(c)

Solution

`int(-x)` converts the float value to integer value and returns only the integer part(sign remains same) and discards the decimal part.

Hence, (c) is correct.

Problem-5

Question

Consider the following code-blocks. `E` is a Boolean expression.

Block-1:

```
1 if E:  
2     print('good')  
3 else:  
4     print('bad')
```

Block-2:

```
1 if E:  
2     print('good')  
3 print('bad')
```

Block-3:

```
1 print('good')  
2 print('bad')
```

Which of the following statements are true? It is a Multiple Select Question (MSQ).

- (a) All three blocks are equivalent to each other.
- (b) Exactly two of these three blocks are equivalent to each other.
- (c) Blocks 1 and 2 print the same output when `E` evaluates to `True`.
- (d) Blocks 1 and 2 print the same output when `E` evaluates to `False`.
- (e) Blocks 2 and 3 print the same output when `E` evaluates to `True`.

Answer

(d), (e)

Solution

For Block-1 possible cases:-

- If `E = True`, output will be `good`
- If `E = False`, output will be `bad`

For Block-2 possible cases:-

- If `E = True`, output will be `good` and `bad` both.
- If `E = False`, output will be `bad`

For Block-3 possible cases:-

- Block-3 code is not dependent on `E` so the output always will be `good` and `bad`.

Hence, (d) and (e) are correct according to the above observation .

Problem-6

Question

`bool_var` is a variable of type `bool`. `x` is a variable of type `int`. Assume that both these variables have already been defined. Now, consider the following code-block:

```
1 if bool_var:  
2     x = x + 1  
3     bool_var = not bool_var  
4     if bool_var:  
5         x = x + 1  
6     else:  
7         x = x - 1  
8 print(x)
```

Which of the following statements are true at the end of execution of the code-block given above?
It is a Multiple Select Question (MSQ).

- (a) The value of variable `x` is independent of the value of `bool_var`.
- (b) The value of variable `x` is dependent on the value of `bool_var`.
- (c) Line-5 is never executed.
- (d) The variable `x` is updated exactly two times.

Answer

(a), (c)

Solution

If `bool_var = True`

- Inside the `if` block, In line 2, `x` will be incremented by 1
- In line 3, the value of `bool_var` becomes `False` so in line 6, `else` block will be executed in which `x` will be decremented by 1. So finally `x` value will be printed which remains the same as the initial. We can see that here line 4 `if` block will never execute.

If `bool_var = False`

- Just print the value of `x`

Hence, (a) and (c) are correct.

Common data for problems 7 and 8

Consider the following code-block. `E_1`, `E_2` and `E_3` are all Boolean variables that have already been defined. `x` is a variable that has NOT been defined before.

```
1 | if E_1:  
2 |     x = 1  
3 | if E_2:  
4 |     x = 2  
5 | if E_3:  
6 |     x = 3  
7 | print(x)
```

Problem-7

Question

When will this code throw an error?

- (a) When all three Boolean variables are `True`.
- (b) When all three Boolean variables are `False`.
- (c) When at least one of the three Boolean variables is `True`.
- (d) When at least one of the three Boolean variables is `False`.
- (e) This code will never throw an error.

Answer

(b)

Solution

All possible cases for `E_1`, `E_2` and `E_3`:-

E_1	E_2	E_3	Output
<code>True</code>	<code>True</code>	<code>True</code>	3
<code>True</code>	<code>True</code>	<code>False</code>	2
<code>True</code>	<code>False</code>	<code>True</code>	3
<code>True</code>	<code>False</code>	<code>False</code>	1
<code>False</code>	<code>True</code>	<code>True</code>	3
<code>False</code>	<code>True</code>	<code>False</code>	2
<code>False</code>	<code>False</code>	<code>True</code>	3
<code>False</code>	<code>False</code>	<code>False</code>	<code>NameError: name 'x' is not defined</code>

We can see that here if any of the Boolean variables is True so x value will be initialized in `if` block otherwise x will not be initialized. So when all three Boolean variables are `False` then name 'x' is not defined error comes. Hence, option (b) is correct.

Problem-8

Question

If the code throws an error, in which line will it occur? Enter an integer between 1 and 7, both endpoints included.

Answer

7

Solution

All possible cases for `E_1`, `E_2` and `E_3` :-

E_1	E_2	E_3	Output
True	True	True	3
True	True	False	2
True	False	True	3
True	False	False	1
False	True	True	3
False	True	False	2
False	False	True	3
False	False	False	NameError: name 'x' is not defined

We can see that here if any of the Boolean variables is True so x value will be initialized in `if` block otherwise x will not be initialized. So when all three Boolean variables are `False` line number 7 gives error name 'x' is not defined . Hence, the answer is 7.

Problem-9

Question

Chose the correct matching option for operator and related function in string .

Operator	Function related to string
1. +	A. Replicates same string multiple times
2. *	B. Membership check
3. []	C. Escape character
4. [:]	D. Concatenates two string
5. in	E. Range slice
6. \	F. Character of the string using indexing

Chose the correct option for given code.

- (a) 1-D, 2-A, 3-F, 4-E, 5-B, 6-C
- (b) 1-D, 2-A, 3-F, 4-E, 5-C, 6-B
- (c) 1-D, 2-A, 3-E, 4-F, 5-B, 6-C
- (d) 1-A, 2-D, 3-E, 4-F, 5-B, 6-C

Answer

- (a) 1-D, 2-A, 3-F, 4-E, 5-B, 6-C

Solution

Correct answer is option (a)

Operator	Function related to string
1. +	D. Concatenates two string
2. *	A. Replicates same string multiple times
3. []	F. Character of the string using indexing
4. [:]	E. Range slice
5. in	B. Membership check
6. \	C. Escape character

Problem-10

Third index was not introduced in the lectures, student has to search through the various resources available on internet and answer the following questions below.

Question

```
1 s = "abcdefghijklmnopqrstuvwxyz"
2 a = int(input())
3 b = int(input())
4 c = int(input())
5 d = int(input())
6 e = int(input())
7 print(s[-a:-len(s):-3])
8 print(s[::-b])
9 print(s[c:0:-3])
10 print(s[len(s):-d:-3])
11 print(s[:e:-3])
```

For what user input of `a`, `b`, `c`, `d` and `e` (where `0 <= a,b,c,d,e <= len(s)`) does the above code-snippet print following output? It is a Numerical Answer Type (NAT) question. Enter your input value separated by `,` without any space like: `7,15,0,4,0`

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

Answer

1,3,26,26,0

or

1,3,25,26,0

or

1,3,26,0,0

or

1,3,25,0,0

Solution

`s[start : end : step]`

`start` is start index.

`end` is last index+1(because end is not inclusive).

`step` define interval of indexing in range (`start , start+step , start+step+step,.....end-1`).

For traverse string left to right:-

- If `start`, `end` and `step` is not defined in slicing then by default `start` will be first index of string, `end` will be one more then from last index of element and `step` will be 1.
- If we are visiting elements from left to right ,we must remember that the `start` index must be at the left side from the `end` index and the `step` value should be positive.

For traverse string right to left:-

- `step` value must be mention and it should be negative
- If we are visiting elements from right to left ,we must remember that the `start` index must be at the right side from the `end` index and the `step` value should be negative.
- If `start` and `end` index is not defined and `step` value is negative so by default `start` index will be the last index of string and `end` index will be one less then from first index of string.

We can see that in the output, characters are printed in reverse order from last character to first character with an interval of 3. So all possible combinations given in the answer will print the same output `zwtqnkheb` 5 times.

Week-2, Graded Assignment (theory)

Week-2, Graded Assignment (theory)

Problem 1

 Question

 Input

 Answer

 Solution

Problem 2

 Question

 Answer

 Solution

Problem 3

 Question

 Answer

 Solution

Problem 4

 Question

 Answer

 Solution

Problem 5

 Question

 Answer

 Solution

Problem 6

 Question

 Answer

 Solution

Problem 7

 Question

 Answer

 Solution

Problem 8

 Question

 Answer

 Solution

Problem 9

 Question

 Answer

 Solution

Problem 10

 Question

 Code-1

 Code-2

 Answer

 Solution

Problem 11

 Question

 Input

 Output

 Answer

 Solution

Problem 12

 Question

[Input](#)

[Output](#)

[Answer](#)

[Solution](#)

[Problem 13](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 14](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 15](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 16](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 17](#)

[Question](#)

[Answer](#)

[Solution](#)

Problem 1

Question

What will be the output of `print(a)` after executing the below code?

```
1 | a, b, c, d = input()
```

Input

```
1 | 1234
```

Answer

```
1
```

Solution

Variable	Value
a	'1'
b	'2'
c	'3'
d	'4'

The statement accepts a string of length 4 and assigns each character to variables a, b, c and d in the order it is entered. If the number of characters in the entered string is not equal to 4, `ValueError` is thrown by the interpreter.

Problem 2

Question

What will be the output of `print(b)` after executing the below code?

```
1 | 1234
```

Answer

```
2
```

Solution

See problem 1

Problem 3

Question

What will be the output of `print(c)` after executing the below code?

```
1 | 1234
```

Answer

3

Solution

See problem 1

Problem 4

Question

What will be the output of `print(d)` after executing the below code?

```
1 | 1234
```

Answer

4

Solution

See problem 1

Problem 5

Question

What is the value stored in the variable `x` after the following line of code is executed? Assume that the user gives `False` as the input.

```
1 | x = bool(False)
```

- (a) `False`
- (b) `True`
- (c) `ValueError`
- (d) `'False'`

Answer

- (a) `False`

Solution

`bool()` is a built-in function that converts a value passed as an argument to a `bool` literal `True` or `False`. If the argument is an empty string, it returns `False`, else it will return `True`. If the value inside the function is a `bool` literal, the same is returned by the function. If in case the argument is an `int` or `float`, `False` is returned when it is `0` or `0.0`, otherwise `True` is returned.

Problem 6

Question

What is the value stored in the variable `x` after the following line of code is executed? Assume that the user gives `False` as the input.

```
1 | x = bool(input())
```

- (a) `False`
- (b) `True`
- (c) `ValueError`
- (d) `'False'`

Answer

(b) `True`

Solution

As you know, the function `input()` reads the value passed by the user as a string. Here, the value entered by user is `False`. The `input()` function returns it as a string literal `'False'`. It is then transformed by the function `bool()` to the boolean literal `True` and is stored in the variable `x`.

The given code can be interpreted as:

```
1 | x = bool('False')
2 | # input to bool is a string
3 | # as input is not the empty string, True is the output of bool('False')
```

Below table shows how values are transformed by `bool()` function:

Type	Values	Boolean Conversion
<code>int</code>	<code>0</code>	<code>False</code>
<code>int</code>	<code>1, 2, 3, ..., -1, -2, ..., 10**2</code>	<code>True</code>
<code>float</code>	<code>0.0, -0.0</code>	<code>False</code>
<code>float</code>	<code>0.00001, 2.718</code>	<code>True</code>
<code>str</code>	<code>"", "a", "hello"</code>	<code>False</code>
<code>str</code>	<code>" ", "a", "hello"</code>	<code>True</code>

Use the below code to answer Question number 7,8 and 9.

Code-1

```
1 | a = int(input())
2 | b = int(input())
3 |
4 | if a > 0:
5 |     if b < 0:
6 |         print('OK')
```

Code-2

```
1 | if int(input()) > 0 and int(input()) < 0:
2 |     print('Check OK')
```

Problem 7

Question

Choose the correct statements. It is a Multiple Select Question (MSQ).

- (a) Code-1: Always accepts two inputs
- (b) Code-2: Always accepts two inputs
- (c) Code-1: If the first input is negative then program completes without printing anything
- (d) Code-2: If the first input is negative then program completes without printing anything
- (e) Code-1: It is possible to change the value of `a` by introducing a new line of code before accepting the value of `b`

Answer

- (a) Code-1: Always accepts two inputs
- (c) Code-1: If the first input is negative then program completes without printing anything
- (d) Code-2: If the first input is negative then program completes without printing anything
- (e) Code-1: It is possible to change the value of `a` by introducing a new line of code before accepting the value of `b`

Solution

Option (a) is correct. The interpreter executes statements sequentially in a program. In Code-1 block, line-1 accepts user input and store in a variable `a`. On line-2, it accepts another input from console and stores in variable `b`. Therefore, it always accepts two inputs.

Option (c) is correct. In Code-1, if the first input number `a` is negative, the condition `a > 0` evaluates to `False`, the control doesn't enter inside the body of this `if` block and therefore subsequent statement `if b < 0:` is skipped.

Option (d) is correct. If the number entered for the first `input()` call is less than `0`, the condition `int(input()) > 0` and `int(input()) < 0` evaluates to `False`. Thus, neither second `input()` is executed, nor the `print('OK')` statements inside the body of this `if` block.

Option (e) is correct. One can change the value of `a` before accepting the second variable `b` as input. Here is an example to reassign `1` to the variable `a` on line-2.

```
1 a = int(input())
2 a = 1
3 b = int(input())
4 if a > 0:
5     if b < 0:
6         print('OK')
```

Problem 8

Question

How many comparisons are done for the following input in Code-1?

1	-1
2	1

Answer

1

Solution

In Code-1, only 1 comparison occurs for given input. Since the first input number `a` is -1, the condition `a > 0` evaluates to `False`, the control doesn't enter inside the body of this `if` block and therefore subsequent conditional statement is skipped.

Problem 9

Question

How many comparisons are done for the following input in Code-2?

1	1
2	1

Answer

2

Solution

In Code-2, 2 comparisons occur. Let us look at the expression in the conditional statement:

```
int(input()) > 0 and int(input()) < 0
```

Since the first input number is greater than 0, the first part of the expression `int(input()) > 0` evaluates to `True`. Therefore, the Python interpreter will go ahead and do the second comparison `int(input()) < 0`. This is because the operator is `and` and the operand to the left of this operator evaluates to `True`. In such a case, the operand to the right should also be evaluated to find the value of the entire expression. Refer short-circuit evaluation in Python for more details.

Problem 10

Question

Code-1

```
1 if a:  
2     if b:  
3         print('OK')  
4     if c:  
5         print('OK')
```

Code-2

```
1 if a and b or c:  
2     print('OK')
```

Choose the correct statements, given that a, b and c are boolean values. It is a Multiple Select Question (MSQ).

- (a) Code-1: OK will be printed once if either of a or b is True given that c is False
- (b) Code-2: OK will be printed once if either of a or b is True given that c is False
- (c) Code-1: OK will be printed twice only if all a , b and c are True
- (d) Code-2: OK will be printed only if all a , b and c are True
- (e) Code-2: OK will be printed if c is True
- (f) Code-1 and Code-2 will give same result for same values of a , b and c

Answer

- (c) Code-1: OK will be printed twice only if all a , b and c are True
- (e) Code-2: OK will be printed if c is True

Solution

Option (c) is correct. In Code-1, the interpreter executes the body of outer `if` block, only if the boolean variable `a` is `True`. The first inner `print('OK')` statement is executed if boolean variable `b` is `True`. Similarly, the second `print('OK')` is executed when the boolean variable `c` is `True`.

Option (e) is correct. In Code-2, the conditional expression `a and b or c` contains two logical operators `and` and `or`. It is parenthesized as `(a and b) or c` according to the precedence rules (`and > or`). This condition evaluates to `True`, if `c` is `True`, irrespective of the values of `a` and `b`.

Use the below code to answer Question number 11,12,13 and 14.

```
1 x = int(input())
2 y = int(input())
3 z = int(input())
4
5 # Block-1 Start
6 if x > 0 or y > 0 or z > 0:
7     if (x > 0 and y > 0) or (y > 0 and z > 0) or (z > 0 and x > 0):
8         if x > 0 and y > 0 and z > 0:
9             print('P3')
10        else:
11            print('P2')
12    else:
13        print('P1')
14 # Block-1 End
15
16 # Block-2 Start
17 if x < 0 or y < 0 or z < 0:
18     if (x < 0 and y < 0) or (y < 0 and z < 0) or (z < 0 and x < 0):
19         if x < 0 and y < 0 and z < 0:
20             print('N3')
21         else:
22             print('N2')
23     else:
24         print('N1')
25 # Block-2 End
```

Problem 11

Question

What will be the value of X in the output for the given input? It is a Numerical Answer Type (NAT) question.

Input

1	-1
2	4
3	1

Output

1	PX
2	NY

Answer

2

Solution

In the first nested `if` structure (Block-1), the outermost `if` condition checks whether at least one of the variables `x`, `y`, `z` is positive. The intermediate `if` block is executed when two of these variables are positive, otherwise `else` block is executed and `P1` is printed.

Inside the intermediate `if` body, the innermost `if` block is executed, and `P3` is printed when all three variables are positive, otherwise `else` block is executed and `P2` is printed on the console.

Similar analogy for Block-2.

Here is a table showing the input and output relation.

Input	Output	X	Y
One of the numbers is positive	P1	1	
Two numbers are positive	P2	2	
All three numbers are positive	P3	3	
One of the numbers is negative	N1		1
Two numbers are negative	N2		2
All three numbers are negative	N3		3
All three numbers are zero			

Problem 12

Question

What will be the value of Y in the output for the given input? It is a Numerical Answer Type (NAT) question.

Input

1	-1
2	4
3	1

Output

1	PX
2	NY

Answer

1

Solution

See problem 11

Problem 13

Question

When does the above code print no value?

- (a) When any two among x , y and z are equal
- (b) When all the values of x , y and z are equal
- (c) When any one among x , y and z is zero
- (d) When all the values of x , y and z are zeros

Answer

- (d) When all the values of x , y and z are zeros

Solution

Option (d) is correct. When all variables set to 0, Outermost if block evaluates False in both Block-1 and Block-2. Hence, nothing is printed.

Problem 14

Question

For any input `x`, `y` and `z` at least one `else` statement will be executed.

- (a) True
- (b) False

Answer

- (b) False

Solution

Option (b) is correct. In below situations, none of the `else` statement are executed.

- All variables are set to `0`,
- All variables are positive,
- All variables are negative

Problem 15

Question

```
1 | import math as ma
```

Select the correct way of accessing the `sqrt` function from `math` library.

- (a) `math.sqrt()`
- (b) `ma.sqrt()`
- (c) `sqrt()`
- (d) `math.ma.sqrt()`

Answer

(b) `ma.sqrt()`

Solution

A library can be included in the program using the keyword `import`. The `as` keyword allows us to call various functions of a library using a custom alias (name). Here `math` library is imported under the alias `ma`. Hence, (b) `ma.sqrt()` is the correct answer.

Problem 16

Question

What should be input to the following code to get the below output. This is a MCQ type question.

```
1 dept    = input()
2 course  = input()
3 prefix   = input()
4 name    = input()
5 roll_no = input()
6 name = prefix + " " + name
7 lib_id = dept[0] + course[0] + roll_no
8 print("Student record:")
9 indent = '    '
10 print(indent+"Dept:", dept)
11 print(indent+"Name:", name)
12 print(indent+"Roll No:", roll_no)
13 print(indent+"Library Card No:", lib_id)
```

Output

```
1 Student record:
2     Dept: ABC
3     Name: MR FNAME LNAME
4     Roll No: 999
5     Library Card No: AX999
```

(a)

```
1 ABD
2 XYZ
3 MR
4 FNAME LNAME
5 999
```

(b)

```
1 ABC
2 XYZ
3 MRS
4 FNAME LNAME
5 999
```

(c)

```
1 ABC
2 XYZ
3 MR
4 FNAME LNAME
5 999
```

(d)

1	ABC
2	MR
3	XYZ
4	FNAME LNAME
5	999

Answer

(c)

1	ABC
2	XYZ
3	MR
4	FNAME LNAME
5	999

Solution

Self explanatory.

Problem 17

```
1 match = False
2 if s.count('(') == s.count(')'):
3     if s.count('[') == s.count(']'):
4         if s.count('{') == s.count('}'):
5             match = True
```

Question

If $s = "abcd(efghijkl){{}))"$ what will be value of **match** at the end of execution and justification?

- (a) False, the number of opening and closing brackets are only considered
- (b) False, the position of opening and closing brackets are not considered.
- (c) True, the number of opening and closing brackets are only considered
- (d) True, the number and position of opening and closing brackets are considered.

Answer

- (c) True, the number of opening and closing brackets are only considered

Solution

The code blocks initialize the boolean variable `match` to `False`. In outermost `if` statement `s.count('(') == s.count(')')`, count of opening parentheses `(` and closing parentheses `)` are compared. If this is `True`, in the intermediate `if` condition, the count of opening and closing square brackets `[` and `]` are compared. If the previous two `if` statements evaluate `True`, the innermost `if` condition executes and verifies whether the count of opening and closing braces `{` and `}` are equal. If all `if` statements evaluate `True`, the boolean variable `match` set to `True`. It only takes into consideration the number of opening and closing brackets (parentheses, square brackets and braces), not their position.

Week-2 Practice Assignment (Programming)

Week-2 Practice Assignment (Programming)

Problem-1

 Question

 Testcases

 Public

 Private

 Answer

 Solution

 Tags

Problem-2

 Question

 Testcases

 Public

 Private

 Answer

 Solution

 Tags

Problem-3

 Question

 Testcases

 Public

 Private

 Answer

 Solution

 Tags

Problem-4

 Question

 Testcases

 Public

 Private

 Answer

 Solution

Problem-5

 Question

 Testcases

 Public

 Private

 Answer

 Solution

 Tags

Problem-1

(1 marks)

Question

Evaluate the below piecewise function using Python.

$$y = \begin{cases} x + 2 & 0 < x < 10 \\ x^2 + 2 & 10 \leq x \\ 0 & \text{otherwise} \end{cases}$$

The value of the variable `x` should be an numerical input from the user.

Testcases

Public

Input	Output
5	7.0
15	227.0

Private

Input	Output
-100	0
0	0
7	9.0
10	102.0
128.3	16462.89

Answer

```
1 x = float(input())
2 if 0 < x < 10:
3     y = x + 2
4 elif 10 <= x:
5     y = x**2 + 2
6 else:
7     y = 0
```

Solution

In this answer code, line 2 `if` statement covered the first condition of the piecewise function. If the first condition is not satisfied then line 4 `elif` statement will cover the second condition of the piecewise function otherwise line 6 `else` statement will cover the third condition of the piecewise function.

Tags

if, elif, piecewise

Problem-2

(2 marks)

Question

Write a Python code to find the quadrant of a point taken as input from the user. The input is given in 2 lines with the first and second lines representing the x coordinate and y coordinate of the point respectively. The possible outputs are I, II, III, IV, X-axis, Y-axis, and Origin. Any other output will not be accepted, Take care of the upper and lower cases while printing the output.

Testcases

Public

Sample Input-1

1	1
2	2

Sample Output-1

1	I
---	---

Sample Input-2

1	1.8
2	-1

Sample Output-2

1	IV
---	----

Private

Input-1

1	0.0
2	0

Output-1

1	Origin
---	--------

Input-2

1	7.0
2	0

Output-2

1	X-axis
---	--------

Input-3

1	0
2	-14.12

Output-3

1	Y-axis
---	--------

Input-4

1	-3.2
2	-3

Output-4

1	X-axis
---	--------

Input-5

1	0.1
2	0.98

Output-5

1	I
---	---

Input-6

1	1.2
2	-7.09

Output-6

1	IV
---	----

Input-7

1	-7.09
2	1.2

Output-7

1	II
---	----

Answer

```
1 x = float(input())
2 y = float(input())
3
4 if x > 0:
5     if y > 0:
6         print('I')
7     elif y < 0:
8         print('IV')
9     else:
10        print('X-axis')
11 elif x < 0:
12     if y > 0:
13         print('II')
14     elif y < 0:
15         print('III')
16     else:
17        print('X-axis')
18 else:
19     if y != 0:
20         print('Y-axis')
21     else:
22         print('Origin')
```

Solution

According to input of x and y value there are 7 possible cases to find the quadrant:-

x	y	output
positive	positive	I
positive	negative	IV
positive or negative	0	X-axis
negative	positive	II
negative	negative	III
0	positive or negative	Y-axis
0	0	Origin

These all cases are covered in answer code using `if-elif-else` statements.

Tags

Problem-3

(2 marks)

Question

Write a Python code to realize the equation of a line given 2 points (x_1, y_1) and (x_2, y_2) . The input is in 5 lines where, the first, second, third, and fourth line represent x_1, y_1, x_2 , and y_2 respectively. The fifth line corresponds to x_3 . Determine y_3 using the line equation given below.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$$

The output should be "Vertical Line" if the line is vertical. In other cases, the output should be 2 lined, where the first line is the value of y_3 and the second line indicates whether the slope of the line is positive or negative. Print "Positive Slope" and "Negative Slope" accordingly.

Testcases

Public

Sample Input - 1

1	1
2	2
3	1
4	6
5	5

Sample Output - 1

1	Vertical Line
---	---------------

Sample Input - 2

1	1
2	4
3	5
4	6
5	2

Sample Output - 2

1	4.5
2	Positive Slope

Private

Input-1

1	1
2	1
3	-1
4	5.3
5	0

Output-1

1	3.15
2	Negative Slope

Input-2

1	1
2	1
3	1
4	5.3
5	0

Output-2

1	Vertical Line
---	---------------

Input-3

1	5.3
2	1
3	1
4	1
5	0

Output-3

1	1.0
2	Horizontal Line

Input-4

1	5.3
2	1
3	1
4	0.5
5	0

Output-4

1	0.38372093023255816
2	Positive Slope

Answer

```
1 x1, y1 = float(input()), float(input())
2 x2, y2 = float(input()), float(input())
3 x3 = float(input())
4 if x1 != x2:
5     m = (y2 - y1) / (x2 - x1)
6     y3 = y1 + m * (x3 - x1)
7     print(y3)
8     if m == 0:
9         print('Horizontal Line')
10    elif m > 0:
11        print('Positive Slope')
12    else:
13        print('Negative Slope')
14 else:
15     print('Vertical Line')
```

Solution

For the vertical line, the slope will be infinity hence it will give zero division error. Hence, in order to avoid the error, the case is avoided and printed directly by checking for the equality of x_1 and x_2 values. After calculation of `m` use `if-elif-else` to print output.

Tags

if, else, elif

Problem-4

(2 marks)

Question

Accept a string and return a new string formed using the middle three characters. If the input string is of even length, make the string of odd length as below

- add a period . at the end if it is not there,
- otherwise remove the period .

Testcases

Public

Input

```
1 | Peter Piper picked a peck of pickled peppers.
```

Output

```
1 | pec
```

Input

```
1 | floccinaucinihilipilification
```

Output

```
1 | hil
```

Private

Input	Output
Look before you leap	re
Readability counts.	ity
counting clocks	ng
Atoms make up everythings.	p e

Answer

```
1 input_string = input("Enter a string: ")
2 length = len(input_string)
3 if length % 2 == 0:
4     if input_string[length - 1] == '.':
5         input_string = input_string[0 : length - 1]
6     else:
7         input_string = input_string + "."
8 middle_position = (length) // 2
9 output_string = input_string[middle_position - 1 : middle_position + 2]
10 print(output_string)
```

Solution

- After accepting input strings from the user find out the length of the string (line 1 to 2).
- If the length of the string is even then make this string of odd length by adding `.` or removing `.` according to the condition given in problem (line 3 to 7).
- Find the middle position and for middle three characters assign start index(`middle_position - 1`) and end index(`middle_position + 2`) in slice range (line 8 to 9).
- Print the output string(line 10).

Problem-5

(3 marks)

Question

Evaluate the output **d** based on three given Boolean inputs **a**, **b** and **c**.

a	b	c	d
False	False	False	<i>False</i>
False	False	True	<i>True</i>
False	True	False	<i>False</i>
False	True	True	<i>True</i>
True	False	False	<i>True</i>
True	False	True	<i>True</i>
True	True	False	<i>False</i>
True	True	True	<i>True</i>

Testcases

Public

Sample Input - 1

```
1 | True
2 | False
3 | True
```

Sample Output - 1

```
1 | True
```

Sample Input - 2

```
1 | True
2 | True
3 | True
```

Sample Output - 2

```
1 | True
```

Sample Input - 3

```
1 | False  
2 | False  
3 | False
```

Sample Output - 3

```
1 | False
```

Private

Input-1

```
1 | False  
2 | False  
3 | False
```

Output-1

```
1 | False
```

Input-2

```
1 | False  
2 | False  
3 | True
```

Output-2

```
1 | True
```

Input-3

```
1 | False  
2 | True  
3 | False
```

Output-3

```
1 | False
```

Input-4

```
1 | False  
2 | True  
3 | True
```

Output-4

```
1 | True
```

Input-5

```
1 | True  
2 | False  
3 | False
```

Output-5

```
1 | True
```

Input-6

```
1 | True  
2 | False  
3 | True
```

Output-6

```
1 | True
```

Input-7

```
1 | True  
2 | True  
3 | False
```

Output-7

```
1 | False
```

Input-8

```
1 | True  
2 | True  
3 | True
```

Output-8

```
1 | True
```

Answer

```
1 | if input() == 'True':  
2 |     a = True  
3 | else:  
4 |     a = False  
5 | if input() == 'True':  
6 |     b = True  
7 | else:  
8 |     b = False
```

```
9 if input() == 'True':  
10     c = True  
11 else:  
12     c = False  
13 d = a and not(b) or c  
14 print(d)
```

Solution

Accept the input from user `True` or `False` as string and assign to variable as bool by `if-else` statement (line 1 to 12). Line 13 expression returns `True` or `False` value to variable `d` according to possible cases given in problem, then print value of `d` in line 14.

Tags

Week-2, Graded, Programming

Week-2, Graded, Programming

Problem-1

 Question

 Input-Output

 Specification

 Examples

 Answer

 Test cases

 Public

 Private

 Solution

Problem-2

 Question

 Input-Output

 Specification

 Examples

 Answer

 Test Cases

 Public

 Private

 Solution

Problem-3

 Question

 Input-Output

 Specification

 Examples

 Answer

 Answer-3

 Test Cases

 Public

 Private

 Solution

Problem-4

 Question

 Answer

 Test Cases

 Public

 Private

 Solution

Problem-5

 Question

 Input-Output

 Specification

 Examples

 Answer

 Answer-2

 Test Cases

 Public

 Private

 Solution

Problem-6

 Question

[Answer](#)

[Test Cases](#)

[Public](#)

[Private](#)

[Solution](#)

Problem-1

Question

Accept three positive integers as input from the user and check if they form the sides of a right triangle. Print YES if they form one, and NO if they do not.

Input-Output

Specification

The input will have three lines, with one integer on each line.

The output will be a single line containing one of these two strings: YES or NO.

Examples

Input-1

1	1
2	2
3	3

Output-1

1	NO
---	----

Input-2

1	3
2	4
3	5

Output-2

1	YES
---	-----

Answer

```
1 x = int(input())
2 y = int(input())
3 z = int(input())
4 if x >= y and x >= z:
5     z, x = x, z
6 elif y >= x and y >= z:
7     z, y = y, z
8 if x ** 2 + y ** 2 == z ** 2:
9     print('YES')
10 else:
11     print('NO')
```

Test cases

Public

Input-1

1	1
2	2
3	3

Output-1

1	NO
---	----

Input-2

1	3
2	4
3	5

Output-2

1	YES
---	-----

Private

Input-1

1	8
2	1
3	5

Output-1

1	NO
---	----

Input-2

1	12
2	13
3	5

Output-2

1	YES
---	-----

Input-3

1	80
2	39
3	89

Output-3

1	YES
---	-----

Input-4

1	32
2	60
3	68

Output-4

1	YES
---	-----

Input-5

1	2
2	1
3	8

Output-5

1	NO
---	----

Solution

```
1 x = int(input())
2 y = int(input())
3 z = int(input())
4 if x >= y and x >= z:
5     z, x = x, z
6 elif y >= x and y >= z:
7     z, y = y, z
8 if x ** 2 + y ** 2 == z ** 2:
9     print('YES')
10 else:
11     print('NO')
```

The basic idea of the solution is as follows: in a Pythagorean triplet, the hypotenuse will always be greater than the other two sides. So, we extract the maximum among the three numbers and see if the sum of squares of the other two numbers is equal to the square of this maximum. One way to implement this is to enforce `z` to hold the maximum value.

If `z` is already the maximum, we don't have to do anything. If not, then we need to check which among `x` and `y` is the maximum, and swap `z` with this variable which holds the maximum. For example, in line-4, we are checking if `x` is the maximum. If it is, then we swap `x` and `z`. Likewise, line-6 checks if `y` is the maximum. If it is, then we swap `y` and `z`. If the conditions in lines 4 and 6 both evaluate to `False`, then we know that `z` is the maximum by default.

At this stage, `z` holds the maximum among the three numbers. Finally, in lines 8 to 11, we check if the three numbers form a Pythagorean triplet or not.

Problem-2

Question

Accept an integer between 0 and 23 (both endpoints included) from the user and print what time of the day it is. Use the following table for reference. If the time is outside this range, print the string INVALID.

Input	Output
$T < 0$	INVALID
$0 \leq T \leq 5$	NIGHT
$6 \leq T \leq 11$	MORNING
$12 \leq T \leq 17$	AFTERNOON
$18 \leq T \leq 23$	EVENING
$T \geq 24$	INVALID

Input-Output

Specification

The input will be a single line containing an integer.

The output will be one of these strings: NIGHT, MORNING, AFTERNOON, EVENING

Examples

Input-1

```
1 | 3
```

Output-1

```
1 | NIGHT
```

Input-1

```
1 | 14
```

Output-2

```
1 | AFTERNOON
```

Answer

```
1 t = int(input())
2 if 0 <= t <= 5:
3     print('NIGHT')
4 elif 6 <= t <= 11:
5     print('MORNING')
6 elif 12 <= t <= 17:
7     print('AFTERNOON')
8 elif 18 <= t <= 23:
9     print('EVENING')
10 else:
11     print('INVALID')
```

Test Cases

Public

Input	Output
3	NIGHT
14	AFTERNOON

Private

Input	Output
9	MORNING
24	INVALID
12	AFTERNOON
-1	INVALID
19	EVENING

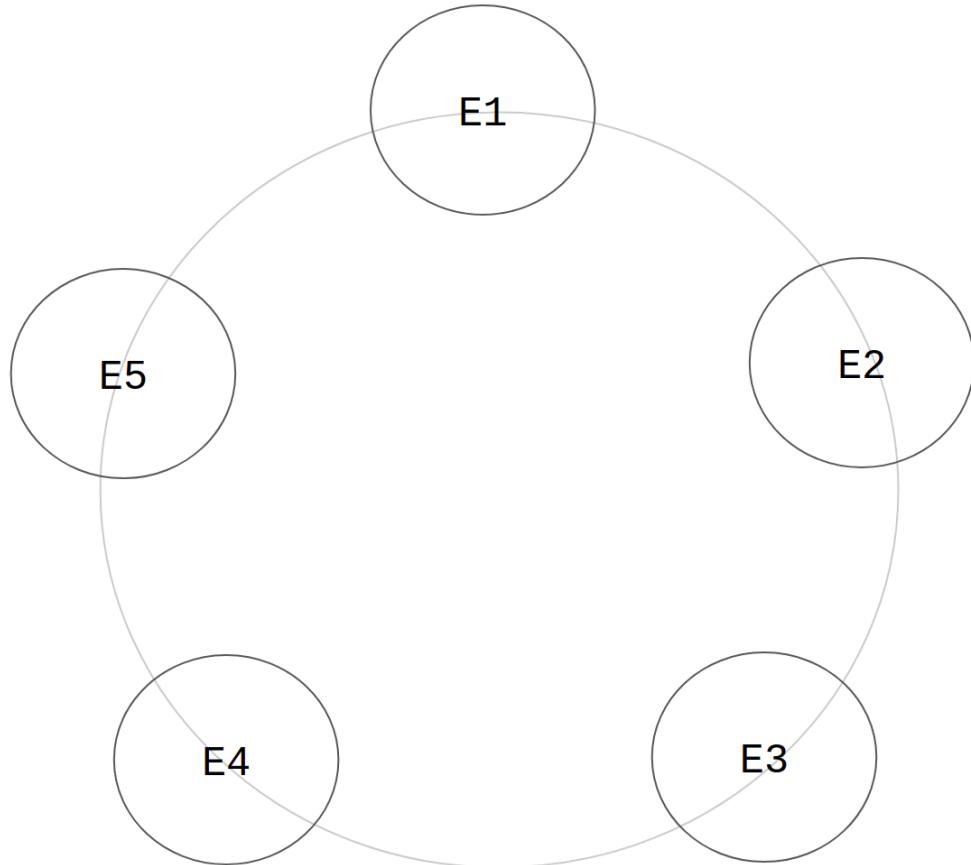
Solution

It is a simple `if-elif-else` ladder that checks for these conditions.

Problem-3

Question

`EvenOdd` is a tech startup. Each employee at the startup is given an employee id which is a unique positive integer. On one warm Sunday evening, five employees of the company come together for a meeting and sit at a circular table:



The employees follow a strange convention. They will continue the meeting only if the following condition is satisfied.

The sum of the employee-ids of any two adjacent employees at the table must be an even number.

They are so lazy that they won't move around to satisfy the above condition. If the current seating plan doesn't satisfy the condition, the meeting will be canceled. You are given the employee-id of all five employees. Your task is to decide if the meeting happened or not.

Input-Output

Specification

The input will be five lines, each line containing an integer. The i^{th} line will have the employee-id of E_i .

The output will be a single line containing one of these two strings:

- YES
- NO

Examples

Input

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

Output

```
1 | NO
```

Input

```
1 | 2
2 | 4
3 | 6
4 | 8
5 | 10
```

Output

```
1 | YES
```

Answer

```
1 e1 = int(input())
2 e2 = int(input())
3 e3 = int(input())
4 e4 = int(input())
5 e5 = int(input())
6 if (e1 + e2) % 2 != 0:
7     print('NO')
8 elif (e2 + e3) % 2 != 0:
9     print('NO')
10 elif (e3 + e4) % 2 != 0:
11     print('NO')
12 elif (e4 + e5) % 2 != 0:
13     print('NO')
14 elif (e5 + e1) % 2 != 0:
15     print('NO')
16 else:
17     print('YES')
```

Answer-3

```
1 e1, e2, e3, e4, e5 = int(input()), int(input()), int(input()), int(input()),
  int(input())
2 if (e1 + e2) % 2 == (e2 + e3) % 2 == (e3 + e4) % 2 == (e4 + e5) % 2 == 0:
3     print("YES")
4 else:
5     print("NO")
```

Test Cases

Public

Input-1

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

Output-1

```
1 NO
```

Input-2

```
1 2
2 4
3 6
4 8
5 10
```

Output-2

```
1 YES
```

Private

Input-1

```
1 1
2 101
3 3
4 303
5 5
```

Output-1

```
1 YES
```

Input-2

1	60
2	98
3	1738
4	52
5	48

Output-2

1	YES
---	-----

Input-3

1	1
2	10
3	3
4	30
5	5

Output-3

1	NO
---	----

Input-4

1	1
2	11
3	111
4	1111
5	11110

Output-4

1	NO
---	----

Solution

If we pick any two adjacent employees, their ids should sum to an even number. For example, if there are five employees, then we have to consider the following pairs:

- 1 and 2
- 2 and 3
- 3 and 4
- 4 and 5
- 5 and 1

If `e_i` and `e_j` are the ids of two adjacent employees, the conditional expression we have to evaluate is:

1	$(e_i + e_j) \% 2 == 0$
---	-------------------------

The use of the term `any` resulted in some confusion. But the first public test case should have given you the idea that every pair of adjacent employee ids should sum to an even number. It is not enough if "some" pair of adjacent employee ids sums to an even number.

Problem-4

Question

Write a program to find which vowels are present in the input string. Print the vowels in lexical / dictionary order.

Input-1

```
1 | "Moon flowers bloom only at night, closing during the day."
```

Output-1

```
1 | aeiou
```

Input-2

```
1 | "1 Yard (yd) = 3 feet (ft)."
```

Output-2

```
1 | ae
```

Answer

```
1 | input_string = input().lower()
2 | vowels = ""
3 | if "a" in input_string:
4 |     vowels += "a"
5 | if "e" in input_string:
6 |     vowels += "e"
7 | if "i" in input_string:
8 |     vowels += "i"
9 | if "o" in input_string:
10 |    vowels += "o"
11 | if "u" in input_string:
12 |    vowels += "u"
13 | print(vowels)
```

Test Cases

Public

Input-1

```
1 | "Moon flowers bloom only at night, closing during the day."
```

Output-1

```
1 | aeiou
```

Input-2

```
1 | "1 Yard (yd) = 3 feet (ft)."
```

Output-2

```
1 | ae
```

Private

Input-1

```
1 | Education
```

Output-1

```
1 | aeiou
```

Input-2

```
1 | Summer is coming
```

Output-2

```
1 | eiou
```

Input-3

```
1 | Python politely said hi to JAVA
```

Output-3

```
1 | aeio
```

Input-4

```
1 | bcd xyz pqrs
```

Output-4

```
1 |
```

Solution

```
1 input_string = input().lower()
2 vowels = ""
3 if "a" in input_string:
4     vowels += "a"
5 if "e" in input_string:
6     vowels += "e"
7 if "i" in input_string:
8     vowels += "i"
9 if "o" in input_string:
10    vowels += "o"
11 if "u" in input_string:
12    vowels += "u"
13 print(vowels)
```

In line-1, we convert the string to lower case. We start with an empty string named `vowels` in line-2. Every time a vowel is found, we concatenate it to the end of `vowels`. The order in which the vowels are checked is: `a, e, i, o, u`. Since we are checking them in alphabetical order, the output will also be in alphabetical order.

Problem-5

Question

You are given the date of birth of two persons, not necessarily from the same family. The task is to find the younger of the two. If both of them share the same date of birth, then print the person whose name comes first in alphabetical order. If both the names starts with the same letter then compare the next letter of the names and so on.

Input-Output

Specification

The input will have four lines. The first two lines correspond to the first person, while the second two lines correspond to the second person. The first line for both the persons contains the name and the second line contains the date of birth in "DD-MM-YYYY" format.

The output will be the name of the younger of the two.

Examples

Input-1

```
1 | Harsh
2 | 10-03-1990
3 | Sparsh
4 | 18-12-1987
```

Output-1

```
1 | Harsh
```

Input-2

```
1 | Harsh
2 | 18-01-2000
3 | Sparsh
4 | 18-03-2000
```

Output-2

```
1 | Sparsh
```

Answer

```
1 | c1_name = input()      # name of the first person
2 | c1_dob = input()        # dob of the first person
3 | c2_name = input()      # name of the second person
4 | c2_dob = input()        # dob of the second person
5 |
6 | # extract the day, month and year from the date of birth
7 | # we can do this by slicing the dob string
8 | ##### person-1
```

```

9  c1_dob_day, c1_dob_month, c1_dob_year = int(c1_dob[: 2]), int(c1_dob[3 : 5]), int(c1_dob[6: ])
10 ##### person-2
11 c2_dob_day, c2_dob_month, c2_dob_year = int(c2_dob[: 2]), int(c2_dob[3 : 5]), int(c2_dob[6: ])
12
13 younger = ''      # variable to store name of younger person
14 # first check for year
15 if c1_dob_year > c2_dob_year:
16     younger = c1_name    # clearly, c1 was born after c2
17 elif c2_dob_year > c1_dob_year:
18     younger = c2_name    # clearly, c2 was born after c1
19 # they share the year of birth
20 else:
21     # check for month; same logic applies
22     if c1_dob_month > c2_dob_month:
23         younger = c1_name
24     elif c2_dob_month > c1_dob_month:
25         younger = c2_name
26 # they share the month of birth; same logic applies
27 else:
28     # check for day
29     if c1_dob_day > c2_dob_day:
30         younger = c1_name
31     elif c2_dob_day > c1_dob_day:
32         younger = c2_name
33     # if all the above conditions fail
34     # then they were born on the same day
35     # this is the moment we have been waiting for
36     # check for alphabetical order
37     else:
38         if c1_name < c2_name:
39             younger = c1_name    # c1 comes before c2 in alphabetical
order
40         else:
41             younger = c2_name    # c2 comes before c1 in aplhabetical
order
42 print(younger)

```

Answer-2

```

1 name1 = input()
2 dob1 = input()
3 name2 = input()
4 dob2 = input()
5
6 # changing the dob to YYYY-MM-DD format
7 dob1 = dob1[6:]+ '-' +dob1[3:5]+ '-' +dob1[:2]
8 dob2 = dob2[6:]+ '-' +dob2[3:5]+ '-' +dob2[:2]
9
10 if dob1 > dob2:
11     print(name1)
12 elif dob1 < dob2:
13     print(name2)
14 else:
15     if name1 < name2:
16         print(name1)

```

```
17     else:  
18         print(name2)  
19
```

Test Cases

Public

Input-1

```
1 Harsh  
2 10-03-1990  
3 Sparsh  
4 18-12-1987
```

Output-1

```
1 | Harsh
```

Input-2

```
1 Harsh  
2 18-01-2000  
3 Sparsh  
4 18-03-2000
```

Output-2

```
1 | Sparsh
```

Private

Input-1

```
1 Neha  
2 01-01-2001  
3 Priya  
4 01-01-2001
```

Output-1

```
1 | Neha
```

Input-2

```
1 David  
2 29-02-2004  
3 Jerold  
4 28-02-2004
```

Output-2

1 | David

Input-3

1	Abdul
2	05-07-2001
3	Raja
4	05-09-2001

Output-3

1	Raja
---	------

Input-4

1	Srivatsan
2	31-01-1932
3	Guru
4	19-11-2001

Output-4

1	Guru
---	------

Solution

Please check the comments associated with the code in answer-1.

Problem-6

Question

Write a python to validate the password given by the user based on the following conditions

1. It should be at least 8 and at most 32 characters
2. It should start with an uppercase or lowercase alphabet
3. It can contain numbers
4. It can contain the characters !, @, #, \$, %, ^, &, _, *, and .
5. It should not have the characters /, \, =, ', " and spaces

Answer

```
1 pw, valid = input(), False # initialize valid to False
2 # first condition is being checked here
3 if 8 <= len(pw) <= 32:
4     # second condition is being checked here
5     if 'a' <= pw[0] <= 'z' or 'A' <= pw[0] <= 'Z':
6         # fifth condition is being checked here
7         if not('/' in pw or '\\\\' in pw or '=' in pw or '!"' in pw or '\\"' in
8             pw or ' ' in pw):
9             valid = True
10    print(valid)
11    # Note that we are not checking for conditions 3 and 4
12    # This is because they are positive conditions
13    # They talk about what the password can contain
14    # Conditions 1, 2 and 5 are constraints
15    # They tell us what should not be there or what is not allowed
16    # So, it is these that we must be concerned with
```

Test Cases

Public

Input	Output
abcd1234	True
pass/word	False

Private

Input	Output
no_password	True
pythonIsSlowerInCompilationThanCPlusPlusandJava	False
rajesh'sBikeisBlack	False
never say never	False
giveA"Quote"	False

Solution

Check comments in answer.

Week-3 Practice Assignment (Theory)

Week-3 Practice Assignment (Theory)

Problem 1

Question

Answer

Solution

Tags

Problem 2

Question

Answer

Solution

Tags

Problem 3

Question

Answer

Solution

Tags

Problem 4

Question

Answer

Solution

Tags

Problem 5

Question

Answer

Solution

Tags

Problem 6

Question

Answer

Solution

Tags

Problem 7

Question

Answer

Solution

Tags

Problem 8

Question

Answer

Solution

Problem 9

Question a

Answer

Solution

Question b

Answer

Solution

Problem 1

Question

`x` is a variable of type `float` that has already been defined. Which of the following options could be the output of the following code-block? It is a Multiple Select Question (MSQ).

```
1 | print(f'{x:1.2f}')
```

- (a) 1.2
- (b) 1.29
- (c) 1234.20
- (d) 1234.234

Answer

(b), (c)

Solution

```
1 | print(f'{x:1.2f}')
```

The specification is that there must be two places after the decimal point.

- If the variable `x` has more than two places after the decimal point, then it will be rounded off to exactly two places.
- If the variable `x` has less than two places after the decimal point, then zeros will be appended at the end so that there are exactly two places after the decimal point.

Only options (b) and (c) conform to this specification.

Tags

basicio

Problem 2

Question

What does the following code-block print?

```
1 | n = int(input("Enter any positive number: "))
2 | for i in range(1,11):
3 |     print(n * i)
```

- (a) First 11 multiples of `i`.
- (b) First 10 multiples of `i`.
- (c) First 10 multiples of `n`.
- (d) First 11 multiples of `n`.

Answer

(c).

Solution

`n` stores the input number. `range(1, 11)` generates numbers from 1 to 10 and assigns each number to `i` in every iteration. In every iteration the value of `(n * i)` gets printed.

Tags

for loop.

Problem 3

Question

If the while loop of below code executes, when does it terminate?

```
1 | x = int(input("Enter any number: "))
2 | while(x % 5 != 0 and x % 10 != 0):
3 |     x = int(input("Enter any number: "))
4 | print("outside loop, the value of x is ", x)
```

- (a) Never terminates it is a infinite loop.
- (b) when we input a number which is multiple of 10.
- (c) when we input a number which is multiple of 5.
- (d) when we input a number which is not a multiple of both 5 and 10.

Answer

(b), (c)

Solution

while loop terminates if the condition `x % 5 != 0 and x % 10 != 0` becomes `False` it happens only when we input a number which is multiple of 5. (multiple of 10 is always a multiple of 5).

Tags

while loop.

Problem 4

Question

What does `average` represent at the end of execution?

```
1 total = 0
2 count = 0
3 for i in range(1000):
4     if i % 2 != 0 and count <= 50:
5         total = total + i
6         count = count + 1
7 average = total / count
8 print(average)
```

- (a) average of 1 to 1000 numbers
- (b) average of first 1000 odd numbers
- (c) average of first 50 odd numbers
- (d) average of first 51 odd numbers

Answer

(d)

Solution

There are 1000 iterations in the for-loop. The loop increments `total` and `count` by `i` and `1` respectively in each iteration when `i` is an odd number and `count` is less than or equal to 50.

Tags

for loop

Problem 5

Question

What does the printed value represents at the end of execution?

```
1 | n = int(input("Enter any positive number: "))
2 | i = 1
3 | while (i <= n):
4 |     if (n % i == 0):
5 |         print(i)
6 |     i = i + 1
```

- (a) Multiples of `i`
- (b) Factors of `i`
- (c) Factors of `n`
- (d) Multiples of `n`

Answer

- (c) Factors of `n`

Solution

If a number `n` is divisible by the number `i`, then `i` is called a factor of `n`. When `i` is a factor of `n`, then `n % i == 0`.

Tags

Problem 6

Question

What is the output of the following code

```
1 total = 0
2 for i in range(1, 5):
3     for j in range(i):
4         total = total + i
5 print(total)
```

(a) 4

(b) 30

(c) 10

(d) 20

Answer

(b)

Solution

- In the first iteration of outer loop $i = 1$, the inner loop iterates 1 time and the total is 1
- In the second iteration of outer loop $i = 2$, the inner loop iterates 2 times and the total is 5
- In the third iteration of outer loop $i = 3$, the inner loop iterates 3 times and the total is 14
- In the fourth iteration of outer loop $i = 4$, the inner loop iterates 4 times and the total is 30

Tags

nested for loop

Problem 7

Question

How many asterisks does the following code print

```
1 | i = -3
2 | while (i <= 13):
3 |     print('**')
4 |     print('*')
5 |     i = i + 1
```

- (a) 51
- (b) 25
- (c) 39
- (d) 48

Answer

- (a)

Solution

In every iteration 3 asterisks will be printed, 17 iterations happens, hence $17 * 3 = 51$ asterisks will be printed.

Tags

while loop.

Problem 8

Question

Which is the correct option that can be used in order to add 'n-1' blank spaces after a given string T?

Note: Variable T is a single character string.

- (a) `print("%ns"%T)`
- (b) `print("-ns"%T)`
- (c) `print("%-ns"%T)`
- (d) `print("-ns"%T)`

Answer

- (c) `print("%-ns"%T)`

Solution

For any string T, `print("%ns"%T)` format method adds `(n-len(T))` blank space to the right of T if n is a negative integer and adds `n-len(T)` blank space to the left of T if n is a positive integer. `n-len(T)` should be greater than 0.

According to the problem statement length of T is 1 so, option (c) is correct to add `n-1` blank spaces after a given string T.

Problem 9

Question a

You are an analyst in a Finance company. You are given a job to print the daily transaction in below format.

```
1 | country_code, currency_code, exchange_rate
```

Select the options that prints transaction records as in the sample output for input given in the table below. Multiple options can be correct (MSQ).

Input

Variable	VALUE
country_code	IN
currency_code	RS
exchange_rate	73.2272

Output

```
1 | IN, RS, 73.23
```

- (a) `print(country_code, currency_code, exchange_rate, sep = ", ")`
- (b) `print(f"{country_code}, {currency_code}, {exchange_rate:2.2f}")`
- (c) `print("{} , {} , {:.2f}".format(country_code, currency_code, exchange_rate))`
- (d) `print("{0} , {1} , {2:2.2f}".format(country_code, currency_code, exchange_rate))`
- (e) `print("{a} , {b} , {c:.2f}".format(a = country_code, b = currency_code, c = exchange_rate))`
- (f) `print("%s , %s , %2.2f%(country_code, currency_code, exchange_rate))`

Answer

(b), (c), (d), (e), (f)

Solution

Since, there is no format string used to round the float value `73.2272` to 2 decimal places, option (a) is incorrect. Option (b), (c), (d), (e), (f) are correct. These options use format string `.2f` to approximate `exchange_rate` to 2 decimal places.

Question b

In continuation to the last question you are also asked to capture the date and time of transaction in the below format.

```
1 | day-month-year hour:minute:second
```

Select the options that prints transaction records as in the sample output for input given in the table below. Multiple options can be correct (MSQ).

Input

variable	VALUE
day	01
month	01
year	2021
hour	12
minute	30
second	00

Output

```
1 | 01-01-2021 12:30:00
```

(a)

```
1 | print(day, month, year, sep = "-", end = " ")
2 | print(hour, minute, second, sep = ":" )
```

(b) `print(f"{day}-{month}-{year} {hour}:{minute}:{second}")`

(c) `print("{}-{}-{} {}:{}:{}".format(day, month, year, hour, minute, second))`

(d) `print("{0}-{1}-{2} {3}:{4}:{5}".format(day, month, year, hour, minute, second))`

(e) `print("{a}-{b}-{c} {d}:{e}:{f}".format(a = day, b = month, c = year, d = hour, e = minute, f = second))`

(f) `print("%s-%s-%s %s:%s:%s%(day, month, year, hour, minute, second))`

Answer

(a), (b), (c), (d), (e), (f)

Solution

All options are valid and prints the expected output.

Week-3 Graded Assignment (Theory)

Week-3 Graded Assignment (Theory)

Problem 1

Question

Answer

Solution

Problem 2

Question

Answer

Solution

Problem 3

Question

Answer

Solution

Problem 4

Question

Answer

Solution

Problem 5

Question

Answer

Solution

Problem 6

Question

Answer

Solution

Problem 7

Question

Answer

Solution

Problem 8

Question

Answer

Solution

Problem 1

Question

What does the following code block print?

```
1 | for i in 'We are in question one':
2 |     if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u':
3 |         continue
4 |     print(i, end = '')
```

- (a) w r n qstn n
- (b) wrnqstnn
- (c) we are in question one
- (d) None of the above

Answer

- (a)

Solution

The `for` loop checks whether each character is a vowel or not. If the character is a vowel ('a', 'e', 'i', 'o', 'u'), the character is skipped and not printed on the console due to `continue` statement.

Problem 2

Question

Is it possible to get the below output without using any loop and without repeating the string `Hello Python!` in Python?

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

- (a) True
- (b) False

Answer

- (a) True

Solution

Multiplying a string with a positive number `n` is equivalent to the string repeated `n` times one after another.

```
1 | print(10*'Hello Python!\n')
```

Please find the below code for Question (3 & 4)

```
1 | x = int(input())
2 | i = 0
3 | while x % 10**i != x:
4 |     i = i + 1
```

Problem 3

Question

What will the variable `i` represent at the end of execution where `x` is a positive integer?

- (a) Number of zeros in `x`
- (b) Number of ones in `x`
- (c) Number of digits in `x`
- (d) Number of non-zero digits

Answer

- (c) Number of digits of `x`

Solution

The above code is equivalent to the below

```
1 | x = int(input())
2 | i = 0
3 | j = x % 10**i
4 | while j != x:
5 |     i = i + 1
6 |     print("i = ", i)
7 |     j = x % 10**i
```

The remainder of `x` is checked on each iteration. `x` is divided by 1, 10, 100,.. in each iteration, where the remainder of `x` will be last 1, 2, 3,.. digits respectively. When the divisor `10 ** i` becomes greater than `x`, the remainder `x % 10**i` will be same as `x`.

The variable, `i` counts the number of times the power of `10` is increased, which in other terms is the number of digits in `x`.

For example, `x` is 1987 then loop exits at `i = 4` when the expression `1987 % 10**4` gives the value of `x`.

Problem 4

Question

What will be output if a negative value is given as input ?

- (a) Number of digits in `x`
- (b) Number of digits in `x - 1`
- (c) Number of digits in `x + 1`
- (d) Infinite loop

Answer

- (d) Infinite loop

Solution

If `x` holds the negative integral value, the expression `x % 10**i` can be expressed as `(-a * 10**i + b) % 10**i` where `a` is `x // 10**i` and `b` is the remainder. The remainder `b` is always a positive integer and smaller than the absolute value of `x`. Hence, the remainder will never be equal to `x`, and the loop continues infinitely.

For example, if `x` is equal to -10 then `x % 6` will be expressed as `(-2 * 6 + 2) % 6` which gives the remainder as 2.

Problem 5

Question

How many times do the break statements get executed? It is a Numerical Type Question (NAT).

```
1 | for i in range(10):
2 |     for j in range(10):
3 |         break
4 |     break
```

Answer

2

Solution

First `break` statement (in line-3) makes the program to exit from the inner loop regardless of how many iterations remaining. Hence, first `break` will exits from the inner for-loop and next `break` (in line-4) will make the program to exit from the outer for-loop. Thus, `break` is executed two times in the above code snippet.

Problem 6

Question

```
1 | for i in range(10, 0, 1):  
2 |     print(i)
```

A programmer wants to print a decreasing sequence. How many times does the `print` statement get executed? And why?

Select the most appropriate statement

- (a) One time because `i` takes only the value 10 and thereafter it will be decremented
- (b) One time because `i` takes only the value 9 and thereafter it will be decremented
- (c) `print` statement will not be executed due to invalid end points
- (d) `print` statement will not be executed due to incompatible step size

Answer

- (d) `print` statement will not be executed due to incompatible increment

Solution

The `start`, `end` and `step` parameters in `range()` are 10, 0 and 1. The variable `i` starts from 10. It should be incremented at each iteration by 1 and should end at 0, which is not possible. The `range(10, 0, 1)` returns no values, so the loop does not run and therefore `print` statement will not be executed.

Problem 7

Question

```
1 | for i in range(1231, -12420, -7):  
2 |     print(i)
```

How many times the `print` statement get executed? It is a Numerical Type Question (NAT).

Answer

1951

Solution

The variable `i` starts at 1231 and ends at -12419. This can be counted using another variable say `c`, as shown in the below code:

```
1 | c = 0  
2 | for i in range(1231, -12420, -7):  
3 |     c += 1  
4 |     print(i)  
5 | print(c)
```

The final printed value will give the number of times the `print` get executed.

An alternate approach will be to use `math.ceil()` function.

```
1 | import math  
2 | print(math.ceil((-12420-1231)/-7))
```

Problem 8

Code-1

```
1 | x = 0
2 | x_ = 1
3 | for i in range(10):
4 |     x, x_ = x_, x + x_
5 | print(x)
```

Code-2

```
1 | x = 0
2 | x_ = 1
3 | for i in range(10):
4 |     x = x_
5 |     x_ = x + x_
6 | print(x)
```

Question

Code-1 and Code-2 will return the same value.

- (a) True
- (b) False

Answer

- (b) False

Solution

In Code-1 the value of `x` and `x_` are assigned simultaneously from `x_` and `x + x_` respectively. Thus, both variable store different values, the pattern leads to the Fibonacci series since the initial value are 0 and 1 for `x` and `x_`.

In Code-2 the value of `x_` is assigned to `x` and `x_` is assigned with the value of expression `x + x_` which is nothing but the twice the value of `x`. Hence, it prints the sequence of value powers of 2.

Week-3 Practice Assignment (Programming)

Week-3 Practice Assignment (Programming)

Problem 1

 Question

 Answer

 Testcases

 Public

 Private

 Solution

Problem 2

 Question

 Answer 1

 Answer 2

 Testcases

 Public

 Private

 Solution

Problem 3

 Question

 Answer

 Testcases

 Public

 Private

 Solution

Problem 4

 Question

 Answer

 Testcases

 Public

 Private

 Solution

 Tags

Problem 5

 Question

 Answer

 Testcases

 Public

 Private

 Solution

 Tags

Problem 6

 Question

 Answer

 Testcases

 Public

 Private

 Solution

 Tags

Problem 1

Question

Write a program to print the numbers which are divisible by 12 and 13 in range of [1000 - 2000]

Answer

```
1 | for i in range(1000, 2001, 1):
2 |     if (i % 12 == 0 and i % 13 == 0):
3 |         print(i)
```

Testcases

Public

Output

```
1 | 1092
2 | 1248
3 | 1404
4 | 1560
5 | 1716
6 | 1872
```

Private

Output

```
1 | 1092
2 | 1248
3 | 1404
4 | 1560
5 | 1716
6 | 1872
```

Solution

The range from 1000 to 2000 can be expressed as `range(1000, 2001, 1)`. For the inclusion of 2000, the end value of the range function is given as 2001. For each `i` the if-statement checks for the divisibility of 12 and 13. If that is satisfied, the number will be printed.

Problem 2

Question

Write a program to find the sum of the digits of the number got from the user.

Input	Output
123456	21
67127	23
182638	28

Answer 1

```
1 n = int(input())
2 total = 0
3 while(n > 0):
4     total = total + n % 10
5     n = n // 10
6 print(total)
```

Answer 2

```
1 n = input()
2 total = 0
3 for i in n:
4     total = total + int(i)
5 print(total)
```

Testcases

Public

Input	Output
123456	21
67127	23
182638	28

Private

Input	Output
000000	0
111111	6
1996	25

Solution

The sum of all digits can be obtained by adding the digits from the last to the variable `total` using the modulo division by 10 and replacing the `n` by the quotient on division by 10. This will occur repeatedly when the `n` is greater than 0.

```
1 n = int(input())
2 total = 0
3 while(n > 0):
4     total = total + n % 10
5     n = n // 10
6 print(total)
```

For example:

if `n` = 1234

Iteration	<code>n > 0</code>	<code>n</code>	<code>n % 10</code>	<code>n // 10</code>	<code>total</code>
1	True	1234	4	123	4
2	True	123	3	12	7
3	True	12	2	1	9
4	True	1	1	0	10
5	False				

Another straight forward approach would be using a for-loop to iterate over the string input for each character and convert each character to integer and add that to `total`.

```
1 n = input()
2 total = 0
3 for i in n:
4     total = total + int(i)
5 print(total)
```

Problem 3

Question

Write a program to find sum of all prime numbers between 1 to `n`, where `n` is a positive integer from user.

Input	Output
10	17
100	1060

Answer

```
1 total = 0
2 n = int(input())
3 if (n == 0 or n == 1):
4     total = 0
5 if (n == 2):
6     total = 2
7 for i in range(2, n):
8     for j in range(2, i):
9         if (i % j == 0):
10            break
11     else:
12         total = total + i
13 print(total)
```

Testcases

Public

Input	Output
10	17
100	1060

Private

Input	Output
124	1593
0	0
1	0

Solution

Any number which is divisible by 1 and itself alone are called prime number. Here, we have to find all prime numbers between 2 and `n`, `n` is the positive integer got from the user. If `n` is `0` or `1` then we are printing directly the value `0` to the screen using the if-statement.

```
1 | if (n == 0 or n == 1):
2 |     total = 0
3 | #...
4 | #...
5 | print(total)
```

Otherwise, a for-loop with variable `i` is used to iterate between 2 and `n`, where the divisibility of `i` by any number less than `n` is checked by the if-statement. If `i` is divisible by any previous number of `i` from 2, then that `i` is exempted from adding to the variable `total` which will hold the sum of prime of numbers up to `n` at the end of the execution using the `flag` variable.

```
1 | for i in range(2, n):
2 |     flag = True
3 |     for j in range(2, i):
4 |         if (i % j == 0):
5 |             flag = False
6 |             break
7 |     if flag:
8 |         total = total + i
9 | print(total)
```

Problem 4

Question

Write a python program to print the following pattern. The number of `*` in the first line taken from the user input (odd number). The example is given for the user input `15`.

```
1 *****
2 ** * *
3 * * * *
4 * * * * *
5 * * * * *
6 * * * * *
7 * * * *
8 *****
9 * * *
10 * * * *
11 * * * *
12 * * * *
13 * * *
14 ** *
15 *****
```

Answer

```
1 n = int(input())
2 i = 0
3 while i < n:
4     j = 0
5     while j < n:
6         if i==j or i+j==n-1 or j == n//2 or i == n//2 or i == 0 or i == n-1
7             or j == 0 or j == n-1:
8             print('*', sep='', end='')
9         else:
10            print(' ', sep='', end='')
11     j = j + 1
12 i = i + 1
13 print()
```

Testcases

Public

Private

Input 1

```
1 | 13
```

Output 1

```
1 | *****
2 | **      *      **
3 | *  *      *      *  *
4 | *  *      *      *  *
5 | *  *      *      *  *
6 | *      ***      *
7 | *****
8 | *      ***      *
9 | *  *      *      *  *
10 | *  *      *      *  *
11 | *  *      *      *  *
12 | **      *      **
13 | *****
```

Input 2

```
1 | 5
```

Output 2

```
1 | *****
2 | *****
3 | *****
4 | *****
5 | *****
```

Input 3

```
1 | 3
```

Output 3

```
1 | ***
2 | ***
3 | ***
```

Input 4

```
1 | 1
```

Output 4

```
1 | *
```

Input 5

```
1 | 9
```

Output 5

```

1 *****
2 ** * *
3 * * * * *
4 * *** * *
5 *****
6 * *** *
7 * * * * *
8 ** * **
9 *****

```

Solution

The above pattern printed using a nested loop where the outer-loop for line and inner-loop for the character printed on the line.

For each line `*` has to be printed in some pattern which is determined by the if-statement. If any condition returns True `*` is printed, otherwise " " (space) is printed.

Condition	Satisfies
<code>i==j</code>	Line from top left to bottom right
<code>i+j==n-1</code>	Line from bottom left to top right
<code>i == 0</code>	Top horizontal line
<code>j == n//2</code>	Middle horizontal line
<code>i == n-1</code>	Bottom horizontal line
<code>j == 0</code>	Left vertical line
<code>i == n//2</code>	Middle vertical line
<code>j == n-1</code>	Right vertical line

Tags

Problem 5

Question

Write a python program to print all the combination that satisfies $x^4 + y^3 = z^2$.

Where,

1. x, y and z should be distinct, positive and less than n obtained from the user.
2. $x < y < z$
3. The output should be printed in the ascending order of x

Hint: Use nested loop

Answer

```
1 n = int(input())
2 for i in range(1, n):
3     for j in range(i+1, n):
4         for k in range(j+1, n):
5             if i**4 + j**3 == k**2:
6                 print(i, j, k)
```

Testcases

Public

Input 1

```
1 | 100
```

Output 1

```
1 | 1 2 3
2 | 5 6 29
3 | 6 9 45
4 | 7 15 76
```

Input 2

```
1 | 10
```

Output 2

```
1 | 1 2 3
```

Private

Input 1

```
1 | 73
```

Output 1

1	1	2	3
2	5	6	29
3	6	9	45

Input 2

1	200
---	-----

Output 2

1	1	2	3
2	5	6	29
3	6	9	45
4	7	15	76
5	8	32	192
6	9	27	162

Solution

Here a triple nested loop is required for the three variables `i`, `j` and `k`. In the outermost for-loop, `i` takes the value from the range of 1 to `n`. In the intermediate for-loop, the variable `j` takes the value from `i+1` to `n`. In the innermost for-loop, `k` takes the value from `j+1` to `n`. These are to ensure the conditions that x, y, z should be distinct, and $x < y < z < n$. Within the innermost loop, the expression `i**4 + j**3 == k**2` is checked and printed if that is True.

```
1 for i in range(1, n):
2     for j in range(i+1, n):
3         for k in range(j+1, n):
4             if i**4 + j**3 == k**2:
5                 print(i, j, k)
```

Tags

Problem 6

Question

Accept two strings from the user and remove all characters from the second string which are present in the first string.

Sample Input - 1

```
1 | aeiou
2 | this is python program
```

Sample Output - 1

```
1 | ths s pytn prgrm
```

Sample Input - 2

```
1 | lo
2 | hello python
```

Sample Output - 2

```
1 | he pytn
```

Sample Input - 3

```
1 | hello python
```

Sample Output - 3

```
1 | hello python
```

Answer

```
1 | string1 = input()
2 | string2 = input()
3 | temp = ''
4 | for i in range(0,len(string1)):
5 |     for j in range(0,len(string2)):
6 |         if (string1[i] == string2[j]):
7 |             continue
8 |         else:
9 |             temp = temp + string2[j]
10 |     string2 , temp = temp , ""
11 | print(string2)
```

Testcases

Public

Input 1

```
1 | aeiou  
2 | this is python program
```

Output 1

```
1 | ths s pytn prgrm
```

Input 2

```
1 | lo  
2 | hello python
```

Output 2

```
1 | he pytn
```

Input 3

```
1 | hello python
```

Output 3

```
1 | hello python
```

Private

Input 1

```
1 | abcdefghijkl  
2 | abcdmnop
```

Output 1

```
1 | mnop
```

Input 2

```
1 | @.  
2 | abc@onlinedegree.iitm.ac.in
```

Output 2

```
1 | abconlinedegreeitmacin
```

Input 3

```
1 | abcdef  
2 | ijklnm
```

Output 3

```
1 | ijklnm
```

Solution

- Accept two string from the user and assign to `string1` and `string2` variables.
- Initialize the `temp` variable with an empty string.
- In the first cycle of the outer loop (`i=0`) using a nested loop check for each index of `string2`, if the character of `string2` is not matched with character of `string1`, then appends this character of `string2` to `temp` string.
- In line 10, assign `temp` value to `string2` and assign an empty string to `temp` for the next cycle of the outer loop.
- So, in each cycle of outer loop, 1 character of `string1` will be removed from `string2`, then finally print the `string2` after the completion of loop.

Tags

Week-3 Graded Assignment (Programming)

Week-3 Graded Assignment (Programming)

Problem 1

Question [8]

Sample Input and Outputs

Answer

Testcases

Public

Private

Solution

Problem 2

Question [8]

Answer 1

Answer 2

Testcases

Public

Private $[2 + 2 + 2 + 2 = 8]$

Solution

Problem 3

Question [14]

Answer

Testcases

Public

Private

Solution

Problem 4

Question [6]

Answer

Testcases

Public

Private $[2 + 2 + 2 = 6]$

Solution

Problem 5

Question [14]

Answer

Testcases

Public

Private

Solution

Problem 1

Question [8]

Write a program to find the sum of the following series up to n terms

$$1 + (1 + 2) + (1 + 2 + 3) + (1 + 2 + 3 + 4) + \dots \dots \dots n \text{ terms}$$

Sample Input and Outputs

Input	Output
3	10
5	35

Answer

```
1 n = int(input())
2 total = 0
3 for i in range(1, n + 1):
4     for j in range(1, i+1):
5         total = total + j
6 print(total)
```

Testcases

Public

Input	Output
3	10
5	35

Private

Input	Output	Weight
0	0	1
10	220	3
234	2162940	3
1	1	1

Solution

The input from the user is parsed into an integer datatype and stored in the variable `n`. `total` is a variable which stores the required output which initialized as 0.

The required sum of the sequence can be mathematically expressed as,

$$1 + (1 + 2) + (1 + 2 + 3) + (1 + 2 + 3 + 4) + \dots n \text{ terms} = \sum_{i=1}^n \sum_{j=1}^i j$$

The above mathematical expression can be written as a nested for-loop of `range(1, n + 1)` and `range(1, i + 1)` respectively. The sum is stored in the variable `total`.

```
1 | for i in range(1, n + 1):
2 |     for j in range(1,i + 1):
3 |         total = total + j
```

Problem 2

Question [8]

Write a program to find all the prime factors of a number which taken as input from user. Assume the accepted input as positive integer which is greater than or equal to 2.

Sample Input 1

```
1 | 15
```

Sample Output 1

```
1 | 3  
2 | 5
```

Sample Input 2

```
1 | 79
```

Sample Output 2

```
1 | 79
```

Sample Input 3

```
1 | 78
```

Sample Output 3

```
1 | 2  
2 | 3  
3 | 13
```

Answer 1

```
1 n = int(input())  
2 if n == 2:  
3     print(2)  
4 if n > 2:  
5     for i in range(2, n+1):  
6         if (n % i == 0):  
7             flag = True  
8             for j in range(2, i):  
9                 if (i % j == 0):  
10                    flag = False  
11                    break  
12            if flag:  
13                print(i)
```

Answer 2

```
1 n = int(input())
2 if n == 2:
3     print(2)
4 if n > 2:
5     for i in range(2, n+1):
6         if (n % i == 0):
7             for j in range(2, i):
8                 if (i % j == 0):
9                     break
10            else:
11                print(i)
```

Testcases

Public

Input 1

```
1 | 15
```

Output 1

```
1 | 3
2 | 5
```

Input 2

```
1 | 79
```

Output 2

```
1 | 79
```

Input 3

```
1 | 78
```

Output 3

```
1 | 2
2 | 3
3 | 13
```

Private [2 + 2 + 2 + 2 = 8]

Input 1

```
1 | 1234
```

Output 1

```
1 | 2  
2 | 617
```

Input 2

```
1 | 6217
```

Output 2

```
1 | 6217
```

Input 3

```
1 | 517
```

Output 3

```
1 | 11  
2 | 47
```

Input 4

```
1 | 2
```

Output 4

```
1 | 2
```

Solution

The prime factors for the given number is the prime numbers which can be divide the given number.

The prime number starts with 2 and keeps going on. So the prime factor for the number 2 is 2, which is directly printed using the if-statement.

```
1 | if n == 2:  
2 |     print(2)
```

If the number from the user is greater than 2, then the control goes to `for` loop with `i` of `range(2, n + 1)` (line-5) which starts at 2 and ends at `n`.

When any `i` divides the input number `n`, we verify if it is a prime number using the nested `for` loop (line-7 onwards).

Problem 3

Question [14]

Accept a string as an input from the user, convert this to lower case and print the string in ascending order using loop.

Note: Ignore non-alphabetic characters in the output

Sample Input 1

```
1 | Bharatanatyam
```

Sample Output

```
1 | aaaaabhmnrtyy
```

Sample Input 2

```
1 | monty python
```

Sample Output

```
1 | hmnooptyy
```

Answer

```
1 | text = input().lower()
2 | length = len(text)
3 | output_string = ""
4 | for char in "abcdefghijklmnopqrstuvwxyz":
5 |     for i in range(0, length):
6 |         if char == text[i]:
7 |             output_string += char
8 | print(output_string)
```

Testcases

Public

Input	Output
Bharatanatyam	aaaaabhmnrtyy
monty python	hmnooptyy

Private

Input	Output	Weight
bharatanatyam	aaaaabhmnrty	4
montypython	hmnooptyy	4
101 @#\$%&		3
James Bond Agent 007	aabdeegjmnnost	3

Solution

In the solution, we iterate over all the alphabets in the "English" language arranged in ascending order. In each iteration, we iterate over (nested iteration) each characters in the given input string and check if the alphabets matches any of the characters in the input string. If yes, we add them to the `output_string`, once iterations are finished the ordered characters sequence stored in this variable is printed.

Problem 4

Question [6]

Accept the string from the user and display the characters located on prime index (2,3,5,7..) of a given string.

Sample Input 1

```
1 | hellopython
```

Sample Output 1

```
1 | 1
2 | 1
3 | p
4 | t
```

Sample Input 2

```
1 | this is python program
```

Sample Output 2

```
1 | i
2 | s
3 | i
4 |
5 | h
6 | n
7 | o
8 | r
```

Answer

```
1 | st = input()
2 | length = len(st)
3 | count = 0
4 | for i in range(2, length):
5 |     count = 0
6 |     for j in range(2, i):
7 |         if (i % j == 0):
8 |             count = count + 1
9 |     if (count == 0):
10 |         print(st[i])
```

Testcases

Public

Input 1

```
1 | hellopython
```

Output 1

```
1 | 1  
2 | 1  
3 | p  
4 | t
```

Input 2

```
1 | this is python program
```

Output 2

```
1 | i  
2 | s  
3 | i  
4 |  
5 | h  
6 | n  
7 | o  
8 | r
```

Private [2 + 2 + 2 = 6]

Input 1

```
1 | abcdefghijklmnopqrstuvwxyz
```

Output 1

```
1 | c  
2 | d  
3 | f  
4 | h  
5 | l  
6 | n  
7 | r  
8 | t  
9 | x
```

Input 2

```
1 | 0123456789
```

Output 2

```
1 | 2  
2 | 3  
3 | 5  
4 | 7
```

Input 3

```
1 | abcdefghijkl
```

Output 3

```
1 | c  
2 | d  
3 | f  
4 | h  
5 | l
```

Solution

Accept the input string from the user and assign it to variable `st`. Calculate the length of the string and assign it to variable `length`. Inside the `for` loop, we check each index in the string (`i` iterates from `2` to `length-1`), if it is a prime number then the character at the index `i` which is `st[i]`, is printed.

Problem 5

Question [14]

Accept phone number from user as string.

A valid phone number should satisfy the following constraints.

- The number should start with 6,7,8 and 9.
- Length of the number should be 10.
- Number should not have any digit repeated more than 5 times in consecutive manner.
- Number should not have any digit more than 7 times.

Input	Output
9852546666	valid
0984587558	invalid
85425@6558	invalid

Answer

```
1 number = input()
2 valid = True
3 if len(number) == 10 and number.isdigit() == True and number[0] in '6789':
4     for digit_index in range(0,5):
5         count = number.count(number[digit_index])
6         if count > 7:
7             valid = False
8             break
9         if 6*number[digit_index] in number:
10            valid = False
11            break
12     else:
13         valid = False
14     if valid == True:
15         print("valid")
16     else:
17         print('invalid')
```

Testcases

Public

Input	Output
9852546666	valid
0984587558	invalid
85425@6558	invalid

Private

Input	Output	Weight
8181818181	valid	3
587888888	invalid	2
8988899998	valid	3
9898999999	invalid	2
8989.89898	invalid	2
9999987999	invalid	2

Solution

- Accept phone number as a string from the user.
- Initialize a variable `valid` with bool value `True`.
- In line 3, check three conditions with the number. If any of them is False then assign `False` Value to `valid` in the else block.
- Otherwise check each digit one by one from 0th to 4th index because only 0th to 4th index digit can be repeated 6 times in 10 digit phone numbers.
- In line 6, check if the counting of any digit is more than 7 then assign `False` value to `valid` and break the loop.
- In line 9, check if the number has digit in consecutive manner 6 times then assign `False` value to `valid` and break the loop.
- In the last, according to the value of `valid`, print the output.

Week-4, Practice, Theory

Week-4, Practice, Theory

[Problem 1](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 2](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 3](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 4](#)

[Question](#)

[Answers](#)

[Solution](#)

[Problem 5](#)

[Question](#)

[Answer](#)

[Solution](#)

[Problem 6](#)

[Question](#)

[Answers](#)

[Solution](#)

Problem 1

Question

Match the statements with respective output in the below table.

Statement	OUTPUT
(a) <code>type(float("10.2"))</code>	(1) Returns <code>ValueError</code>
(b) <code>int("2.3")</code>	(2) Returns <code>str</code>
(c) <code>del x</code>	(3) Returns <code>Float</code>
(d) <code>type(str(100))</code>	(4) <code>SyntaxError</code>
(e) <code>bool('False')</code>	(5) Deletes variable <code>x</code> from memory if exists, otherwise throws <code>NameError</code>
(f) <code>"12345".index(1)</code>	(6) Returns <code>True</code>
(g) <code>from random import randint()</code>	(7) Returns <code>TypeError</code>

Select the correct option. This is a MCQ type question.

- (A) (a) - (1), (b) - (3), (c) - (5), (d) - (2), (e) - (6), (f) - (7), (g) - (4)
(B) (a) - (3), (b) - (1), (c) - (5), (d) - (2), (e) - (6), (f) - (7), (g) - (4)
(C) (a) - (3), (b) - (5), (c) - (1), (d) - (6), (e) - (2), (f) - (7), (g) - (4)
(D) (a) - (3), (b) - (5), (c) - (1), (d) - (2), (e) - (6), (f) - (4), (g) - (7)

Answer

(B)

Solution

(a) , (d) `type()` function gives the data type of the value passed to this function.

(b) `int()` function converts an integer in `str` form to `int` data type. To change a string of `float` value to an `int`, it should be first converted a `float` value using `float()` first and then `int()` should be applied. Hence, `ValueError` is shown to the user.

(c) `del` is used to delete a variable in python.

(e) `bool` function converts the value passed to it to a `bool` type. Since the value inside is a non empty `string`, it is converted to `True`.

(f) `index()` function returns the index of first occurrence of the character in the given string. Since, the string `'12345'` contains string `'1'` but does not contain the integer `1`, it throws `TypeError`.

(g) Correct syntax is: `from random import randint`, hence it shows `SyntaxError`.

Problem 2

Question

What is the output of the following code? The input is a string of odd length. This is a MSQ type question.

```
1 | input_string = input()
2 | x = input_string[len(input_string)//2]
3 | while x in input_string:
4 |     input_string = input_string[:-1]
5 |     print(x, end = "")
```

- (a) It prints a string where each character is the first character of `input_string`
- (b) It prints a string of the length `len(input_string) - input_string.index(x)`
- (c) It prints `input_string` in the reverse order
- (d) It prints a string where each character is the middle character of the `input_string`
- (e) It prints a string of almost same length as the `input_string`

Answer

(b), (d)

Solution

Option (b) and (d) are correct. Since the `input_string` is of odd length, the position `len(input_string)//2` indicates the middle index and `x` represents the character at this index. The `while` loop checks if `x` exists in the `input_string`. Inside the body of the loop, in each iteration `input_string` is trimmed from right by one character till the character stored in `x` is found in this trimmed `input_string`.

Hence, it prints a string where each character is the middle character of the `input_string`. The length of this string is `len(input_string) - input_string.index(x)`.

Problem 3

Question

Python has __ statement as a null statement. This is a "Fill in the blank" type question.

Answer

pass

Solution

The `pass` keyword is used to represent an empty / blank statement. It does not perform anything in the program. It is sometimes used as a placeholder statement so that real code can be filled later.

Problem 4

Question

`L` is a non-empty list of integers and `x` is an integer. Assume that both `L` and `x` have already been defined. The following code does not throw any error when executed. Lines 1 and 6 will be used to refer to the state of the variables before and after key sections of the code are executed.

```
1 pass          # before
2 count = 0
3 while len(L) > 0:
4     L.remove(x)
5     count += 1
6 pass          # after
```

If the value of `count` is 10 at the end of execution of the code (line-6), which of the following statements are true? It is a Multiple Select Question.

- (a) `x` is an element of `L` at line-1.
- (b) `L` has at least two different (unequal) elements in it at line-1
- (c) Length of `L` is 10 at line-1
- (d) Length of `L` is 0 at line-6

Answers

(a), (c), (d)

Solution

The first observation is that the loop goes through 10 iterations. This can be inferred from the value of `count`. Next, we know that the code doesn't throw an error. The only line which could introduce a bug in the code is line-4. Combining this with the first observation we see that `x` is certainly present in `L` at line-1. In each iteration, the element `x` is removed from `L`. This means that the element `x` occurs at least 10 times. Can it occur more number of times? No, then the value of `count` would have been more than 10. Can `L` have any other element in it other than `x` at line-1? This is also not a possibility as the list becomes empty after 10 elements are removed from it.

Problem 5

Question

`P` is a non-empty list of distinct integers that has already been defined. Which of the following statements are true at the end of execution of the code given below? It is a Multiple Select Question (MSQ).

```
1 Q, R = [ ], [ ]
2 for x in P:
3     Q.append(-x)
4 Q.sort()
5 for x in Q:
6     R.append(-x)
```

- (a) `P` is always equal to `R`, i.e., `P == R` returns the value `True`
- (b) Every element in `P` is present in `R`, but not necessarily in the same sequence.
- (c) `R` is a list of integers sorted in descending order.
- (d) `R` is a list of integers sorted in ascending order.

Answer

(b), (c)

Solution

The negative of each element in `P` is added to the list `Q`. Then `Q` is sorted in ascending order. Finally, the negative of each element in `Q` is added to `R`. This means that `R` will be a list sorted in descending order. Every element in `P` would be present in `R`. Let us take the simple example of `P = [5, 2, 4, 1, 3]`. Just before line-4 is executed, `Q` will be `[-5, -2, -4, -1, -3]`. After line-4 is executed, `Q` will be `[-5, -4, -3, -2, -1]`. After the entire code is executed, `R` will be `[5, 4, 3, 2, 1]`.

Problem 6

Question

Let `M` be a matrix of numbers that has already been defined and populated. We wish to find the sum of the elements in each row and store all such row-sums in a list called `rsum`. For the i^{th} row, `rsum[i]` should be the sum of all elements in that row. Select the correct code snippets that achieve this. It is a Multiple Select Question (MSQ).

(a)

```
1 dim = len(M)
2 rsum = []
3 for i in range(dim):
4     # one element for each row
5     rsum.append(0)
6     # we will now go through all cells in the ith row
7     for j in range(dim):
8         # rsum[-1] is the same as rsum[i] in this case
9         # add the jth element in the ith row to this sum
10        rsum[-1] += M[i][j]
```

(b)

```
1 dim = len(M)
2 for i in range(dim):
3     # this is WRONG; rsum should be initialized outside the nested loop
4     rsum = []
5     rsum.append(0)
6     for j in range(dim):
7         rsum[-1] += M[i][j]
```

(c)

```
1 dim = len(M)
2 rsum = []
3 for i in range(dim):
4     rsum.append(0)
5 for i in range(dim):
6     for j in range(dim):
7         # this is WRONG; rsum[j] should be rsum[i] or rsum[-1]
8         rsum[j] += M[i][j]
```

(d)

```
1 dim = len(M)
2 rsum = [ ]
3 # different from option (a)
4 # rsum is first initialized with zeros
5 # one zero for every row
6 for i in range(dim):
7     rsum.append(0)
8 for i in range(dim):
9     for j in range(dim):
10         rsum[i] += M[i][j]
```

Answers

(a), (d)

Solution

Check comments in the code.

Week-4, Graded Theory

Week-4, Graded Theory

Problem 1

Question

Answer

Solution

Problem 2

Question

Answer

Solution

Problem 3

Question

Answer

Solution

Problem 4

Question

Answer

Solution

Problem 5

Question

Answer

Solution

Problem 6

Question

Answer

Solution

Problem 7

Question

Answer

Solution

Problem 8

Question

Answer

Solution

Problem 9

Question

Answer

Problem 10

Question

Answer

Solution

Problem 1

Question

`A`, `B` and `C` are boolean variables. Which of the following boolean expressions are equivalent to `C`? This is a MSQ type question.

A	B	C
<code>False</code>	<code>False</code>	<code>False</code>
<code>False</code>	<code>True</code>	<code>True</code>
<code>True</code>	<code>False</code>	<code>False</code>
<code>True</code>	<code>True</code>	<code>True</code>

- (a) `B`
- (b) `(not(A) and B) or (A and B)`
- (c) `(A and not(A)) or B`
- (d) `((A or not(A)) and B)`
- (e) None of these

Answer

(a), (b), (c), (d)

Solution

All boolean expressions are equivalent. A user can follow any of the expressions and achieve the same result. There are various ways to transform expressions in one form to another, for example Sum of Products (SOP), Products of Sum (POS) etc. However, this simple exercise does not require these structured approach. It can be figured out by carefully observing the expressions.

The below code shows all the expressions results in the same output:

```
1 X = [[False, False, False],[False, True, True],[True, False, False],[True, True, True]]
2 for A, B, C in X:
3     print(B, (not(A) and B) or (A and B), ((A and not(A)) or B), ((A or not(A)) and B))
```

Problem 2

Question

Match the statements with respective output in the below table.

```
string = "this is old python course"
```

Statement	OUTPUT
(a) <code>string.count("is")</code>	(1) <code>True</code>
(b) <code>string.replace("old", "new")</code>	(2) <code>12</code>
(c) <code>string.index("python")</code>	(3) <code>'this is new python course'</code>
(d) <code>string.startswith("this")</code>	(4) <code>2</code>
(e) <code>string.isalnum()</code>	(5) <code>False</code>
(f) <code>string.title()</code>	(6) <code>'This Is Old Python Course'</code>
(g) <code>string.strip("this is old")</code>	(7) <code>'python course'</code>

Select the correct option. This is a MCQ type question.

- (A) (a) - (4), (b) - (3), (c) - (2), (d) - (1), (e) - (5), (f) - (6), (g) - (7)
- (B) (a) - (5), (b) - (3), (c) - (2), (d) - (1), (e) - (4), (f) - (7), (g) - (6)
- (C) (a) - (1), (b) - (3), (c) - (2), (d) - (5), (e) - (4), (f) - (6), (g) - (7)
- (D) (a) - (4), (b) - (2), (c) - (3), (d) - (1), (e) - (6), (f) - (5), (g) - (7)

Answer

(A)

Solution

- (a) `string.count("is")` counts the number of times substring `"is"` appears in `"this is old python course"`.
- (b) `string.replace("old", "new")` replaces `"old"` to `"new"` in `"this is old python course"`.
- (d) `string.startswith("this")` returns bool literal `True`, if the string starts with `"this"`.
- (e) `string.isalnum()` returns bool literal `True`, if the string contains only alphabets and numbers.
- (f) `string.title()` returns a string where the first character of each word is capitalized.
- (g) `string.strip("this is old")` removes the string value passed to it from beginning and end of the original string `"this is old python course"`.

Problem 3

Question

What should be the value of `a`, `b`, `c`, `d` to print the list of first 10 positive even numbers regardless of the order. This is a MSQ type question.

```
1 | for num in range(a, b, c):  
2 |     num = num + d  
3 |     print(num)
```

- (a) 2, 21, 2, 0
- (b) 20, 0, -2, 1
- (c) 0, 19, 2, 2
- (d) 22, 2, -2, -2
- (e) None of these

Answer

(a), (c), (d)

Solution

The `range(a, b, c)` gives an arithmetic sequence given below which starts at `a`, elements are incremented by value `c` till it reaches `b`. The endpoint `b` is not included.

```
1 | a, a + c, a + 2 * c, a + (n - 1) * c; where n is an arbitrary positive  
integer, the last value in this sequence should be less than `b`.
```

For $a = 0$, $b = 19$, $c = 2$, the resulting sequence is 0, 2, 4, 8, 10, 12, 14, 16, 18.

In for loop, at each iteration, the variable `num` is assigned the value from this sequence. Inside the body of the loop, `num` is incremented by 2 and printed.

In the very first iteration, `num` is set to 0. The value is then incremented by 2 inside the loop and the same is printed. On the next iteration, the loop variable `num` stores the second element from the sequence which is 2, this is incremented by 2 inside the body and printed on the console. This repeats for the rest of the iterations.

```
1 | a, b, c, d = 0, 19, 2, 2  
2 | for num in range(a, b, c):  
3 |     print("Loop variable:", num, end = ", ")  
4 |     num = num + d  
5 |     print("Updated value: ", num)
```

Problem 4

Question

What should be the value of `x`, if the following code prints `python` on the console. This is a MSQ type question.

```
1 | for num in x:  
2 |     print(num, end="")
```

- (a) `["python"]`
- (b) `["p", "y", "t", "h", "o", "n"]`
- (c) `"python"`
- (e) None of these

Answer

(a), (b), (c)

Solution

Option (a) prints the first element of the list, which is the string `python`. In option (b), all the 6 elements are printed one after another, therefore the text 'python' displayed on the console. In option (c), the characters are printed from the string `python` one after another. The string `python` behaves in the similar way as the list `["p", "y", "t", "h", "o", "n"]` here.

Problem 5

Question

A data structure is called heterogeneous if it can hold elements of different data types. Otherwise it is called homogeneous. Does a list need to be homogeneous?

- (a) Yes, it needs to be homogeneous
- (b) No, it need not be homogeneous

Answer

(b)

Solution

The `list` allows different data types as its element. Hence, it is a heterogeneous data type.

Problem 6

Question

Which of the following options are correct? This is a MSQ type question.

```
1 | fruit_list = ["apple", "banana", "orange", "apple", "pineapple" ]
```

- (a) `fruit_list.remove("apple")` removes the first occurrence (from the left) of the string "apple"
- (b) `fruit_list.remove("apple")` removes the last occurrence (from the left) of the string "apple"
- (c) The length of `fruit_list` becomes 4 after the execution of `fruit_list.remove("apple")`
- (d) The datatype of `fruit_list` is `int`
- (e) `fruit_list.append("apple")` removes the last occurrence (from the left) of the string "apple"

Answer

(a), (c)

Solution

`fruit_list.remove("apple")` removes the last occurrence of the string "apple". There were 5 elements in the list. The length of the list reduced to 4 after removing one element.

Problem 7

Question

You are given two lists:

```
1 | list_1 = ["I play", "You play"]
2 | list_2 = ["Badminton", "Cricket"]
```

Which of the following option(s) gives the below output. This is a MSQ type question.

```
1 | I play Badminton
```

- (a) `print(list_1[0] + " " + list_2[0])`
- (b) `print("{a} {b}".format(a = list_1[0], b = list_2[0]))`
- (c)

```
1 | for x in list_1[:-1]:
2 |     for y in list_2[:-1]:
3 |         print(x, y)
```

- (d) None of these

Answer

- (a), (b), (c)

Solution

All the above options are correct. The first element of the two lists `list_1` and `list_2` are accessed and `'I play Badminton'` is printed on the console.

Common data for problems 8 and 9

Assume that `L` is a non-empty list of positive integers. Also assume that the list is a distinct collection of numbers, i.e., no two numbers are alike. Consider the following code.

```
1 | s = 0
2 | for x in L:
3 |     s += x
4 |
5 | flag = False
6 | y = -1
7 | for x in L:
8 |     if x * len(L) == s:
9 |         flag = True
10 |        y = x
11 |        break
```

Problem 8

Question

If `flag` is `True` at the end of execution of the code given above, which of the following statements are true? Note that the options should be true for any list `L` that satisfies the conditions given in the common data. Multiple options could be correct.

- (a) `y` is an element in the list `L`
- (b) `y` is the smallest number in the list
- (c) `y` is the greatest number in the list
- (d) `y` is the average (arithmetic mean) of the numbers in the list
- (e) `y` is the element at index `len(L) // 2` in the list `L`

Answer

(a), (d)

Solution

The main idea behind the code is as follows. `s` holds the sum of the elements in the list `L`. Line-8 inside the `for` loop essentially amounts to this:

If each element in the list is replaced by `x`, will the sum of the elements still be equal to `s`?

If the answer to this question is yes, then the `flag` is changed to `True` and `y` is set to `x` and we break out of the loop. If the answer to this question is `False`, then we simply move onto the next iteration. At the end of execution, if `flag` remains `False`, then no such element `x` with the required characteristic is found.

Moving on, if `x * len(L) == s`, for some element `x` in the list `L`, then `y` is this element at the end of execution of the above code. Then we see that `y` is nothing but `s // len(L)`. We can use the floor division operator because `s` is divisible by `len(L)`. So, `y` is the average of the numbers in the list.

Problem 9

Question

Assume that `L` is a list of the first n positive integers, where $n > 0$. Under what conditions will the variable `flag` be `True` at the end of execution of the code given above?

- (a) `n` is an odd integer
- (b) `n` is an even integer

Answer

(a)

If `L` is a list of the first n positive integers then, $S = \frac{n(n+1)}{2}$. For `flag` to be true, there must

be some number x between 1 and n that satisfies the following equation:

$$x \cdot n = \frac{n(n+1)}{2}$$

We immediately see that $x = \frac{n+1}{2}$. Since the list contains only integers, n would have to be odd.

Problem 10

Question

Let `x` be a vector. We wish to compute the dot product of `x` with itself and store it in a variable called `dp`. Select the correct code snippet(s) from the options given below. It is a Multiple Select Question (MSQ).

(a)

```
1 dp = 0
2 # Dot product of a vector with itself
3 # is just the sum of squares of the
4 # components of the vector
5 for elem in x:
6     dp += elem * elem
```

(b)

```
1 dp = 0
2 # this is also a valid method
3 for i in range(len(x)):
4     dp += x[i] ** 2
```

(c)

```
1 dp = 0
2 # this is WRONG
3 # range(len(x)) is the right command
4 for i in range(x):
5     dp += x[i] * x[i]
```

(d)

```
1 i, dp = 0, 0
2 # this is WRONG
3 # it should be i < len(x)
4 while i <= len(x):
5     dp += x[i] ** 2
6     i += 1
```

(e)

```
1 i, dp = 0, 0
2 #
3 while i < len(x):
4     dp += x[i] ** 2
5     i += 1
```

Answer

(a), (b), (e)

Solution

Check the comments in the code.

Week-4, Practice, Programming

Week-4, Practice, Programming

Problem 1

 Question

 Answer

 Test Cases

 Public

 Private

Problem 2

 Question

 Answer

 Test Cases

 Public

 Private

Problem 3

 Question

 Answer

 Test Cases

 Public

 Private

Problem 4

 Question

 Answer

 Testcases

 Public

 Private

Problem 5

 Question

 Answer

 Test Cases

 Public

 Private

Problem 6

 Question

 Answer

 Test Cases

 Public

 Private

Note: Check the comments in the code for the solution.

Problem 1

Question

Accept electricity `units` as a positive integer from the user and write a program to print total bill amount according to the following criteria:

Units	Cost per unit (Rs)
0 to 100	0
101 to 200	5
201 to 500	8
501 and above	10

Answer

```
1 units = int(input())
2 bill = 0
3 #if units are greater than 500
4 if units > 500:
5     bill += 5*100 + 300*8 + (units-500)*10
6 #if units are 201 to 500
7 elif units > 200:
8     bill += 5*100 + (units-200)*8
9 #if units are 101 to 200
10 elif units > 100:
11     bill += 5*(units-100)
12 #if units are 0 to 100
13 else:
14     pass
15 print(bill)
```

Test Cases

Public

Input	Output
75	0
150	250
250	900

Private

Input	Output
0	0
200	500
300	1300
600	3900

Problem 2

Question

Write a program to accept a string from the user that contains `(,)`, `{, }` and `[,]` in it. Print `True` if all the brackets are opened and closed properly. Otherwise print `False`.

Note:

- `{ } [] ()` are the opening and closing brackets which needs to be verified - All the opening brackets should be closed with the same type of closing bracket.

Input	Output
<code>(jhdhd}{sdddd){}</code>	<code>False</code>
<code>a(h{g\$2[j)h]h}</code>	<code>False</code>
<code>{abc(ddd)ee[ff()dd]ee}</code>	<code>True</code>

Answer

```
1 s = input()
2 o = '({['    # Opening brackets
3 c = '})]'    # Closing brackets for the opening bracket of same index
4 b = ''        # A string variable takes the open brackets
5 match = True # Boolean variable for validation
6 for i in s:
7     if i in o:
8         b = b+i # concatenated to b if it i a opening bracket
9     if i in c:
10        # o[c.index(i)] gives the matching opening bracket for the closing
11        # bracket i
12        # o[c.index(i)] in b provides the presence of same type of bracket
13        # is opened before
14        # o[c.index(i)] should be the last opened bracket to be closed.
15        if o[c.index(i)] in b and o[c.index(i)] == b[-1]:
16            b = b[:-1] # removing the last brack because it is properly
17            closed
18        else:
19            match = False # validated to False if above conditions are not
20            satisfied
21        break
22 if len(b) != 0: # checking for remaining brackets which are not matched
23     match = False
24 print(match)
```

Test Cases

Public

Input	Output
(jhhdhd}{sdddd){}	False
[{(sa]sa(aaa)}	False
[]{{}()}{}[{}())]	True

Private

Input	Output
a{kjjf(ddfffs)hh[f(hh)d]h}d(hhd)	True
[{{(([jjhh])}}]]	True
[{{(([jjhh])}}]]][False
(({}))	False

Problem 3

Question

Accept a string from the user and print the encrypted string according to the following conditions:-

- Each letter should be replaced by the letter which is at the same position from reverse in alphabets like `a` is replaced by `z`, `b` is replaced by `y`..... `y` is replaced by `b`, `z` is replaced by `a`
- Uppercase letters should be in uppercase and lowercase letters should be in lowercase after conversion.
- Each digit should be replaced by a digit which is at the same position from reverse in `(0,1,2,...9)` like, `0` is replaced by `9`, `1` is replaced by `8` `8` is replaced by `1` and `9` is replaced by `0`.
- Blank space should be replaced by `'_'` and other types of character remain the same.

Answer

```
1 #getting input
2 message1 = input()
3 alp = "abcdefghijklmnopqrstuvwxyz"
4 nm = "0123456789"
5 message2 = "";
6 # Read the each character from message1 one by one
7 for i in message1:
8     # If character is alphabet
9         if i.isalpha() == True:
10            # If character is in uppercase
11                if i.isupper() == True:
12                    index = alp.index(i.lower())
13                    message2 += (alp[25-index]).upper()
14            # If character is in lowercase
15                else:
16                    index = alp.index(i)
17                    message2 += (alp[25-index])
18        # If character is digit
19        elif i.isdigit() == True:
20            index=nm.index(i)
21            message2 += (nm[9-index])
22    # If character is blank space
23    elif i == " ":
24        message2 += "_"
25    # For other character
26    else:
27        message2 += i
28 print(message2)
```

Test Cases

Public

Input	Output
abcde123	zyxwv876
This is Data Science course	Gsrh_rh_Wzgz_Hxrvmxv_xlfihv
abc@123.com	zyx@876.xln

Private

Input	Output
zyxwvutsrqp	abcdefghijklm
@#&^*.()	@#&^*.()
ABCDEFGHIJK	ZYXWVU@TSRQP
abcd efgh @ 9876543210	zyxw_vuts @_0123456789

Problem 4

Question

Accept a non-empty list of space-separated positive integers as input from the user and print all numbers in the list which are greater than the average in non-descending order. The output format should be a sequence of space-separated integers. For example:

Input

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

Output

```
1 | 4 5 6 7
```

Explanation

Average is $(5 + 6 + 3 + 2 + 7 + 1 + 4 + 3)/8 = 3.875$.

Answer

```
1 # Getting input and after split from blank space assign to n
2 n=input().split(" ")
3 l=[]
4 total=0
5 # Append each number in l from n after convert str to int and calculate
6 # total
7 for i in n:
8     l.append(int(i))
9     total+=int(i)
10 # Sort the list elements
11 l.sort()
12 # Calculate average
13 average=total/len(n)
14 # Print each number which is greater than average
15 for i in range(len(l)):
16     if l[i]>average and i!=len(l)-1:
17         print(l[i],end=" ")
18     elif l[i]>average and i==len(l)-1:
19         print(l[i])
```

Testcases

Public

Input 1

```
1 | 9 8 7 6 5 4 3 2 1
```

Output 1

```
1 | 6 7 8 9
```

Input 2

1	2 2 2 2 2 2 2 3 3 3 3 3
---	-------------------------

Output 2

1	3 3 3 3 3
---	-----------

Input 3

1	5 5 5 5 6 6 6 6 4 4 4 4
---	-------------------------

Output 3

1	6 6 6 6
---	---------

Private**Input 1**

1	0 1 3 5 7 9 13 11 10 8 6 4 2
---	------------------------------

Output 1

1	7 8 9 10 11 13
---	----------------

Input 2

1	100 50 0 150 200
---	------------------

Output 2,

1	150 200
---	---------

Input 3

1	1 1 1 1 1 1 1 1 1 2
---	---------------------

Output 3

1	2
---	---

Problem 5

Question

Write a program to accept a non-empty sequence of numbers separated by comma. Print this sequence in the same line separated by comma after removing all duplicate values while preserving the original order. For example:

Input

```
1 | 6,5,9,2,6,9,5
```

Output

```
1 | 6,5,9,2
```

Answer

```
1 # Getting input and after split from blank space assign to l1
2 l1=input().split(",")
3 l2=[]
4 l3=[]
5 l=len(l1)
6 # Append each element of l1 in l2 after convert from str to int
7 for i in l1:
8     l2.append(int(i))
9 # Check each element from l2 if it is not in l3 then append it to l3
10 for i in l2:
11     if i not in l3:
12         l3.append(i)
13 # Print the elements of l3
14 for i in l3[:-1]:
15     print(i,end=",")
16 print(l3[-1])
```

Test Cases

Public

Input 1

```
1 | 6,5,9,2,6,9,5
```

Output 1

```
1 | 6,5,9,2
```

Input 2

```
1 | 1,2,3,4,5,6,7,8,8,7,6,5,4,3,2,1
```

Output 2

```
1 | 1,2,3,4,5,6,7,8
```

Private

Input 1

```
1 | 12,24,35,24,88,120,155,88,120,155
```

Output 1

```
1 | 12,24,35,88,120,155
```

Input 2

```
1 | 1,2,3,4,5,6,7,8,9,0
```

Output 2

```
1 | 1,2,3,4,5,6,7,8,9,0
```

Input 3

```
1 | 1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,5
```

Output 3

```
1 | 1,2,3,4,5
```

Input 4

```
1 | -1,-3,-4,-5,1,2,3,4,5
```

Output 4

```
1 | -1,-3,-4,-5,1,2,3,4,5
```

Problem 6

Question

A clockwise rotation of a list consists of taking the last element and moving it to the beginning of the list. For instance, if we rotate the list [1,2,3,4,5], we get [5,1,2,3,4]. If we rotate it again, we get [4,5,1,2,3].

Write a program to accept a non-empty sequence of numbers separated by space and a positive integer k and print the list elements in same line separated by space after k rotations. For example:

Input

1	1 2 3 4 5
2	3

Output

1	3 4 5 1 2
---	-----------

Answer

```
1 # Getting input and after split from blank space assign to seq
2 seq = input().split(' ')
3 l = []
4 # Append each element of seq in l after convert from str to int
5 for i in seq:
6     l.append(int(i))
7 n = len(l)
8 # Calculate the remainder for reduce the rotation if k is larger than length
of l
9 k = int(input())%n
10 # Copy all elements from list l to list rt
11 rt = l[0:]
12 # Assign number from l at correct place in rt after k rotation
13 for i in range(0,n):
14     rt[i] = l[i - k]
15 # Print All elements of list rt
16 for i in range(n-1):
17     print(rt[i],end = " ")
18 print(rt[n-1])
```

Test Cases

Public

Input 1

1	1 2 3 4 5
2	3

Output 1

1	3 4 5 1 2
---	-----------

Input 2

1	9 8 7 6 5 4 3 2 1
2	9

Output 2

1	9 8 7 6 5 4 3 2 1
---	-------------------

Input 3

1	2 3 2 3 2 3 2 3
2	29

Output 3

1	3 2 3 2 3 2 3 2
---	-----------------

Input 4**Private****Input 1**

1	5 4 3 2 1
2	1

Output 1

1	1 5 4 3 2
---	-----------

Input 2

1	2 5 6 8 4 9 7 3 1 9 8 6 8
2	5

Output 2

1	1 9 8 6 8 2 5 6 8 4 9 7 3
---	---------------------------

Input 3

1	2 2 2 2 2 2 2 2
2	5

Output 3

```
1 | 2 2 2 2 2 2 2
```

Input 4

1	3 4 2 1 5 6 2 1 7 8 2 1 9 0 2 1 -2 -5
2	95

Output 4

1	0 2 1 -2 -5 3 4 2 1 5 6 2 1 7 8 2 1 9
---	---------------------------------------

Week-4 Graded Assignment (Programming)

Week-4 Graded Assignment (Programming)

Problem 1 [10]

Question

Answer

Test Cases

 Public

 Input 1

 Output 1

 Input 2

 Output 2

 Private

 Input 1

 Output 1

 Input 2

 Output 2

 Input 3

 Output 3

 Input 4

 Output 4

Solution

Tags

Problem 2 [10]

Question

Answer

Test Cases

 Public

 Private

Solution

Tags

Problem 3 [15]

Question

Answer

Test Cases

 Public

 Input 1

 Output 1

 Input 2

 Output 2

 Private

 Input 1

 Output 1

 Input 2

 Output 2

 Input 3

 Output 3

Solution

Problem 4 [15]

Question

Answer 1

Answer 2

[Test Cases](#)

[Public](#)

[Input 1](#)

[Output 1](#)

[Input 2](#)

[Output 2](#)

[Private](#)

[Input 1](#)

[Output 1](#)

[Input 2](#)

[Output 2](#)

[Input 3](#)

[Output 3](#)

[Input 4](#)

[Output 4](#)

[Solution](#)

[Tags](#)

Problem 1 [10]

Question

Calculate the standard deviation with respect to two significant decimals from the multiline numerical value obtained from the user. If the user gives the value `END` then that is the end of the data points. X_i be the data points and \bar{X} be the average of data points.

$$\sigma = \sqrt{\frac{\sum_i^n (X_i - \bar{X})^2}{n - 1}}$$

Answer

```
1 s, l = 0, []
2 x = input() # Getting the first input
3 while x != 'END': # Evaluating for the first and further inputs
4     l.append(float(x))
5     x = input()
6
7 if len(l) > 1: # Only one input leads to zero division error in calculation
8     standard deviation
9     avg = sum(l) / len(l)
10    for i in l:
11        S += (i-avg)**2 # Summation of square of difference with mean
12    SD = (S / (len(l)-1))**0.5 # Evaluation for standard deviation
13    print(f'{SD:.2f}')
```

Test Cases

Public

Input 1

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

Output 1

```
1 | 2.74
```

Input 2

```
1 124
2 1124
3 -1342
4 -214
5 -153
6 -215
7 -15
8 END
```

Output 2

```
1 | 721.94
```

Private**Input 1**

```
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 END
```

Output 1

```
1 | 0.00
```

Input 2

```
1 10
2 -5
3 2
4 -1
5 0
6 0
7 0
8 0
9 0
10 7
11 END
```

Output 2

```
1 | 4.24
```

Input 3

```
1 | 1.531
2 | 1.32523
3 | 4.32143
4 | 9.524
5 | 8.2452
6 | END
```

Output 3

```
1 | 3.77
```

Input 4

```
1 | 1.234e5
2 | -9.234e4
3 | -73e3
4 | -43332
5 | 98245
6 | 4e3
7 | END
```

Output 4

```
1 | 90089.92
```

Solution

Tags

while, list

Problem 2 [10]

Question

Write a program to accept a string from the user that contains (and) brackets. If the string has properly matched parentheses, then print the maximum nesting depth. If the brackets are not properly matched, print Not matched .

Note:

- Parentheses (and) are matched if every (has a matching) after it.
- !(a)b, has a nesting depth of 1. a1(ad(d4)2)a4 has a depth of 2 and so on.

Input	Output
(7)(a	Not matched
a)*?	Not matched
((jk1)78(A)&l(8(dd(FJI:),):)?)	4

Answer

```
1 # Getting input and initialize max_depth = 0 and match =False
2 s = input()
3 max_depth = 0
4 match = True
5 # Match counting of open brackets and close brackets
6 if s.count("(") == s.count(")"):
7     depth = 0
8     for i in s:
9         # Check each character from s,if character is "(" then increase depth by 1
10        if i == "(":
11            depth = depth + 1
12        # if depth value become greater than max_depth then assign depth = max_depth
13        if depth > max_depth:
14            max_depth = depth
15        # if character is ")" then decrease depth by 1
16        elif i == ")":
17            depth = depth - 1
18        # if depth value become -1, means brackets are not matched, assign match = false and break the loop
19        if depth == -1:
20            match = False
21            break
22        # if character is other than "(" or ")" then skip without any operation
23        else:
24            pass
25    #if brackets counts are not matched
26    else:
27        match = False
28    # print the output according to match value
29    if match == True:
30        print(max_depth)
31    else:
```

```
32 |     print("Not matched")
```

Test Cases

Public

Input	Output
(7)(a	Not matched
a)*(?	Not matched
((jk1)78(A)&l(8(dd(FJI:,):)?)	4

Private

Input	Output
(hhfgfhh(fffff))9()	Not matched
a)(*(?	Not matched
((((aaaaa)AA)AA)A)SS(S(S(S(D(D(D))))))	7

Solution

Tags

Problem 3 [15]

Question

Write a program to obtain integers in multiple lines and print all pairs where the sum of any two integers is present in the obtained input.

Note:

- The output should be in non-descending order with respect to the first printed number in the line.
- Final line of input will be an empty line

Answer

```
1 # Getting the input and appending into the list
2 l = []
3 n = input()
4 while n:
5     l.append(int(n))
6     n = input()
7 l.sort() # sorting to maintain a non-descending order
8
9 # A two level loop for comparision of two element with each other
10 for i in range(len(l)):
11     for j in range(len(l)):
12         if l[i] + l[j] in l and i != j: # required condition and prevention
13             print(l[i], l[j])
```

Test Cases

Public

Input 1

```
1 1
2 6
3 8
4 9
5
```

Output 1

```
1 1 8
2 8 1
```

Input 2

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

Output 2

1	2 2
2	2 4
3	2 4
4	2 6
5	2 6
6	2 7
7	2 2
8	2 4
9	2 4
10	2 6
11	2 6
12	2 7
13	3 4
14	3 4
15	3 6
16	3 6
17	4 2
18	4 2
19	4 3
20	4 4
21	4 2
22	4 2
23	4 3
24	4 4
25	6 2
26	6 2
27	6 3
28	6 2
29	6 2
30	6 3
31	7 2
32	7 2

Private

Input 1

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

Output 1

1	-10	2
2	-10	2
3	-5	-3
4	-5	2
5	-3	-5
6	-3	9
7	2	-10
8	2	-10
9	2	-5
10	2	6
11	2	6
12	6	2
13	6	2
14	9	-3

Input 2

1	2
2	4
3	7
4	6
5	4
6	

Output 2

1	2	4
2	2	4
3	4	2
4	4	2

Input 3

1	624
2	620
3	566
4	623
5	340
6	693
7	333
8	446
9	827
10	728
11	

Output 3

1	
---	--

Solution

Problem 4 [15]

Question

Write a program to obtain a matrix from the user and rotate it in the anti-clockwise direction by 90 degrees.

$$\begin{pmatrix} a & b & c \\ e & f & g \end{pmatrix} \rightarrow \begin{pmatrix} c & g \\ b & f \\ a & e \end{pmatrix}$$

Note:

- The user input will be in multiple lines.
- Each line represents the elements of the rows where the number will be separated by spaces.
- Final line of input will be an empty line.
- Inputs need not be integers.
- No space at the end of line.

Answer 1

```
1 # Creating a nested list from the input
2 M = []
3 row = input()
4 while row:
5     t = []
6     for i in row.strip().split(' '):
7         t.append(i)
8     M.append(t)
9     row = input()
10
11
12 # New nested list for the rotated matrix
13 M_ = []
14 for i in range(len(M[0])):
15     M_.append([])
16     for j in range(len(M)):
17         M_[i].append(0)
18
19 # Transformation
20 ## Transpose
21 for i in range(len(M)):
22     for j in range(len(M[0])):
23         M_[j][i] = M[i][j]
24
25 ## Flipping the rows of the transposed matrix
26 M_ = M_[::-1]
27
28 # Printing Rotated Matrix
29 for i in range(len(M_)):
30     for j in range(len(M_[0])):
31         if j != len(M_[0])-1:
32             print(M_[i][j], end=' ')
33         else:
34             print(M_[i][j], end=' ')
```

Answer 2

```

1 # Some parts of the code used will be covered in the later weeks
2 M, row = [], input()
3 while row:
4     M.append([int(i) for i in row.strip().split(' ')])
5     row = input()
6 M_ = [[M[j][i] for j in range(len(M))] for i in range(len(M[0]))][::-1]
7 for i in range(len(M_)):
8     print(*M_[i])
9 print()

```

Test Cases

Public

Input 1

1	1 2 3
2	4 5 6
3	

Output 1

1	3 6
2	2 5
3	1 4

Input 2

1	a b
2	c d
3	

Output 2

1	b d
2	a c

Private

Input 1

1	1 2 0 0 0
2	9 8 7 6 1
3	

Output 1

```
1 | 0 1
2 | 0 6
3 | 0 7
4 | 2 8
5 | 1 9
```

Input 2

```
1 | 0.987 0.1 0.0
2 | 0.0 0.0 0.0
3 | 1.1 567 43
4 | 0 9.8 -7.3
5 |
```

Output 2

```
1 | 0.0 0.0 43 -7.3
2 | 0.1 0.0 567 9.8
3 | 0.987 0.0 1.1 0
```

Input 3

```
1 | ab bc ce ef
2 | gh hi ij jk
3 | lm mn no pq
4 |
```

Output 3

```
1 | ef jk pq
2 | ce ij no
3 | bc hi mn
4 | ab gh lm
```

Input 4

```
1 | * * * *
2 | * * . .
3 |
```

Output 4

```
1 | *
2 | *
3 | *
4 | *
```

Solution

Tags

Week-5, Practice, Theory

Week-5, Practice, Theory

Problem 1

Question

Answer

Solution

Problem 2

Question

Answer

Solution

Common data for problem 3 and 4

Problem 3

Answer

Solution

Problem 4

Answer

Solution

Common data for problem 5 and 6

Problem 5

Answer

Solution

Problem 6

Answer

Solution

Problem 7

Question

Answers

Solution

Problem 8

Question

Answer

Solution

Problem 9

Question

Answers

Solution

Problem 10

Question

Answers

Solution

Problem 11

Question

Answers

Solution

Problem 12

Question

Answer

Solution

Problem 13

Question

Answer

Solution

Problem 14

Question

Answer

Solution

Problem 1

Question

What is the value of `l1` at the end of execution of the code given below?

```
1 | l1 = [1,2,3,4,5,6,7,8,9]
2 | l1[0:2] = [10,20,30,40,50]
```

- (a) [10, 20, 30, 40, 50, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- (b) [10, 20, 30, 40, 50, 3, 4, 5, 6, 7, 8, 9]
- (c) [[10, 20, 30, 40, 50], 3, 4, 5, 6, 7, 8, 9]
- (d) [10, 20, 30, 40, 50, 6, 7, 8, 9]
- (e) Error

Answer

(b)

Solution

Python allows you to assign new slices to replace old slices of a list in a single operation. This is called slice assignment. So in this problem element `l1[0:2]` will be replaced by `[10,20,30,40,50]`. Hence, option b is correct.

Problem 2

Question

What will be the output of the following code-snippet?

```
1 l1=[1,2,3,4,5,6,7,8,9]
2 l1[0:1]=[10]
3
4 l2=[1,2,3,4,5,6,7,8,9]
5 l2[0]=[10]
6
7 print(l1==l2)
```

- (a) True
- (b) False
- (c) Error

Answer

(b)

Solution

In line 2 element `1` of `l1` will be replaced by `10` and in line 5 element `1` of `l2` will be replaced by `[10]`. After execution of code, `l1` becomes `[10, 2, 3, 4, 5, 6, 7, 8, 9]` and `l2` becomes `[[10], 2, 3, 4, 5, 6, 7, 8, 9]`, so print statement will return False. Hence, option `b` is correct.

Common data for problem 3 and 4

```
1 def fact(n):
2     if(n==0):
3         return 1
4     else:
5         return xxxxx
```

Problem 3

Complete the following recursive function for calculating the factorial of a positive integer `n`. It is a Multiple Select Question(MSQ).

- (a) xxxxx: `n*fact(n-1)`
- (b) xxxxx: `(n-1)*fact(n+1)`
- (c) xxxxx: `(n-1)*fact(n-1)`
- (d) xxxxx: `n*fact(n+1)`
- (e) xxxxx: `fact(n-1)*n`

Answer

(a), (e)

Solution

In the above code-snippet, `xxxxx` can be replaced by `n*fact(n-1)` and `fact(n-1)*n` to compute factorial of `n`.

Problem 4

How many times will the function `fact` be called for computing the factorial of 10? It is a Numerical Answer Type (NAT) Question.

Answer

11

Solution

```
1 def fact(n):
2     global count
3     count += 1
4     if (n==0):
5         return 1
6     else:
7         return n * fact(n-1)
8 count = 0
9 print(fact(10))
10 print(count)
```

You can see the count value is 11 after executing the above code which define the number of execution of `fact` function. So the answer is 11.

Common data for problem 5 and 6

Observe the following definitions for the function named `display_sum`. The purpose of this function is to display the sum of the numbers passed as arguments to it.

1.

```
1 | def display_sum(a=0,b=0,c=0):  
2 |     print(a+b+c)
```

2.

```
1 | def display_sum(a,b=0,c=0):  
2 |     print(a+b+c)
```

3.

```
1 | def display_sum(a=0,b,c=0):  
2 |     print(a+b+c)
```

4.

```
1 | def display_sum(a,b=0,c):  
2 |     print(a+b+c)
```

5.

```
1 | def display_sum(a,b,c=0):  
2 |     print(a+b+c)
```

6.

```
1 | def display_sum(a=0,b=0,c):  
2 |     print(a+b+c)
```

Problem 5

Which of the above function definitions are invalid. It is a Multiple Select Question(MSQ).

Answer

3, 4 and 6

Solution

Non-default arguments cannot follow default arguments in function definition. So option 3, 4 and 6 not following this order because of this 3, 4 and 6 are invalid.

Problem 6

Which of the above definitions print the correct answer for `display_sum(10,10)` . It is a Multiple Select Question(MSQ).

Answer

1, 2 and 5

Solution

Option 1, 2 and 5 have valid definition of function, so for `display_sum(10,10)` we are passing `a` and `b` value in function , and `c` already defined as a default value to 0 ,so these three functions print the correct output .

Problem 7

Question

Which of the following statements are correct? It is a Multiple Select Question(MSQ).

- (a) If the `return` statement is not used inside the function, the function will return 0.
- (b) In order to change the value of the global variable inside the function, keyword `global` is used.
- (c) Default argument can be defined before keyword argument in the function.
- (d) Functions are an effective way to reuse program codes.
- (e) Function can be passed as an argument in another function.
- (f) The scope of the local variable is limited to the function where it is defined.

Answers

(b), (d), (e), (f)

Solution

Option (a) is incorrect because If the `return` statement is not used inside the function, the function will return `None`.

option (c) is incorrect because Non-default arguments cannot follow default arguments in function definition.

Other than a and c, all options are correct.

Problem 8

Question

What does the following code-snippet print?

```
1 | x = 10
2 | y = 20
3 | z = 30
4 | def show(x,y):
5 |     print(x + y + z)
6 |     x = 40
7 |     y = 50
8 |     print(x + y + z)
9 |
10| show(100,300)
11| print(x + y + z)
```

(a)

```
1 | 60
2 | 120
3 | 60
```

(b)

```
1 | 430
2 | 120
3 | 60
```

(c)

```
1 | 430
2 | 120
3 | 120
```

(d)

```
1 | 430
2 | 430
3 | 120
```

Answer

(b)

Solution

In first print statement inside function, x will be 100 and y will be 300 and z will be 30 the, print statement print 430. Here x and y are local variable for function.

In second print statement inside function, x will be 40 and y will be 50 and z will be 30 the, print statement print 120. Here x and y are local variable for function.

In third print statement outside the function, x will be 10 and y will be 20 and z will be 30 the, print statement print 6. Here x and y are local variable for function.

Hence, Correct option is b. Here x ,y and z are global variable for function.

Problem 9

Question

Which positive integer input from the user will make this program display 264 as the output? It is a Numerical Answer Type (NAT) Question.

```
1 def func(x):
2     return x + 1
3 n = int(input())
4 print(int(func(n / 2) * func(n + 1)))
```

Answers

21

Solution

For input value n=21 `func(n/2)` will return 11.5 and `func(n + 1)` will return 23 after that `print(int(11.5*23))` print 264 in output.

Problem 10

Question

Given the following function description

- Returns True if it is possible to add two of the arguments to get the remaining one and False otherwise.

Example:-

- `Check(1, 2, 3)` returns True
- `Check(3, 1, 2)` returns True
- `Check(3, 2, 2)` returns False

Which of the following function definitions produces the desired effect?

(a)

```
1 | def Check(a, b, c):  
2 |     l = [a,b,c]  
3 |     l.sort()  
4 |     return min(l) + min(l[1:]) == max(a,b,c)
```

(b)

```
1 | def check(a, b, c):  
2 |     if ((a + b == c) and (a + c == b) and (b + c == a)):  
3 |         return True  
4 |     return False
```

(c)

```
1 | def Check(a, b, c):  
2 |     if (a + b == c):  
3 |         return True  
4 |     elif ((a + c == b) or (b + c == a)):  
5 |         return True  
6 |     return False
```

(d)

```
1 | def Check(a, b, c):  
2 |     if (a + b != c):  
3 |         return False  
4 |     elif (a + c != b):  
5 |         return False  
6 |     elif (b + c != a):  
7 |         return False  
8 |     else:  
9 |         return True
```

(e)

```
1 | def check(a, b, c):
2 |     l = [a,b,c]
3 |     l.sort()
4 |     return l[0] + l[1] == l[2]
```

Answers

(a), (c) and (e)

Solution

Options a, c and e have the correct logic to give the correct output for the above problem..

Problem 11

Question

Code

```
1 x = 10
2 y = 20
3 z = 30
4 def show(z):
5     global x , y
6     x = 40
7     y = z
8     z = 60
9 n=int(input())
10 show(n)
11 print(x + y + z)
```

Which positive integer input `n` from the user will make this program display the following output?
It is a Numerical Answer Type (NAT) Question.

1 | 120

Answers

50

Solution

In the `show` function `x` and `y` are defined as a global variables, so the outer variable's value can be changed inside the function and reflected outside. So after the call function in line 11, `x` becomes 40 and `y` becomes `z` (passing value `n`) and `z` remains 30. So for `n=50` this prints statement will print 120 in output.

Problem 12

Question

Code

```
1 | l1 = [1,2,3,4,5]
2 | l2 = [6,7,8,9]
3 | ### --Fill Code--
4 | print(newlist)
```

Select the correct code to fill in the above code-snippet to print the following output. It is a Multiple Select Question.

Output

```
1 | [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

(a)

```
1 | newlist = l1 + l2
```

(b)

```
1 | newlist = extend(l1,l2)
```

(c)

```
1 | newlist = l1.extend(l2)
```

(d)

```
1 | newlist.extend(l1,l2)
```

(e)

```
1 | l1.extend(l2)
2 | newlist = l1
```

Answer

(a) , (e)

Solution

Here option a and e code statement is correct to print the above output.

Problem 13

Question

Code

```
1 | def fun(l):
2 |     l.append(4)
3 |     return l
4 | l=[1,2,3]
5 | print(type(fun))
```

What does the following code-snippet print?

(a)

```
1 | <class 'list'>
```

(b)

```
1 | <class 'int'>
```

(c)

```
1 | <class 'function'>
```

(d)

```
1 | Error
```

Answer

(c)

Solution

`type(function_name)` is returns `<class 'function'>`

Problem 14

Question

Code

```
1 | def fun(l):
2 |     l=l*5
3 |     return l
4 | l=[1]
5 | print(type(fun(l)))
6 | print(fun(l))
```

What does the following code-snippet print?

(a)

```
1 | <class 'list'>
2 | [5]
```

(b)

```
1 | <class 'list'>
2 | [1, 1, 1, 1, 1]
```

(c)

```
1 | <class 'int'>
2 | 5
```

(d)

```
1 | <class 'function'>
2 | [1, 1, 1, 1, 1]
```

Answer

(b)

Solution

`type(function_name())` returns the type of return value by function. If the `return` statement is not used inside the function, the function will return None. Here function returns a list so first print statement print `<class 'list'>` and when we multiply any list by n then all elements of list repeated n times. So second print statement will print [1, 1, 1, 1, 1].

##

Week-5 Graded Assignment (Theory)

Week-5 Graded Assignment (Theory)

Problem 1 [3]

Question

Answer

Solution

Tags

Problem 2

Question a [2]

Answer a

Question b [2]

Answer b

Solution

Tags

Problem 3 [4]

Question

Answer

Solution

Tags

Problem 4 [4]

Question

Answer

Solution

Tags

Problem 5

Question a [4]

Answer a

Solution

Question b [4]

Answer b

Solution

Question c [4]

Answer c

Solution

Tags

Problem 6 [4]

Question

Answer

Solution

Tags

Problem 7

Question a [2]

Answer

Solution

Question b [4]

Answer

Solution

Question c [4]

Answer

Solution

Tags

Problem 8 [5]

Question

[Answer](#)

[Solution](#)

[Problem 9 \[4\]](#)

[Question](#)

[Answer](#)

[Solution](#)

Problem 1 [3]

```
1 | def f(a=1, b=2, c, d):  
2 |     return a + b + c + d  
3 | print(f(0, 0, 3, 4))
```

Question

What will be the output of the above code-snippet?

- (a) 10
- (b) 3
- (c) 7
- (d) SyntaxError

Answer

(d) SyntaxError

Solution

Keyword argument should given at the end after the place holding arguments

Tags

functions

Problem 2

```
1 | def f(a, b, c=10, d=19):  
2 |     return a + b + c + d
```

Question a [2]

What will `f(1, 2)` return?

- (a) 3
- (b) 32
- (c) 29
- (d) SyntaxError

Answer a

(b) 32

Question b [2]

What will `f(c=1, d=2, 1, 2)` return?

- (a) 3
- (b) 32
- (c) 29
- (d) SyntaxError

Answer b

(d) SyntaxError

Solution

`a` and `b` take the values `1` and `2`. `c` and `d` have the default values `10` and `19`. Hence, the return value be $1 + 2 + 10 + 19 = 32$.

Tags

functions

Problem 3 [4]

Question

```
1 def assert(x):
2     return x==int(str(x))
3 assert(1)
```

What will be the output of the code?

- (a) True
- (b) False
- (c) Return no value
- (d) SyntaxError

Answer

(d) SyntaxError

Solution

keywords cannot made as function names.

Tags

functions, keywords

Problem 4 [4]

Question

```
1 def f():
2     return x
3 x = 10
4 print(f())
```

What will be the output of the above code?

- (a) None
- (b) 10
- (c) NameError
- (d) SyntaxError

Answer

- (b) 10

Solution

`x` have the global scope, hence `x` used inside the function it will have have the value 10.

Note: When the value of `x` is altered inside the function then it will result in an error. It can be resolved defining `x` inside the function or using `global x` which lets the program to know `x` is global variable.

Tags

functions

Problem 5

```
1 def f(x):
2     if x == 0:
3         return x
4     elif x > 5:
5         return g(x-2)
6     else:
7         return f(x-1)
8
9 def g(x):
10    if x == 0:
11        return x
12    elif x > 5:
13        return f(x-1)
14    else:
15        return g(x-2)
```

Question a [4]

Given that $0 \leq x \leq 10$, when does a recursion error occur in the given code? It is a Multiple Select Question (MSQ).

- (a) At any level of recursion when a positive odd number which is less than 5 is passed into `g()`
- (b) At any level of recursion when a positive odd number which is less than 5 is passed into `f()`
- (c) At any level of recursion 5 is passed into `g()`
- (d) At any level of recursion 5 is passed into `f()`

Answer a

- (a) At any level of recursion when a positive odd number which is less than 5 is passed into `g()`
- (c) At any level of recursion 5 is passed into `g()`

Solution

function f	Returning value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	RecursionError
8	0
9	0
10	RecursionError

function g	Returning value
0	0
1	RecursionError
2	0
3	RecursionError
4	0
5	RecursionError
6	0
7	0
8	RecursionError
9	0
10	0

Question b [4]

For $5 < x < 10$, enter the values of x for which $f(x)$ results in a RecursionError. It is a Numerical Answer Type (NAT) question.

Answer b

7

Solution

Self explanatory

Question c [4]

For `5 < x < 10`, enter the values of `x` for which `g(x)` results in a `RecursionError`. It is a Numerical Answer Type (NAT) question.

Answer c

8

Solution

Self explanatory

Tags

Problem 6 [4]

Question

```
1 import random
2 x = random.random()
3 for i in range(4):
4     print(x)
```

What is/are the possible outputs for the given code? It is a Multiple Select Question (MSQ).

(a)

```
1 0.8050796857929327
2 0.594136093400026
3 0.34224707799217524
4 0.6758211524540326
```

(b)

```
1 0.5830809720043407
2 0.5830809720043407
3 0.5830809720043407
4 0.5830809720043407
```

(c)

```
1 0.922785520795311
2 0.531735912593371
3 0.5791968649226946
4 0.5791968649226946
```

(d)

```
1 0.8507248045932425
2 0.8547127179926637
3 0.8757817076206122
4 0.383942471128478
```

Answer

(b)

```
1 0.5830809720043407
2 0.5830809720043407
3 0.5830809720043407
4 0.5830809720043407
```

Solution

`x` holds the random real number between 0 to 1 using `random` library's `random` function. The same `x` is printed four times using the for statement. Hence, the all lines of output should be same.

Tags

Problem 7

Refer <https://docs.python.org/3/library/math.html#math.hypot>

Question a [2]

For a right-angled triangle, what are the appropriate arguments that can be passed into the function `math.hypot()`?

- (a) Lengths of the sides which are adjacent to the right angle (90 degrees)
- (b) Lengths of the three sides of the right-angled triangle
- (c) All three angles between the sides (in degrees) of the right-angled triangle
- (d) Any two angles between the sides (in degrees) of the right-angled triangle

Answer

- (a) Length of the sides which are adjacent to the right angle (90 degrees)

Solution

`math.hypot()` takes the arguments and returns the square root of sum of squares $\sqrt{a^2 + b^2}$. By taking two sides adjacent to the right angle will give the hypotenuse.

Question b [4]

What is the minimum number of arguments required for the function `math.hypot()` for it to return a value without throwing any error?

- (a) 0
- (b) 1
- (c) 2
- (d) 3

Answer

- (a) 0

Solution

By default `math.hypot()` with no parameter will indeed return a value `0`. This is true in Python version 3.8 and above. However, in earlier versions of Python, this is not true: at least two arguments are required.

Question c [4]

Is the following statement true or false?

`math.hypot(a, b)` always returns a positive value, where `a` and `b` are real numbers.

- (a) True
- (b) False

Answer

(a) True

Solution

There is a slight ambiguity here. `math.hypot` always returns a non-negative value. It could also return zero.

Tags

Problem 8 [5]

Question

Fill `XXXX` and `YYYY` to print 0 to 4 each in a new line. It is a Multiple Select Question (MSQ).

```
1 def f(XXXX):
2     for i in range(n):
3         print(i)
4 YYYY
5 f()
```

Expected Output

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

- (a) `XXXX: n = 10` and `YYYY: n = n-5`
- (b) `XXXX: n = 5` and `YYYY:`
- (c) `XXXX:` and `YYYY:`
- (d) `XXXX:` and `YYYY: n = 5`

Answer

(b), (d)

Solution

Option	Comment
<code>XXXX: n = 10</code> and <code>YYYY: n = n-5</code>	At line 5, <code>n = n-5</code> will throw an error where <code>n</code> is not defined.
<code>XXXX: n = 5</code> and <code>YYYY:</code>	<code>n=5</code> is given as default argument hence it will print expected output.
<code>XXXX:</code> and <code>YYYY:</code>	Throw an error inside the function that <code>n</code> is not defined.
<code>XXXX:</code> and <code>YYYY: n = 5</code>	<code>n</code> is defined outside the function with global scope and can be accessed inside the function.

Problem 9 [4]

Question

Choose the correct code to get the output 8.0.

```
1 import math
2 '''Fill in the code'''
3 print(s(64))
```

- (a) s = math.sqrt
- (b) s = math.sqrt()
- (c) s = sqrt
- (d) s = 8.0

Answer

- (a) s = math.sqrt

Solution

s becomes a function by assigning the math.sqrt to it. Thus, s and math.sqrt are equivalent.

Week-5, Practice, Programming

Week-5, Practice, Programming

[Problem-1](#)

[Question](#)

[Test Cases](#)

[Answer](#)

[Suffix Code Block](#)

[Problem-2](#)

[Question](#)

[Test Cases](#)

[Answer](#)

[Suffix Code Block](#)

[Problem-3](#)

[Question](#)

[Test Cases](#)

[Answer](#)

[Suffix Code Block](#)

[Problem 4](#)

[Question](#)

[Test Cases](#)

[Public](#)

[Private](#)

[Answer](#)

[Suffix Code Block](#)

[Alternate Answers](#)

Problem-1

Question

Solve the following system of linear equations:

$$\begin{aligned} a_1x + b_1y + c_1 &= 0 \\ a_2x + b_2y + c_2 &= 0 \end{aligned}$$

Assumptions

- All coefficients are non-zero integers.
- The system has a unique solution.
- The solution will always be a pair of integers.

Task

The coefficients will be given as a list of integers:

- `eq1 : [a1, b1, c1]`
- `eq2 : [a2, b2, c2]`

Write a function named `solve` that accepts these two lists as input and returns the solution as output. Do not modify the name of the function.

```
1 | def solve(eq1, eq2):  
2 |     '''Return solution (x, y) as output'''  
3 |     pass
```

Note

- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Test Cases

Type	Input	Output
Public	<code>1 -2 6</code> <code>3 5 -15</code>	<code>0 3</code>
Public	<code>10 -4 -58</code> <code>6 4 10</code>	<code>3 -7</code>
Private	<code>2 -1 -3</code> <code>3 2 -1</code>	<code>1 -1</code>
Private	<code>-3 2 34</code> <code>5 -3 -56</code>	<code>10 -2</code>

Answer

```
1 def solve(eq1, eq2):
2     '''Return (x, y) as output'''
3     # The string in triple quotes above
4     # is called a doc string.
5     a1, b1, c1 = eq1      # coefficients for eq1
6     a2, b2, c2 = eq2      # coefficients for eq2
7     # What we have done in the code given above
8     # is called sequence unpacking.
9     # We have unpacked the list eq1 into the
10    # three coefficients. Ditto for eq2.
11    # Since there are three elements in each list,
12    # those three are going to be assigned to
13    # the three variables ai, bi, ci.
14
15    # Simple substitution by eliminating
16    # the variables x and y
17    x = (b1 * c2 - b2 * c1) / (a1 * b2 - a2 * b1)
18    y = (c1 * a2 - c2 * a1) / (a1 * b2 - a2 * b1)
19
20    # The assumptions are useful here.
21    # We can convert x and y to int.
22    # We won't be losing information that way.
23    return int(x), int(y)
```

Suffix Code Block

```
1 # This method of accepting input is called list-comprehension
2 eq1 = [int(word) for word in input().split()]
3 # List comprehension will be covered in upcoming weeks
4 eq2 = [int(word) for word in input().split()]
5 x, y = solve(eq1, eq2)
6 print(x, y)
```

Problem-2

Question

Find the transpose of a given matrix.

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

Assumptions

- All cells in the matrix are integers.
- The dimension of the matrix is $m \times n$, where $m, n \geq 1$. The first line in the input will have the dimension as space-separated integers.
- The next m lines in the input will be a sequence of n space-separated integers.
- The output will be the transpose of this matrix. Each row of the matrix is given as a sequence of space-separated integers.

Task

Write a function `transpose` that accepts a matrix as input and returns its transpose. You can assume that the matrix is a nested list.

```
1 | def transpose(mat):
2 |     '''Returns transpose of mat'''
3 |     pass
```

Note

- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Test Cases

Type	Input	Output
Public	3 2 1 2 3 4 5 6	1 3 5 2 4 6
Public	2 4 1 2 3 4 5 6 7 8	1 5 2 6 3 7 4 8
Private	1 5 1 2 3 4 5	1 2 3 4 5
Private	3 3 1 2 3 4 5 6 7 8 9	1 4 7 2 5 8 3 6 9

Answer

```

1 def transpose(mat):
2     '''Return transpose of mat'''
3     trans = [ ]
4     # mat is of dimensions m x n
5     # len(mat) is the number of rows
6     # mat[0] is the first row
7     # len(mat[0]) is the number of columns
8     m, n = len(mat), len(mat[0])
9     # While printing a matrix, we do the following:
10    # We go from left-> right, then top->bottom (row-first)
11    # To get the transpose, we do the following
12    # We go from top->bottom, then left->right (column-first)
13    for j in range(n):
14        trans.append([ ])
15        for i in range(m):
16            trans[j].append(mat[i][j])
17    return trans

```

Suffix Code Block

```

1 dims = input().split()
2 m, n = int(dims[0]), int(dims[1])
3 mat = [ ]
4 for i in range(m):
5     mat.append([ ])

```

```
6     for num in input().split():
7         mat[i].append(int(num))
8
9     trans = transpose(mat)
10    for i in range(n):
11        for j in range(m):
12            if j != m - 1:
13                print(trans[i][j], end = ' ')
14            else:
15                print(trans[i][j])
```

Problem-3

Question

In large-scale programming projects, organizing files and folders becomes crucial. Consider the following folder structure. The path to a file is expressed in the following manner:

```
1 /home
2 /home/mark
3 /home/mark/facebook
4 /home/mark/facebook/src
5 /home/mark/facebook/src/newsfeed.py
```

- Line-1: The `/home` folder is like a huge palace. Users have their own rooms in it. This folder is at level-1.
- Line-2: In the `home` folder, we have a user named `mark`. `/home/mark` is his room. This folder is at level-2.
- Line-3: Mark is working on a project called `facebook`, naturally within his room. This folder is at level-3.
- Line-4: Within this project, he maintains a folder named `src` to store all his `Python` files. This folder is at level-4.
- Line-5: The path points to `newsfeed.py`, one of the files in the folder. This file is at level-5.

The level of a file or folder quantifies its depth within the folder structure.

Assumptions

- There are only two types of entities: files and folders.
- Files could end with one of the following extensions:
 - `.py` — code
 - `.cpp` — code
 - `.jpg` — image
 - `.png` — image
- Unlike the chicken-egg problem, only files can reside in folders. Folders cannot be present in files.
- Folder names will always be alphanumeric characters.

Task

Given a path as input, your task is to write the following functions:

- `is_folder(path)`: this accepts a path as input and returns `True` if the path points to a folder, `False` otherwise
- `is_file(path)`: this accepts a path as input and returns `True` if the path points to a file, `False` otherwise
- `is_code(path)`: this accepts a path as input and returns `True` if the path points to a code file and `False` otherwise
- `is_image(path)`: this accepts a path as input and returns `True` if the path points to an image file and `False` otherwise
- `level(path)`: this accepts a path as input and returns the level at which it is found.

Do not print anything to the console. Your task is to write these five functions. The input will be a `path`. The output will be the outcome of the following function calls in this exact sequence:

```
1 | is_folder(path)
2 | is_file(path)
3 | is_code(path)
4 | is_image(path)
5 | level(path)
```

Test Cases

Type	Input	Output
Public	/home/mark/facebook/src/newsfeed.py	False True True False 5
Public	/home/guido/microsoft/secret/mystery/src	True False False False 6
Public	/home/einstein/relativity.jpg	False True False True 3
Private	/home	True False False False 1
Private	/home/numpy	True False False False 2
Private	/home/project/something/jpg	True False False False 4
Private	/home/random/image.png	False True False True 3
Private	/home/d1/d2/d3/d4/d5/file1.cpp	False True True False 7

Answer

```
1 # Look for .py or .cpp extension
2 def is_code(path):
3     if path[-3:] == '.py' or path[-4:] == '.cpp':
4         return True
5     return False
6
7 # Look for .jpg or .png extension
8 def is_image(path):
9     if path[-4:] == '.jpg' or path[-4:] == '.png':
10        return True
11    return False
12
13 # Check if it is a code or an image
14 def is_file(path):
15     return is_code(path) or is_image(path)
16
17 # A folder is the negation of a file
18 def is_folder(path):
19     return not is_file(path)
20
21 # Split the string based on / and find the length of the resulting list
22 def level(path):
23     return len(path.strip('/').split('/'))
```

Suffix Code Block

```
1 path = input()
2 print(is_folder(path))
3 print(is_file(path))
4 print(is_code(path))
5 print(is_image(path))
6 print(level(path))
```

Problem 4

Question

The Pearson correlation coefficient, $r(x, y)$ is a measure of association between two continuous variables x and y . The correlation $r(x, y)$ is given defined in the mathematical form as below.

$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\sum_{i=1}^n (x_i y_i) - ((\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)/n)}{\sqrt{(\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2/n)} * \sqrt{(\sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2/n)}}$$

Symbol	Meaning
n	number of data points (or count of numbers)
x_i	i-th value of the variable x , $i = 1, 2, 3, 4, \dots, n$
y_i	i-th value of the variable y , $i = 1, 2, 3, 4, \dots, n$
\bar{x}	mean or average value of the variable x
\bar{y}	mean or average value of the variable y

Implement the body of the function named `pearson_correlation`. The input is two lists of `float` values: `x` and `y`. Each floating point number is limited to 2 places after decimal.

- The function should return the correlation coefficient as a `float` value rounded to 1 decimal place.
- The function should return `0.0`, when the lengths of the two input variables are unequal.

```
1 def pearson_correlation(x, y):
2     # Write function body
3     #
4     #
5     #
```

Test Cases

Public

Input	Output
<code>1 2 3 4 5 6 7 8 9</code> <code>1 2 3 2 3 4 3 4 5</code>	<code>0.9</code>
<code>1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0</code> <code>-1.0 -2.0 -3.0 -4.0 -5.0 -1.0 -2.0 -3.0 -4.0 -5.0</code>	<code>-0.5</code>

Private

Input	Output
10.0 8.0 13.0 9.0 11.0 14.0 6.0 4.0 12.0 7.0 5.0 8.04 6.95 7.58 8.81 8.33 9.96 7.24 4.26 10.84 4.82 5.68	0.8
10.0 8.0 13.0 9.0 11.0 14.0 6.0 4.0 12.0 7.0 5.0 9.14 8.14 8.74 8.77 9.26 8.10 6.13 3.10 9.13 7.26 4.74	0.8
10.0 8.0 13.0 9.0 11.0 14.0 6.0 4.0 12.0 7.0 5.0 7.46 6.77 12.74 7.11 7.81 8.84 6.08 5.39 8.15 6.42 5.73	0.8
8.0 8.0 8.0 8.0 8.0 8.0 8.0 19.0 8.0 8.0 8.0 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.50 5.56 7.91 6.89	0.8
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0	-1.0
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0	1.0
0 1 2 3 4 5 5 4 3 2 1 0 5 4 3 2 1 0 0 1 2 3 4 5	-1.0
1.1 2.2 3.3 4.4 5.5 6.6 -7.7 -8.8 -9.9 -1 -2 -3 -2 -3 -4 3 4 5	-1.0
1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 9.9 -1.1 -2.2 -3.3 -4.4 -5.5 -6.6 -7.7 -8.8 -9.9	-1.0

Answer

```
1 def pearson_correlation(x, y):  
2     # convert input (a string of float values separated by spaces) into a  
3     # list of strings  
4     # verify the length  
5     if len(x) != len(y):  
6         return 0.0  
7     n = len(x)  
8     sum_x, sum_y = 0.0, 0.0  
9     for i in range(n):  
10        sum_x, sum_y = sum_x + x[i], sum_y + y[i]  
11    # calculate mean for x and y  
12    mean_x, mean_y = sum_x / len(x), sum_y / len(y)  
13    # calculate numerator and denominator  
14    numerator, denominator_x, denominator_y = 0.0, 0.0, 0.0  
15    for i in range(n):  
16        numerator += (x[i] - mean_x) * (y[i] - mean_y)  
17        denominator_x += (x[i] - mean_x) ** 2  
18        denominator_y += (y[i] - mean_y) ** 2  
19    # get pearson correlation coefficient  
20    denominator = (denominator_x ** 0.5) * (denominator_y ** 0.5)  
21    if denominator - 0.0 > 0.00001:  
        pearson_corr = numerator / denominator
```

```
22     return(round(pearson_corr, 1))
```

Suffix Code Block

```
1 x, y = input().split(), input().split()
2 for i in range(len(x)):
3     x[i], y[i] = float(x[i]), float(y[i])
4 print(pearson_correlation(x, y))
```

Alternate Answers

```
1 def pearson_correlation(x, y):
2     # convert input (a string of float values separated by spaces) into a
3     # list of strings
4     x, y = x.split(), y.split()
5     # verify the length
6     if len(x) != len(y):
7         return 0.0
8     n = len(x)
9     sum_x, sum_y = 0.0, 0.0
10    # change string element to float values in the list x and y
11    for i in range(n):
12        x[i], y[i] = float(x[i]), float(y[i])
13        sum_x, sum_y = sum_x + x[i], sum_y + y[i]
14
15    # calculate numerator and denominator
16    sum_x_sq, sum_y_sq, sum_xy = 0.0, 0.0, 0.0
17    for xi in x:
18        sum_x_sq += xi ** 2
19    for yi in y:
20        sum_y_sq += yi ** 2
21    for i in range(n):
22        sum_xy += x[i] * y[i]
23    numerator = sum_xy - (sum_x * sum_y) / n
24    denominator = ( (sum_x_sq - sum_x ** 2 / n) * (sum_y_sq - sum_y ** 2 /
25 n) ) ** 0.5
26    # get pearson correlation coefficient
27    if denominator - 0.0 > 0.00001:
28        pearson_corr = numerator / denominator
29    return(round(pearson_corr, 1))
```

Verification using library function.

```
1 from scipy.stats import spearmanr
2 import numpy
3 def pearson_correlation(x, y):
4     # convert input (a string of float values separated by spaces) into a
5     # list of strings
6     x, y = x.split(), y.split()
7     # verify the length
8     if len(x) != len(y):
9         return 0.0
10    n = len(x)
11    sum_x, sum_y = 0.0, 0.0
12    # change string element to float values in the list x and y
```

```
12     for i in range(n):
13         x[i], y[i] = float(x[i]), float(y[i])
14     # get pearson correlation coefficient
15     # return round(spearmanr(x, y)[0], 1)
16     return(round(numpy.corrcoef(x, y)[0][1], 1))
```

Week-5, Graded, Programming

Week-5, Graded, Programming

Problem 1

Question

Answer

Suffix Code Block

Test Cases

 Public

 Private

Problem 2

Question

Suffix Code Block

Answer

Test Cases

 Public

 Private

Problem 3

Question

Suffix Code Block

Answer

Test Cases

 Public

 Private

Problem-4

Question

Test Cases

Answer

Suffix Code Block

Problem-5

Question

Test Cases

Answer

Suffix Code Block

Problem 1

Question

A perfect number is a positive integer that is equal to the sum of its proper positive divisors (excluding itself). Fill the body of the function `perfect_number` to check whether an input integer is perfect or not. It should return boolean literal `True` if the number is perfect. Otherwise, it should return `False`.

```
1 | def perfect_number(num):  
2 |     # Write function body  
3 |     #  
4 |     #  
5 |     #  
6 |
```

Note

- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Answer

```
1 | """  
2 | A perfect number is a positive integer that is equal to the sum of its  
3 | proper positive divisors. That is, the sum of its positive divisors  
4 | excluding the number itself (also known as its aliquot sum).  
5 | """  
6 | def perfect_number(num):  
7 |     sum = 0  
8 |     if num < 2:  
9 |         return False  
10 |     else:  
11 |         for x in range(1, num):  
12 |             if num % x == 0:  
13 |                 sum += x  
14 |     return sum == num
```

Suffix Code Block

```
1 | print(perfect_number( int(input()) ))
```

Test Cases

Public

Input	Factor	Output
8	1, 2, 4	False
28	1, 2, 4, 7, 14	True

Private

Input	Factor	Output
496	1, 2, 3	True
8128	1, 2, 4, 7, 14	True
1	1	False
-1	-1	False

Problem 2

Question

You are a forum admin. You wish to find out the total score of each user using the scoring logic given below. If the user scores more than 50, then the user is given a Leader badge, otherwise a Basic badge is given.

Action	Points
read a post	1
replied to a post	3
created a new post	5

Implement the body of the function `user_score`. The function accepts 3 integer arguments — `read_count`, `reply_count`, `new_post_count`. It calculates the score based on the input counts and returns `Leader` or `Basic`.

```
1 def user_score(read_count, reply_count, new_post_count):  
2     # Write function body  
3     #  
4     #  
5     #
```

Note

- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Suffix Code Block

```
1 read_count, reply_count, new_post_count = input().split(",")  
2 print(user_score( int(read_count), int(reply_count), int(new_post_count) ))
```

Answer

```
1 def user_score(read_count, reply_count, new_post_count):  
2     score = read_count * 1 + reply_count * 3 + new_post_count * 5  
3     if score > 50:  
4         return "Leader"  
5     else:  
6         return "Basic"
```

Test Cases

Public

Input	Output
1, 1, 1	Basic
10, 10, 10	Leader

Private

Input	Output
0, 0, 0	Basic
50, 0, 0	Basic
0, 16, 0	Basic
0, 0, 10	Basic
51, 0, 0	Leader
0, 17, 0	Leader
1, 0, 10	Leader

Problem 3

Question

In the Gregorian calendar, a leap year has a total of 366 days instead of the usual 365 as a result of adding an extra day (February 29) to the year. This calendar was introduced in 1582, to replace the flawed Julian Calendar. Below criteria are used to determine if a year is a leap year or not.

- If a year is divisible by 100 then it will be a leap year if it is also divisible by 400
- If a year is not divisible by 100, then it will be a leap year if it is divisible by 4.

Implement the body of the function `check_leap_year` to incorporate above logic. The function takes any year from 1600 to 9999 (endpoints included) as an input number and should return Boolean value `True` when the year is a leap year, otherwise it should return `False`.

```
1 | def check_leap_year(year):  
2 |     # Write function body  
3 |     #  
4 |     #  
5 |     #
```

Suffix Code Block

```
1 | print(check_leap_year( int(input()) ))
```

Answer

```
1 | def check_leap_year(year):  
2 |     if (year % 400 == 0) or (year % 100 != 0 and year % 4 == 0):  
3 |         return True  
4 |     else:  
5 |         return False
```

Test Cases

Public

Input	Output
2020	True
2021	False

Private

Input	Output
1900	False
2100	False
2000	True
9999	False

Problem-4

Question

A $n \times n$ square matrix of positive integers is called a magic square if the following sums are equal:

- row-sum: sum of numbers in every row; there are n such values, one for each row
- column-sum: sum of numbers in every column; there are n such values, one for each column
- diagonal-sum: sum of numbers in both the main diagonals; there are two values

There are $n + n + 2 = 2n + 2$ values involved. All these values must be the same for the matrix to be a magic-square.

Task

Write a function `is_magic` that accepts a matrix as input and returns `YES` if it is a magic-square and `NO` if it isn't one. You can assume the following:

- The first line of input will have the dimension n of the matrix.
- The next n lines will be a sequence of n space-separated integers.

```
1 | def is_magic(mat):
2 |     """Magic square!"""
3 |     pass
```

Note

- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Test Cases

Type	Input	Output
Public	2 1 2 2 1	NO
Public	3 4 9 2 3 5 7 8 1 6	YES
Private	4 2 16 13 3 11 5 8 10 7 9 12 6 14 4 1 15	YES
Private	3 1 2 3 4 5 6 7 8 9	NO
Private	2 1 1 1 1	YES

Answer

```

1 def is_magic(mat):
2     # first get the dimension of the matrix
3     m = len(mat)
4     # the sum of the two diagonals
5     d1sum, d2sum = 0, 0
6     # (i, i) goes from top-left -> bottom-right
7     # (i, m - i - 1) goes from top-right -> bottom-left
8     # note that a single loop is enough; no nesting required
9     for i in range(m):
10         d1sum += mat[i][i]
11         d2sum += mat[i][m - i - 1]
12     # if the two diagonal sums are unequal, we can return NO
13     # unnecessary computation can be avoided
14     if not(d1sum == d2sum):
15         return 'NO'
16     # get row-sum and column-sum
17     for i in range(m):
18         rsum, csum = 0, 0
19         for j in range(m):
20             rsum += mat[i][j]
21             csum += mat[j][i]
22         if not(rsum == csum == d1sum):
23             return 'NO'
24     # if the code reaches this level
25     # then all requirements of a magic-square are satisfied
26     # so we can safely return YES

```

Suffix Code Block

```
1 m = int(input())
2 mat = [ ]
3 for i in range(m):
4     mat.append([ ])
5     for num in input().split():
6         mat[i].append(int(num))
7
8 print(is_magic(mat))
```

Problem-5

Question

n integers are written on a blackboard in a classroom. Consider the following process:

- Head to the board and pick the two smallest integers from it.
- Make a note of them in your head and then erase the two numbers from the board.
- Find the absolute value of their difference and write the result on the board.

Repeat the whole process again. Keep doing it until only one number is left on the board. Write this number down on a piece of paper and erase the board. If there is only one number to begin with, all other operations can be avoided. You just have to record this number on paper and then erase it from the board.

Assumptions

- There will be at least one number on the board to begin with.
- The board should be empty after the process is complete. No numbers must be written on it.

Task

The following functions are given to you and you don't have to write them:

- `board.IsEmpty()`: returns `True` if the board is empty and `False` otherwise
- `board.write(num)`: accepts `num` as an argument and writes the `num` on the board
- `board.erase(num)`: accepts `num` as an argument and erases the `num` from the board
- `board.min()`: returns the minimum number on the board; doesn't erase it from the board

Your task is to write a function named `process` that will execute this process given in the problem statement using the functions given above and return the number on the piece of paper.

```
1 def process():
2     # Execute the process #
3     # Return the number on the piece of paper #
4     pass
```

Input-Output

Each test case corresponds to some collection of numbers on the board. The actual input (numbers on the board) will be hidden from your view. Instead the input column in each test case will contain the test-case number.

The output will be the number you have written down in the piece of paper corresponding to that test case. Remember, we will also be checking if the board has been wiped clean at the end!

Hint

The first public test case has the following integers written on the board:

```
1 2 3 4 5
```

Numbers on the board at the beginning of each iteration of the process:

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

Test Cases

Type	Input	Output
Public	1	3
Public	2	4
Private	3	100
Private	4	30
Private	5	28

Answer

```
1 def process():
2     # There will be at least one number on the board
3     while True:
4         # Extract the minimum number
5         n1 = board_min()
6         # Erase it from the board
7         board_erase(n1)
8         # If the board is empty; then our job is done
9         if board_isEmpty():
10             return(n1)
11             break
12         # As the board is not empty, there is one more number
13         n2 = board_min()
14         # Erase it
15         board_erase(n2)
16         # Find the difference
17         n3 = abs(n1 - n2)
18         # Write it on the board
19         board_write(n3)
20         # Loop back
```

Suffix Code Block

```
1 ## Prefix Code Block ##
2 def board_isEmpty():
3     return l == [ ]
4
5 def board_write(num):
6     l.append(num)
7
```

```
8 def board_erase(num):
9     l.remove(num)
10
11 def board_min():
12     return min(l)
13
14 test_cases = [ [1, 2, 3, 4, 5],
15                 [2, 4, 6, 8],
16                 [100],
17                 [120, 90],
18                 [10, 30, -4, -90, 100, 39, -4, 1, 10]
19             ]
20
21 test_case_id = int(input()) - 1
22 l = test_cases[test_case_id]
23 process()
24 assert l == [ ]
```

Week-6, Practice, Theory, Solution

Week-6, Practice, Theory, Solution

Problem 1

 Question

 Answer

 Solution

Problem 2

 Question

 Answer

 Solution

Problem 3

 Question

 Answer

 Solution

Problem 4

 Question

 Answer

 Solution

Problem 5

 Question

 Answers

 Solution

Problem 6

 Question

 Answer

 Solution

Problem 7

 Question

 Answer

 Solution

Problem 8

 Question

 Answer

 Solution

Problem 9

 Question

 Answer

 Solution

Problem 10

 Question

 Answers

 Solution

Problem 11

 Question

 Answer

 Solution

Problem 12

 Question

 Answer

 Solution

Problem 1

Question

What are the p and q representing in the following program?

```
1 def addone(index):
2     global q
3     if index in fruit:
4         fruit[index] += 1
5     else:
6         fruit[index] = 1
7         q += 1
8 fruit = {}
9 q = 0
10 l = ['Apple', 'Banana', 'Apple', 'Mango', 'Apple', 'Orange', 'Mango', 'Banana']
11 for x in l:
12     addone(x)
13 p = 0
14 for f in fruit:
15     p += fruit[f]
```

- (a) `p` represent number of types of fruits and `q` represent number of total fruits.
- (b) `p` represent number of total fruits and `q` represent number of types of fruits.
- (c) `p` and `q` both represent number of total fruits.
- (d) `p` and `q` both represent number of types of fruits

Answer

(b)

Solution

`p` represent number of total fruits that calculated in code line 13-15 where each fruit's value from dictionary `fruit` will be added in `p` and `q` represent number of types of fruits that are incremented by 1 inside function `addone` when new fruit name added in the dictionary `fruit`. Hence, Option b is correct

Problem 2

Question

```
1 s1={1, 2, 3, 4, 5}  
2 s2={3, 4, 5, 6, 7}  
3 ###Statement
```

Statement	Output
1. print(s1.difference(s2))	A. {1, 2, 3, 4, 5, 6, 7}
2. print(s1.intersection(s2))	B. {1, 2, 6, 7}
3. print(s1.union(s2))	C. {1, 2}
4. print(s1.symmetric_difference(s2))	D. {6, 7}
5. print(s2.difference(s1))	E. {3, 4, 5}

Select the correct match of statement and respective output

- (a) 1 - C , 2 - B , 3 - A , 4 - E , 5 - D
- (b) 1 - D , 2 - B , 3 - A , 4 - E , 5 - C
- (c) 1 - D , 2 - E , 3 - A , 4 - B , 5 - C
- (d) 1 - C , 2 - E , 3 - A , 4 - B , 5 - D

Answer

(d)

Solution

Correct match of statement and respective output is:-

Statement	Output
1. print(s1.difference(s2))	C. {1, 2}
2. print(s1.intersection(s2))	E. {3, 4, 5}
3. print(s1.union(s2))	A. {1, 2, 3, 4, 5, 6, 7}
4. print(s1.symmetric_difference(s2))	B. {1, 2, 6, 7}
5. print(s2.difference(s1))	D. {6, 7}

Problem 3

Question

Which one of the following denotes an empty set? It is a Multiple Select Question(MSQ).

- (a) `s = set()`
- (b) `s = {}`
- (c) `s = set([])`
- (d) `s = set('')`
- (e) `s = set(0)`

Answer

(a), (c) and (d)

Solution

option (b) is not correct because `{}` represents a dictionary and option (e) is not correct because `int` object is not `iterable` to convert in set. Other options are correct to denote an empty set.

Problem 4

Question

Assume that `i`, `s`, `l`, `t` and `st` are variables that have already been defined. All three Boolean expressions given below evaluate to `True`.

```
1 | type(i) == int
2 | type(s) == str
3 | type(l) == list
4 | type(t) == tuple
5 | type(st) == set
```

Which of the following snippets of code will execute without throwing an exception (runtime error)? It is a Multiple Select Question (MSQ).

(a)

```
1 | d = {i: 'int', s: 'string'}
```

(b)

```
1 | d = {s: len(s), i: st}
```

(c)

```
1 | d = {l: s, s: t}
```

(d)

```
1 | d = {i: l, st: len(l)}
```

(e)

```
1 | d = {t: l, s: t}
```

Answer

(a), (b)

Solution

Dictionary key can only be an immutable value like integer, string and tuple. Here option (c) is not correct because list(mutable) `l` can not be assigned as a key. option (d) is not correct because set(mutable) `st` can not be assign as a key and option (e) is not correct due to one corner case if tuple have list inside then it will give error `unhashable type: 'list'`.for example:

```
1 | t=([1,2,3],2)
2 | d={t:[1,2,3], 'A':t}
```

Problem 5

Question

Code

```
1 s1 = {1, 2, 3, 4, 4, 1, 2}
2 s2 = {4, 3, 2, 1}
3 l1 = [4, 3, 2, 1]
4 l2 = [1, 2, 3, 4, 4, 1, 2]
5 print(list(s1)==l2)
6 print(set(l1)==s2)
7 print(s1==s2)
```

(a)

```
1 True
2 True
3 True
```

(b)

```
1 False
2 True
3 True
```

(c)

```
1 False
2 False
3 True
```

(d)

```
1 False
2 True
3 False
```

Answers

(b)

Solution

`print(list(s1)==l2)` returns False because set removes all duplicate values so `list(s1)` is not equal to `l2`. `print(set(l1)==s2)` return True because after converting list `l1` into set , its elements are similar to `s2` . `print(s1==s2)` return True because set removes all duplicate elements then `s1` and `s2` have the same elements.

Problem 6

Question

Select all the correct options to remove "Orange" from the set. It is a Multiple Select Question(MSQ).

```
1 | sampleSet = {"Yellow", "Orange", "Black"}
```

- (a) `sampleSet.pop("Orange")`
- (b) `sampleSet.discard("Orange")`
- (c) `sampleSet.remove("Orange")`
- (d) `del sampleSet["Orange"]`

Answer

(b) and (c)

Solution

Option (a) is not correct because in set, `pop()` takes no arguments (here 1 argument is given) and option (d) is not correct because set object does not support item deletion by `del` keyword.

Problem 7

Question

Code-1

```
1 | d = {'virat' : 100, 'Dhoni' : 140, 'Rohit' : 60}
2 | d['sachin']=150
```

Code-2

```
1 | d = {'virat' : 100, 'Dhoni' : 140, 'Rohit' : 60}
2 | d['virat']=150
```

Which of the following statements are true about Code-1 and Code-2 . It is a Multiple Select Question(MSQ).

- (a) Code-1: Give error because `sachin` does not exist in dictionary d.
- (b) Code-2: Update the value of key(`virat`) in dictionary d.
- (c) Code-1: Add new item(key=`sachin` and value=`150`) in dictionary d.
- (d) Code-2: Add new item(key=`virat` and value=`150`) in dictionary d.
- (e) Code-2: Give error because `virat` already exists in dictionary d and its value can not be changed.

Answer

(b) and (c)

Solution

In Dictionary, when we assign value by `dict_name[key]=value` statement, if key already exists in dictionary then value will be updated by new value and if key does not exist in dictionary the new item will be added in dictionary with key and value. Hence, option (b) and (c) are correct.

Problem 8

Question

Select the correct match of method of Dictionary and respective description

Method	Description
1.items()	A. Returns a list of all the values in the dictionary
2.keys()	B. Returns a dictionary with the specified keys and value
3.values()	C. Returns the value of the specified key
4.get()	D. Returns a list containing a tuple for each key value pair
5.fromkeys()	E. Returns a list containing the dictionary's keys

(a) 1 - B , 2 - E , 3 - A , 4 - C , 5 - D

(b) 1 - D , 2 - E , 3 - C , 4 - A , 5 - B

(c) 1 - D , 2 - E , 3 - C , 4 - B , 5 - A

(d) 1 - D , 2 - E , 3 - A , 4 - C , 5 - B

Answer

(D)

Solution

Correct match of method of Dictionary and respective description:-

Method	Description
1.items()	D. Returns a list containing a tuple for each key value pair
2.keys()	E. Returns a list containing the dictionary's keys
3.values()	A. Returns a list of all the values in the dictionary
4.get()	C. Returns the value of the specified key
5.fromkeys()	B. Returns a dictionary with the specified keys and value

Problem 9

Question

Which of the following statements are true about Dictionaries in python. It is a Multiple Select Question(MSQ).

- (a) The values of a dictionary can be accessed using keys
- (b) The keys of a dictionary can be accessed using values
- (c) Keys must be immutable
- (d) Values of a dictionary must be unique
- (e) Duplicate keys are allowed in Dictionary
- (f) Dictionaries are mutable

Answer

- (a), (c), and (f)

Solution

Option (b) is incorrect because The keys of a dictionary can not be accessed using values. Option (d) is incorrect because the value of the dictionary can be duplicated and Option (e) is incorrect because the dictionary key can not be duplicated. Remaining options (a), (c), and (f) are true about the dictionary.

Problem 10

Question

Which of the following snippets of code will execute without throwing an exception (runtime error)? .if `t` is already initialized as a tuple. It is a Multiple Select Question(MSQ).

- (a) `t[3]=40`
- (b) `t.append(50)`.
- (c) `t.remove(30)`
- (d) `t = t * 3`
- (e) `t = t + (40 , 50)`
- (e) None of the above

Answers

(d), (e)

Solution

tuple is immutable and its value can not be changed after initializing. but in option (d) and (e) new tuple is initializing with value return by `t * 3` or `t + (40 , 50)`. For example:

```
1 | t=(1, 2)
2 | t=t*3
3 | print(t)
4 | t=t +(40 , 50)
5 | print(t)
```

Output

```
1 | (1, 2, 1, 2, 1, 2)
2 | (1, 2, 1, 2, 1, 2, 40, 50)
```

Problem 11

Question

What will be the output of the following code snippet

```
1 | D = {1 : 1, 2 : '2', '1' : 1, '2' : 3}
2 | D['1'] = 2
3 | print(D[D[D[str(D[1])]]])
```

- (a) 2
- (b) 3
- (c) '2'
- (d) Error

Answer

(b)

Solution

After updating `D['1']=2` in dictionary

```
for print(D[D[D[str(D[1])]]])
    str(D[1]) return '1'
    D['1'] return 2
    D[2] returns '2'
```

and finally

```
D['2'] returns 3
```

Hence, option (b) is correct

Problem 12

Question

What will be the output of the following code snippet

```
1 D = dict()
2 for i in range (3):
3     for j in range(2):
4         D[i] = j
5 print(D)
```

- (a) {0: 0, 1: 0, 2: 0}
- (b) {0: 1, 1: 1, 2: 1}
- (c) {0: 0, 1: 0, 2: 0, 0: 1, 1: 1, 2: 1}
- (d) TypeError

Answer

(b)

Solution

For each value of `i` the last value of `j` which is 1 will be the final update in the dictionary for key `i`. Hence option (b) is correct.

Week-6, Graded, Theory

Week-6, Graded, Theory

Problem-1

[Question-1 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-2 \[4 marks\]](#)

[Answer](#)

[Solution](#)

[Question-3 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-4 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-5 \[4 marks\]](#)

[Answer](#)

[Solution](#)

[Question-6 \[4 marks\]](#)

[Answer](#)

[Solution](#)

Problem-2

[Question-7 \[2 marks\]](#)

[Answer](#)

[Solution](#)

Problem 3

[Question-8 \[2 marks\]](#)

[Answers](#)

[Solution](#)

Problem 4

[Question-9 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-10 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-11 \[2 marks\]](#)

[Answer](#)

[Solution](#)

[Question-12 \[2 marks\]](#)

[Answer](#)

[Solution](#)

Problem-1

Random Number Wheel

Questions 1 to 6 are based on a common theme.

Question-1 [2 marks]

We wish to populate a list of 10000 integers, where each integer is drawn at random from the range 1 to 10, both endpoints included. Which of the following code snippets can be used to achieve this?

Note

- As an example, a list of 20 integers drawn at random from this range would look like this:

```
1 | [2, 1, 3, 5, 1, 4, 9, 5, 9, 4, 1, 3, 8, 6, 4, 7, 10, 1, 7, 5]
```

- The list shall be called `nums`. This list will be used in questions 1 to 6

(a)

```
1 | import random
2 |
3 | nums = []
4 | for i in range(10000):
5 |     nums.append(random.randint(1, 10))
```

(b)

```
1 | import random
2 |
3 | nums = []
4 | for i in range(10):
5 |     nums.append(random.randint(1, 10000))
```

(c)

```
1 | import random
2 |
3 | nums = []
4 | for i in range(10000):
5 |     nums.append(random.randint(0, 11))
```

(d)

```
1 | import random
2 |
3 | nums = []
4 | for i in range(10000):
5 |     nums.append(random.randint(1, 11))
```

Answer

(a)

Solution

```
1 import random
2
3 nums = []
4 for i in range(10000):
5     nums.append(random.randint(1, 10))
```

`random.randint(a, b)` returns a random number between `a` and `b`, both endpoints inclusive. The loop runs 10,000 times. Each time, a number in this range is added to `nums`.

Question-2 [4 marks]

Using the list `nums` obtained in the previous question, we wish to find the frequency of occurrence of each of the numbers in the range 1 to 10, endpoints included, in the list `nums`. If `P` is a dictionary that stores this information, which of the following snippets of code is/are an appropriate choice? It is a Multiple Select Question (MSQ).

Note

- As an example, consider a list of size 20.

```
1 | [2, 1, 3, 5, 1, 4, 9, 5, 9, 4, 1, 3, 8, 6, 4, 7, 10, 1, 7, 5]
```

Number	Frequency
1	4
2	1
3	2
...	...

- It is up to you to figure out what the keys and values represent in the dictionary `P` by scanning the options given below.

(a)

```
1 | P = { }
2 | for num in range(1, 10):
3 |     P[num] = 0
4 |
5 | for num in nums:
6 |     P[num] += 1
```

(b)

```
1 | P = { }
2 | for num in range(1, 11):
3 |     P[num] = 0
4 |
5 | for num in nums:
6 |     P[num] += 1
```

(c)

```
1 | P = { }
2 |
3 | for num in nums:
4 |     P[num] += 1
```

(d)

```
1 | P = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 0}
2 |
3 | for num in nums:
4 |     P[num] += 1
```

Answer

(b), (d)

Solution

Let us take one of the two solutions:

```
1 | P = { }
2 | for num in range(1, 11):
3 |     P[num] = 0
4 |
5 | for num in nums:
6 |     P[num] += 1
```

Here, an empty dict `P` is initialized in line-1. Then, lines 2 and 3 add the keys from 1 to 10 to `P`. The value corresponding to each of these keys is `0` to begin with. Lines 5-6 update the values corresponding to these keys. Every time the number `num` is encountered in the list `nums`, its count is incremented by 1.

Question-3 [2 marks]

Using the dictionary `P` obtained in the previous question and the list `nums` that was generated in question-1, consider the following code-block.

```
1 def check(P, N):
2     S = 0
3     for num in P:
4         S += P[num]
5     return S == N
6
7 print(check(P, len(nums)))
```

What is the output of this code block?

- (a) It is `True` if and only if `nums` is sorted in ascending order.
- (b) It is `True` if and only if `nums` is sorted in descending order.
- (c) It is always `True` and doesn't depend on the order in which elements appear in `nums`.
- (d) It is always `False` and doesn't depend on the order in which elements appear in `nums`.

Answer

(c)

Solution

`S` holds the sum of the frequency of occurrence of each of the numbers from 1 to 10 in `nums`. This sum is nothing but the total number of elements in `nums`. The function `check` is checking if this is indeed the case at line-5. So, it will always return `True`, irrespective of the order in which elements appear in `nums`.

Question-4 [2 marks]

A number is picked at random from the list `nums`. Which of the following expressions gives the probability of obtaining a `5`? Assume that any number in the list is equally likely to be chosen. Use the dictionary `P` obtained from question-2 if needed.

(a)

```
1 | nums[5] / 10000
```

(b)

```
1 | P[5] / 10000
```

(c)

```
1 | 5 / 10000
```

(d)

```
1 | P[5] / nums[5]
```

Answer

(b)

Solution

`P[5]` is the number of times the number 5 occurs in `nums`. There are 10,000 numbers in total. Therefore, `P[5] / 10000` is the required probability.

Question-5 [4 marks]

Let us continue with the list `nums` and dictionary `P`. We wish to find the number which occurs the most number of times in `nums`. `most_freq` is a function which accepts the dictionary `P` as input and returns the number in the list `nums` that has the greatest frequency. Select the correct code fragment to achieve this.

Note

- `most_freq` should return the number which has the greatest frequency and not the frequency itself.
- If multiple numbers have the same maximum frequency, then return the largest such number.

(a)

```
1 def most_freq(P):
2     freq_num, freq = 1, P[1]
3     for num in range(1, 11):
4         if P[num] >= freq:
5             freq_num, freq = num, P[num]
6     return freq
```

(b)

```
1 def most_freq(P):
2     freq_num, freq = 1, P[1]
3     for num in range(1, 11):
4         if P[num] >= freq_num:
5             freq_num, freq = num, P[num]
6     return freq_num
```

(c)

```
1 def most_freq(P):
2     freq_num, freq = 1, P[1]
3     for num in range(1, 11):
4         if P[num] > freq:
5             freq_num, freq = num, P[num]
6     return freq_num
```

(d)

```
1 def most_freq(P):
2     freq_num, freq = 1, P[1]
3     for num in range(1, 11):
4         if P[num] >= freq:
5             freq_num, freq = num, P[num]
6     return freq_num
```

Answer

(d)

Solution

Let us look at the correct answer:

```
1 | def most_freq(P):
2 |     freq_num, freq = 1, P[1]
3 |     for num in range(1, 11):
4 |         if P[num] >= freq:
5 |             freq_num, freq = num, P[num]
6 |     return freq_num
```

We see that `freq_num` holds the number that has the maximum frequency; `freq` holds the frequency of this particular number. We begin with `freq_num = 1` and `freq = P[1]`. In the loop, we go over each of the numbers from 1 to 10. If we see that for some `num`, `P[num]` exceeds the current maximum, then we update `freq_num` and `freq`. Notice that we use `>=` in the if-condition in line-4. This is because of the second point in the note. If there are two numbers which share the maximum frequency, then we need to return the greatest among them. Had we used `>` in line-4, we would have ended up returning the smallest number which has the maximum frequency.

Some points to note regarding the wrong answers:

- (a) is very similar to (d). (a) is wrong because it returns the frequency and not the number having the maximum frequency. This is as per the condition in note-1.
- (b) is wrong because, at line-4, it is checking if `P[num] >= freq_num`. It should be compared against `freq`.
- (c) is wrong because it uses `>` instead of `>=` at line-4.

Question-6 [4 marks]

Let us continue with the list `nums`. The list is said to feature a `streak` if the number `5` occurs at least five times in a row. `streak` is a function which accepts the list `nums` as input and returns `True` if it has a streak, and `False` otherwise. Select the most appropriate option. Some sample test cases for smaller lists are given below:

<code>nums</code>	<code>streak(nums)</code>
<code>[1, 2, 5, 5, 5, 5, 5, 1, 2, 10]</code>	<code>True</code>
<code>[4, 5, 5, 2, 5, 5, 5, 2, 5]</code>	<code>False</code>
<code>[10, 5, 5, 5, 5, 5, 5, 1, 6, 7, 8, 9]</code>	<code>True</code>

Note

- Remember that your function should work for the list `nums` that we generated in question-1.

(a)

```
1 def streak(nums):
2     if nums == [1, 2, 5, 5, 5, 5, 5, 1, 2, 10]:
3         return True
4     if nums == [4, 5, 5, 2, 5, 5, 5, 2, 5]:
5         return True
6     if nums == [10, 5, 5, 5, 5, 5, 5, 5, 1, 6, 7, 8, 9]:
7         return True
8     return False
```

(b)

```
1 def streak(nums):
2     if 5 not in nums:
3         return False
4     count = 0
5     for num in nums:
6         if num == 5:
7             count += 1
8             if count == 5:
9                 return True
10            else:
11                return False
12            else:
13                count = 0
14    return False
```

(c)

```

1 def streak(nums):
2     if 5 not in nums:
3         return False
4     count = 0
5     for num in nums:
6         if num == 5:
7             count += 1
8             if count == 5:
9                 return True
10            else:
11                count = 0
12    return False

```

(d)

```

1 def streak(nums):
2     if 5 not in nums:
3         return False
4     count = 0
5     for num in nums:
6         if num == 5:
7             count += 1
8             if count == 5:
9                 return True
10    return False

```

Answer

(c)

Solution

Let us look at the correct answer:

```

1 def streak(nums):
2     if 5 not in nums:
3         return False
4     count = 0
5     for num in nums:
6         if num == 5:
7             count += 1
8             if count == 5:
9                 return True
10            else:
11                count = 0
12    return False

```

If 5 is not even present in `nums`, then there is no point checking for a streak. This is the use of lines 2 and 3. The loop starting at line-5 goes through each number in `nums`. Whenever we come across a 5, `count` is incremented. If it is not a 5, then `count` is reset to zero. This happens in the `else` block at line-10. Notice that the `else` in line-10 is paired with the `if` at line-6. The nested `if` at line-8 checks if the `count` has become 5. If yes, then we have a streak. Finally, if the code manages to come to line-12, then there is certainly no streak. So, we have to return False there.

Let us look at why other answers are wrong:

- (a) It is obvious because it is checking for only the sample cases given in the question.
- (b) It is wrong because of the else block at line-10. If `count` is not 5, then it is immediately going to return False without waiting for it to reach 5 ever.
- (d) It is wrong because `count` is never reset to zero. So, even if 5 doesn't occur consecutively for five times, it will return True.

Problem-2

Question-7 [2 marks]

Which of the following declarations are valid for a Tuple? It is a Multiple Select Question (MSQ).

(a) `t = (1, 2, 3)`

(b) `t = (10)`

(c) `t = 20,`

(d) `t = tuple([10])`

(e) `t = tuple((10))`

(f) `t = 1, 2, 3`

Answer

(a), (c), (d) and (f)

Solution

In option (b) `t=(10)`, the type of t is `int` and option (e) `t = tuple((10))` gives error because `int` object is not iterable. Other than (b) and (e) all options, (a), (c), (d) and (f) are valid for a `tuple`.

Problem 3

Question-8 [2 marks]

Which of the following declarations are valid for a Set. It is a Multiple Select Question (MSQ).

- (a) `s = {}`
- (b) `s = set([1])`
- (c) `s = {1}`
- (d) `s = set()`
- (e) `s = set((1))`
- (f) `s = set(1)`

Answers

(b), (c) and (d)

Solution

In option (a), the type of `s` is `set` and Option (e) and (f) gives error because `int` object is not iterable. Other than (a), (e) and (f) all options (b), (c) and (d) are valid for a `set`.

Problem 4

Common data for questions 9 to 12.

Each student has a unique ID, which is an integer. For example, the student Ramanujan's ID is 1 while the student Ravi's ID is 2.

```
1 report_card = { 1:{  
2             'name': 'Ramanujan',  
3             'age': 18,  
4             'school': 'KV',  
5             'marks': {'Physics':75, 'Math':80, 'Chemistry':60}  
6         },  
7         2:{  
8             'name': 'Ravi',  
9             'age': 19,  
10            'school': 'KV',  
11            'marks': {'Physics':95, 'Math':70, 'Chemistry':90}  
12        }  
13     }
```

Question-9 [2 marks]

Choose the correct statement to print Ravi's marks in Math.

- (a) `print(report_card['Ravi']['marks']['Math'])`
- (b) `print(report_card[2][marks][Math])`
- (c) `print(report_card[2]['marks']['Math'])`
- (d) `print(report_card['2']['marks']['Math'])`

Answer

(c)

Solution

`print(report_card[2]['marks']['Math'])` is Correct way to access Ravi's marks in math.

Question-10 [2 marks]

Choose the correct statement to change the key `school` to `school_name` for both the students. It is a Multiple Select Question (MSQ).

(a)

```
1 | for i in range(1,3):
2 |     report_card[i]['school_name'] = report_card[i].popitem('school')
```

(b)

```
1 | for i in range(1,3):
2 |     report_card[i]['school_name'] = report_card[i].pop('school')
```

(c)

```
1 | report_card[1]['school_name'] = report_card[1].popitem('school')
2 | report_card[2]['school_name'] = report_card[2].popitem('school')
```

(d)

```
1 | for i in range(1,3):
2 |     report_card[i]['school_name'] = report_card[i].remove('school')
```

(e)

```
1 | report_card[1]['school_name'] = report_card[1].pop('school')
2 | report_card[2]['school_name'] = report_card[2].pop('school')
```

Answer

(b) and (e)

Solution

`pop(key)` method is used for removing items from the dictionary and after removing it return corresponding value. So option (b) and (e) are the correct way to change the key `school` to `school_name` for both the students. Here the return value of `pop('school')` will be assigned to the new key `school_name`.

Question-11 [2 marks]

Choose the correct statement to add a new entry of student `Ram` whose student ID is 3. It is a Multiple Select Question (MSQ). Assume that we go back to the dictionary given in the common data. So, we will be using `school` and not `school_name` in this particular question.

Entry	Details
name	Ram
age	20
school	KV
marks	Physics: 96 Math: 85 Chemistry: 75

(a)

```
1 | report_card.add({3:{'name':'Ram', 'age': 20, 'school':'KV', 'marks':  
2 |     {'Physics':96, 'Math':85, 'Chemistry':75}}})
```

(b)

```
1 | report_card[3]={ 'name': 'Ram', 'age': 20, 'school': 'KV', 'marks':  
|     {'Physics':96, 'Math':85, 'Chemistry':75}}
```

(c)

```
1 | report_card={3:{'name':'Ram', 'age': 20, 'school':'KV', 'marks':  
|     {'Physics':96, 'Math':85, 'Chemistry':75}}}
```

(d)

```
1 | report_card.update({3:{'name':'Ram', 'age': 20, 'school':'KV', 'marks':  
2 |     {'Physics':96, 'Math':85, 'Chemistry':75}}})
```

(e)

```
1 | report_card[3].update({'name':'Ram', 'age': 20, 'school':'KV', 'marks':  
2 |     {'Physics':96, 'Math':85, 'Chemistry':75}})
```

Answer

(b) and (d)

Solution

The `update()` method inserts the specified items to the dictionary. The specified items can be a dictionary, or an iterable object with key value pairs or we can assign a new value with a new key like `dict[key]=value` to add new items in the dictionary. Hence, option (b) and (d) are correct.

Question-12 [2 marks]

Choose the correct statement to delete the all records of student Ravi. It is a Multiple Select Question (MSQ).

(a)

```
1 | report_card.pop(2)
```

(b)

```
1 | report_card.popitem(2)
```

(c)

```
1 | report_card.remove(2)
```

(d)

```
1 | del report_card[2]
```

(e)

```
1 | report_card.discard(2)
```

Answer

(a) and (d)

Solution

Option (a) and (d) are correct ways to remove all information about Ravi.

`report_card.popitem(2)` gives an error because `popitem()` takes no argument.

`report_card.remove(2)` gives error because `remove()` is not supported by

`dictionary.report_card.discard(2)` gives error because `discard()` is not supported by dictionary.

Week 6 - Practice Programming

Week 6 - Practice Programming

Problem 1

Test Cases

 Suffix Code Block - Hidden

Answer

Problem 2

Question

 Suffix Code Block - Hidden

Test Cases

Answer

Problem 3

 Prefix Code Block - Hidden

 Suffix Code Block - Visible

Test Cases

Answer

Problem 4

Question

Sample Input and Output

 Prefix Code Block - Hidden

 Suffix Code Block - Visible

 Public

 Private

Answer

Problem 5

Question

 Prefix Code Block - Hidden

 Suffix Code Block - Visible

Test Cases

Answer#1

Alternative Answer#2

Alternative Answer#3

Alternative Answer#4

Problem 1

Suppose a crowd is waiting outside a shop counter forming a waiting line for collecting valuables. Each person holds a piece of paper with a order number written on it. One of the person at a given position leaves the queue for some reason and joins at the end of the queue.

Please write a python program `shift_a_person` to imitate this scenario.

1. The function has two parameters :
 - o A list named `order_list` of datatype `list` consists of positive integers representing the order number of each person in the list and
 - o A variable `position` of type `int` indicating the initial position of a person to be shifted at the end of the list. The position is counted from left to right of the list starting from `1`.
2. The person located at the `position` is moved to the end of the list. This means the corresponding order number becomes the rightmost element of the updated list. If the `position` is an invalid number (less than 1 or more than the length of the list), it should be considered the first person standing in the list from the left.
3. If two or more persons have the same order number, a person holding duplicate order number is removed from the beginning of the list until every person is left with a unique order number.
4. In the function point-2 is followed by point-3, this means they are executed in the same order as they are listed. The updated list is returned by the function.

```
1 | def shift_a_person(order_list, position = 1):  
2 |     # write function body  
3 |     #  
4 |     #
```

Test Cases

Type	Input	Output
Public	1 2 3 4 5 6 7 8 9 6	[1, 2, 3, 4, 5, 7, 8, 9, 6]
Public	1 2 3 4 5 6 7 2 9 6	[1, 3, 4, 5, 7, 2, 9, 6]
Private	1 3 4 5 6 7 7 8 3	[1, 3, 5, 6, 7, 8, 4]
Private	27 73 13 98 56 37 9	[73, 13, 98, 56, 37, 27]
Private	27 73 13 98 56 37 13 13 2	[27, 98, 56, 37, 13, 73]
Private	27 27 27 27 2	[27]

Suffix Code Block - Hidden

```
1 | order_list = list(map(int, input().split(" ")))
2 | position = int(input())
3 | print(shift_a_person(order_list, position))
```

Answer

```
1 | def shift_a_person(order_list, position = 1):
2 |     # position set to 1 if it is out of bound [1, len(order_list)]
3 |     if position not in range(1, len(order_list) + 1):
4 |         position = 1
5 |     # create a new list by slicing original list
6 |     # and append element indicated by position to the end of this list
7 |     new_order_list = order_list[:position-1] + order_list[position:] +
8 |         order_list[position-1:position]
9 |     # remove an element if it occurs multiple times till only first
10 |    occurrence are left
11 |    for num in new_order_list[:]: # new_order_list[:] is creating a copy of
12 |        new_order_list
13 |        if new_order_list.count(num) > 1:
14 |            new_order_list.remove(num)
15 |
16 |    return new_order_list
```

Problem 2

Question

You are given data in the form described in the first column of the below table. The data is stored as a list of dictionaries. Each dictionary item is a list of values that represents a data column. This is called a `tabular form (or column measure)` of data.

The second row of the table shows one sample test case input and output.

Input form (tabular form)	Output form (record form)
<pre>[{col_A: [A_val_1, A_val_2, A_val_3,...]}, {col_B: [B_val_1, B_val_2, B_val_3,...]}, ]</pre>	<pre>[{col_A: A_val_1, col_B: B_val_1, col_C: C_val_1,}, {col_A: A_val_2, col_B: B_val_2, col_C: C_val_3,}, ]</pre>
<pre>[{'Roll' : [1, 2, 3]}, {'Name' : ['A', 'B', 'C']}, {'Course' : ['x', 'Y', 'z']}]</pre>	<pre>[{'Roll': 1, 'Name': 'A', 'Course': 'x'}, {'Roll': 2, 'Name': 'B', 'Course': 'Y'}, {'Roll': 3, 'Name': 'C', 'Course': 'z'}]</pre>

You are asked to convert this data into another form called `record form (or row measure)`. This is also a list of dictionaries. Here, a dictionary element is composed of data values from each data columns of `tabular form`.

Write a generic python function `table_to_record` that can take data in the `tabular` form and return the data in `record form`. Please see a public test case for an example input and output.

The program validates the function `table_to_record` with different data inputs based on the test case number.

```
1 def table_to_record(table_db):  
2     # write function body  
3     #  
4     #
```

Suffix Code Block - Hidden

```
1 # test cases(key) and input test data  
2 table_db = {  
3     1: [{"Roll" : [1, 2, 3]}, {"Name" : ['A', 'B', 'C']}, {"Course" : ['x', 'Y',  
  'z']}],  
4     2: [{"A" : [1, 2, 3]}, {"B" : ["1", "2", "3"]}, {"C" : [1, 2.0, "abc"]}],  
5     3: [{1 : [1, 2, 3]}, {2 : ["A", "B", "C"]}, {3 : ["X", "Y", "Z"]}],  
6     4: [{"roll": 100, 200, 300}, {"name": ["Joy", "Anand", "Sunita"]}, {"city":  
  ["Delhi", "Kochi", "Pune"]}]  
7 }  
8  
9 print(table_to_record(table_db[int(input())]))
```

Test Cases

Type	Input	data	Output
Public	1	[{'Roll' : [1, 2, 3]}, {'Name' : ['A', 'B', 'C']}, {'Course' : ['X', 'Y', 'Z']}]	[{'Roll': 1, 'Name': 'A', 'Course': 'X'}, {'Roll': 2, 'Name': 'B', 'Course': 'Y'}, {'Roll': 3, 'Name': 'C', 'Course': 'Z'}]
Private	2	[{"A" : [1, 2, 3]}, {"B" : ["1", "2", "3"]}, {"C" : [1, 2.0, "abc"]}]]	[{'A': 1, 'B': '1', 'C': 1}, {'A': 2, 'B': '2', 'C': 2.0}, {'A': 3, 'B': '3', 'C': 'abc'}]
Private	3	[{1 : [1, 2, 3]}, {2 : ['A', 'B', 'C']}, {3 : ['X', 'Y', 'Z']}]]	[{1: 1, 2: 'A', 3: 'X'}, {1: 2, 2: 'B', 3: 'Y'}, {1: 3, 2: 'C', 3: 'Z'}]
Private	4	[{"roll": [100, 200, 300]}, {"name": ["Joy", "Anand", "Sunita"]}, {"city": ["Delhi", "Kochi", "Pune"]}]]	[{'roll': 100, 'name': 'Joy', 'city': 'Delhi'}, {'roll': 200, 'name': 'Anand', 'city': 'Kochi'}, {'roll': 300, 'name': 'Sunita', 'city': 'Pune'}]

Answer

```

1 def table_to_record(table_db):
2     # creating a list of empty dictionaries
3     record_db = []
4     for i in range(len(table_db)):
5         record_db.append(dict())
6
7     # using list comprehension
8     # record_db =[{} for idx in range(len(table_db))] # using list
comprehension
9
10    # adding value from each column of table_db to the dictionary element
11    idx = 0
12    for data in table_db:
13        for key, value_list in data.items():
14            for ele in value_list:
15                record_db[idx][key] = ele
16                idx += 1
17            idx = 0
18    return(record_db)

```

Problem 3

Write a program to accept entity details from user and store in a dictionary of dictionary database say `data_table`.

A sample entity details can be represented as: `entity_dict = {"uid": entity_id, "a": a_value, "b": b_value, ...}`.

The key (or field) `"uid"` of the dictionary `entity_dict` is mandatory and its value `entity_id` is always a unique positive integer. This field's value is used as a key in the `data_table` to store entity details record `entity_dict`.

In an entity record dictionary, all keys are of string type while values except `entity_id` can be either string or number.

Functions	Uses
<code>add_record(data_table, entity_dict)</code>	Adds an entity details <code>entity_dict</code> to the entity database <code>data_table</code> and return this modified <code>data_table</code> .
<code>search_record(data_table, entity_id)</code>	Returns entity details as a dictionary (similar to <code>entity_dict</code>), if entity <code>uid</code> is found, otherwise, it returns <code>None</code> .
<code>update_record(data_table, entity_id, key, value)</code>	Updates specific key's value of an entity record if found and return the updated <code>data_table</code> . If not record found, nothing is updated, return the <code>data_table</code> as it is.
<code>delete_record(data_table, entity_id)</code>	Deletes a specific entity record and return the modified <code>data_table</code> . If the record is not found, nothing is deleted, return the <code>data_table</code> as it is.

The database `data_table` is empty at the beginning and these functions are called in the same order as they are listed. Write a program to implement the body of these functions.

```
1 def add_record(data_table, entity_dict):
2     # write function body
3     #
4     #
5 def search_record(data_table, entity_id):
6     # write function body
7     #
8     #
9 def update_record(data_table, entity_id, key, value):
10    # write function body
11    #
12    #
13 def delete_record(data_table, entity_id):
14    # write function body
15    #
16    #
```

Prefix Code Block - Hidden

```
1 # uses json library
2 # convert an input dictionary string to dictionary object
3 import json
4 entity_dict = json.loads(input())
5 # accept entity_id as int input used in json dictionary string
6 entity_id = int(input())
```

Suffix Code Block - Visible

```
1 # create a blank dictionary
2 data_table = {}
3 # add a record
4 data_table = add_record(data_table, entity_dict)
5 print(data_table)
6 # search a record
7 print(search_record(data_table, entity_id))
8 # update a field in the existing record
9 data_table = update_record(data_table, entity_id,
10 list(data_table[entity_id].keys())[1], "X")
11 print(data_table)
12 # delete a record
13 data_table = delete_record(data_table, entity_id)
14 print(data_table)
```

Test Cases

Type	Input	Output
Public	{"uid": 101, "emp_name": "Hari", "emp_sal": 100000} 101	{101: {"uid": 101, "emp_name": "Hari", "emp_sal": 100000}} {101: {"uid": 101, "emp_name": "Hari", "emp_sal": 100000}} {101: {"uid": 101, "emp_name": "X", "emp_sal": 100000}} {}
Private	{"uid": 1, "book_name": "Python", "book_author": "Guido van Rossum"} 1	{1: {"uid": 1, "book_name": "Python", "book_author": "Guido van Rossum"} {1: {"uid": 1, "book_name": "Python", "book_author": "Guido van Rossum"} {1: {"uid": 1, "book_name": "X", "book_author": "Guido van Rossum"} {}}
Private	{"uid": 1, "A": 1, "B": 1, "C": 1, "D": 1} 1	{1: {"uid": 1, "A": 1, "B": 1, "C": 1, "D": 1}} {1: {"uid": 1, "A": 1, "B": 1, "C": 1, "D": 1}} {1: {"uid": 1, "A": "X", "B": 1, "C": 1, "D": 1}} {}}
Private	{"uid": 1, "A": 1, "B": 1.0, "C": 1, "D": "1"} 1	{1: {"uid": 1, "A": 1, "B": 1.0, "C": 1, "D": "1"} {1: {"uid": 1, "A": 1, "B": 1.0, "C": 1, "D": "1"} {1: {"uid": 1, "A": "X", "B": 1.0, "C": 1, "D": "1"} {}}

Answer

```

1 # Adds an entity details `entity_dict` to the entity database `data_table`  

2 # and return this modified `data_table`.  

3 def add_record(data_table, entity_dict):  

4     data_table[list(entity_dict.values())[0]] = entity_dict  

5     return data_table  

6  

7 # Returns entity details as a dictionary (similar to `entity_dict`), if  

8 # entity `uid` is found,  

9 # otherwise, it returns `None`.  

10 def search_record(data_table, entity_id):  

11     if entity_id in data_table.keys():  

12         return data_table[entity_id]  

13     else:  

14         return None  

15  

16 # Updates specific key's value of an entity record if found and return the  

17 # updated `data_table`.  

18 # If not record found, nothing is updated, return the `data_table` as it is.  

19  

20 def update_record(data_table, entity_id, key, value):  

21     if entity_id in data_table.keys():  

22         data_table[entity_id][key] = value  

23     return data_table

```

```
22 # Deletes a specific entity record and return the modified `data_table`.  
23 # If the record is not found, nothing is deleted, return the `data_table` as  
it is.  
24 def delete_record(data_table, entity_id):  
25     # using membership  
26     if entity_id in data_table:  
27         del data_table[entity_id]  
28     return data_table  
29  
30     # using dictionary method pop()  
31     # data_table.pop(entity_id, None)  
32     # return data_table  
33  
34     # Another approach  
35     # new_dict = {}  
36     # for key in data_table.keys():  
37     #     if key != entity_id:  
38     #         new_dict[key] = data_table[key]  
39     # return new_dict
```

Problem 4

Question

Given below are the list of dictionary methods and their uses.

Function	Uses
<code>min_dict_key(data)</code>	Returns the minimum key in the dictionary <code>data</code>
<code>max_dict_key(data)</code>	Returns the maximum key in the dictionary <code>data</code>
<code>min_value_dict_key(data)</code>	Returns the key corresponding to the minimum value in the dictionary <code>data</code> , if there are two items with equal values, the key to first item is returned
<code>max_value_dict_key(data)</code>	Returns the key corresponding to the maximum value in the dictionary <code>data</code> , if there are two items with equal values, the key to first item is returned
<code>sort_by_key(data, order="asc")</code>	Returns the list of tuples <code>[(key, value), (key value), ...]</code> . Each tuple in this list represents an item from the dictionary <code>data</code> sorted by its keys. The order of sorting is given by the parameter <code>order</code> which can be either <code>"asc"</code> or <code>"desc"</code> with <code>"asc"</code> being default value. if there are two items with equal values, their order is preserved while sorting the items.
<code>sort_by_value(data, order="asc")</code>	Returns the list of tuples <code>[(key, value), (key value), ...]</code> . Each tuple in this list represents an item from the dictionary <code>data</code> sorted by its values. The order of sorting is given by the parameter <code>order</code> which can be either <code>"asc"</code> or <code>"desc"</code> with <code>"asc"</code> being default value. if there are two items with equal values, their order is preserved while sorting the items.

Write the function body for these dictionary methods. The output is displayed per function call in the **suffix** code block. The program validates the these function with different data inputs based on the test case number.

Sample Input and Output

Input	data	Output
1	{1:40, 2:39, 3:20, 4:33, 5:36}	<p>1 5 3 1</p> <p>[(1, 40), (2, 39), (3, 20), (4, 33), (5, 36)] [(5, 36), (4, 33), (3, 20), (2, 39), (1, 40)] [(3, 20), (4, 33), (5, 36), (2, 39), (1, 40)] [(1, 40), (2, 39), (5, 36), (4, 33), (3, 20)]</p>
2	{4:40, 5:39, 3:20, 6:33, 2:36}	<p>2 6 3 4</p> <p>[(2, 36), (3, 20), (4, 40), (5, 39), (6, 33)] [(6, 33), (5, 39), (4, 40), (3, 20), (2, 36)] [(3, 20), (6, 33), (2, 36), (5, 39), (4, 40)] [(4, 40), (5, 39), (2, 36), (6, 33), (3, 20)]</p>

```

1 def min_dict_key(data):
2     # write function body
3     #
4     #
5 def max_dict_key(data):
6     # write function body
7     #
8     #
9 def min_value_dict_key(data):
10    # write function body
11    #
12    #
13 def max_value_dict_key(data):
14    # write function body
15    #
16    #
17 def sort_by_key(data, order="asc"):
18    # write function body
19    #
20    #
21 def sort_by_value(data, order="asc"):
22    # write function body
23    #
24    #

```

Prefix Code Block - Hidden

```

1 # test cases(key) and input test data
2 table = {
3     1: {1:40, 2:39, 3:20, 4:33, 5:36},
4     2: {4:40, 5:39, 3:20, 6:33, 2:36},
5     3: {1:40, 2:20, 3:20, 4:36, 5:36},
6     4: {5:40, 4:20, 3:20, 2:36, 1:36},
7     5: {5:5, 4:4, 3:3, 2:2, 1:1},
8     6: {1:5, 2:4, 3:3, 4:2, 5:1}
9 }
10 data = table[int(input())]

```

Suffix Code Block - Visible

```

1 print(min_dict_key(data))
2 print(max_dict_key(data))
3 print(min_value_dict_key(data))
4 print(max_value_dict_key(data))
5 print(sort_by_key(data, "asc"))
6 print(sort_by_key(data, "desc"))
7 print(sort_by_value(data, "asc"))
8 print(sort_by_value(data, "desc"))

```

Public

Input	data	Output
1	{1:40, 2:39, 3:20, 4:33, 5:36}	1 5 3 1 [(1, 40), (2, 39), (3, 20), (4, 33), (5, 36)] ([(5, 36), (4, 33), (3, 20), (2, 39), (1, 40)]) ([(3, 20), (4, 33), (5, 36), (2, 39), (1, 40)]) ([(1, 40), (2, 39), (5, 36), (4, 33), (3, 20)])
2	{4:40, 5:39, 3:20, 6:33, 2:36}	2 6 3 4 [(2, 36), (3, 20), (4, 40), (5, 39), (6, 33)] ([(6, 33), (5, 39), (4, 40), (3, 20), (2, 36)]) ([(3, 20), (6, 33), (2, 36), (5, 39), (4, 40)]) ([(4, 40), (5, 39), (2, 36), (6, 33), (3, 20)])

Private

Input	data	Program Output
3	{1:40, 2:20, 3:20, 4:36, 5:36}	1 5 2 1 [(1, 40), (2, 20), (3, 20), (4, 36), (5, 36)] [(5, 36), (4, 36), (3, 20), (2, 20), (1, 40)] [(2, 20), (3, 20), (4, 36), (5, 36), (1, 40)] [(1, 40), (4, 36), (5, 36), (2, 20), (3, 20)]
4	{5:40, 4:20, 3:20, 2:36, 1:36}	1 5 4 5 [(1, 36), (2, 36), (3, 20), (4, 20), (5, 40)] [(5, 40), (4, 20), (3, 20), (2, 36), (1, 36)] [(4, 20), (3, 20), (2, 36), (1, 36), (5, 40)] [(5, 40), (2, 36), (1, 36), (4, 20), (3, 20)]
5	{5:5, 4:4, 3:3, 2:2, 1:1}	1 5 1 5 [(1, 1), (2, 2), (3, 3), (4, 4), (5, 5)] [(5, 5), (4, 4), (3, 3), (2, 2), (1, 1)] [(1, 1), (2, 2), (3, 3), (4, 4), (5, 5)] [(5, 5), (4, 4), (3, 3), (2, 2), (1, 1)]
6	{1:5, 2:4, 3:3, 4:2, 5:1}	1 5 5 1 [(1, 5), (2, 4), (3, 3), (4, 2), (5, 1)] [(5, 1), (4, 2), (3, 3), (2, 4), (1, 5)] [(5, 1), (4, 2), (3, 3), (2, 4), (1, 5)] [(1, 5), (2, 4), (3, 3), (4, 2), (5, 1)]

Answer

```

1 # finds the item with minimum key in a given dictionary
2 def min_dict_key(data):
3     min_key, min_key_value = list(data.items())[0]
4     for key, value in data.items():
5         # store item with minimum key into min_key
6         if min_key > key:
7             min_key = key
8     return min_key
9
10 # finds the item with maximum key in a given dictionary
11 def max_dict_key(data):
12     max_key, max_key_value = list(data.items())[0]
13     for key, value in data.items():
14         # store item with maximum key into min_key

```

```

15     if max_key < key:
16         max_key = key
17     return max_key
18
19 # finds the item key with minimum value in a given dictionary
20 def min_value_dict_key(data):
21     min_value_key, min_value = list(data.items())[0]
22     for key, value in data.items():
23         # store item with minimum key into min_value_key and min_value
24         if min_value > value:
25             min_value = value
26             min_value_key = key
27     return min_value_key
28
29 # finds the item key with maximum value in a given dictionary
30 def max_value_dict_key(data):
31     max_value_key, max_value = list(data.items())[0]
32     for key, value in data.items():
33         # store item with maximum value into max_value_key and max_value
34         if max_value < value:
35             max_value = value
36             max_value_key = key
37     return max_value_key
38
39 # return a list of tuples where tuples are sorted items of data dictionary
40 # by key
41 def sort_by_key(data, order="asc"):
42     data = dict(data)
43     sorted_dict = {}
44     sorted_list = []
45     n = len(data.keys())
46     for _ in range(n):
47         # iteratively removing item from data dictionary with maximum key
48         # and appending to a list as tuple, it gives descending items list
49         # by key
50         if order == "desc":
51             max_key = max_value_dict_key(data)
52             sorted_dict[max_key] = data[max_key]
53             sorted_list.append((max_key, data[max_key]))
54             del data[max_key]
55         # iteratively removing item from data dictionary with minimum key
56         # and appending to a list as tuple, it gives ascending items list by
57         # key
58         elif order == "asc":
59             min_key = min_value_dict_key(data)
60             sorted_dict[min_key] = data[min_key]
61             sorted_list.append((min_key, data[min_key]))
62             del data[min_key]
63     #return sorted_dict # returns sorted dictionary, for python version >=
64     # 3.7
65     return sorted_list # returns sorted list
66
67 # return a list of tuples where tuples are sorted items of data dictionary
68 # by value
69 def sort_by_value(data, order="asc"):
70     data = dict(data)
71     sorted_dict = {}
72     sorted_list = []

```

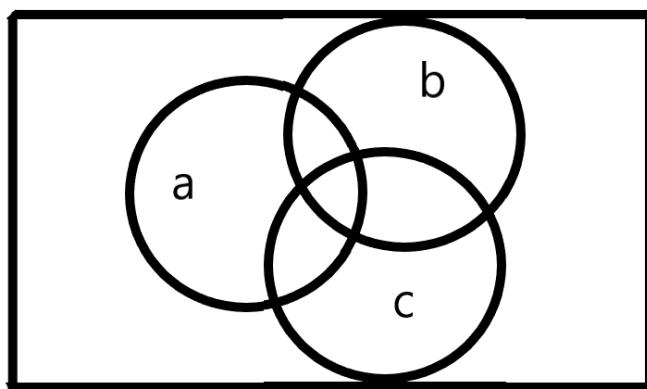
```
68     n = len(data.keys())
69     for _ in range(n):
70         # iteratively removing item from data dictionary with maximum key
71         # and appending to a list as tuple, it gives descending items list
72         by value
73         if order == "desc":
74             max_value_key = max_value_dict_key(data)
75             sorted_dict[max_value_key] = data[max_value_key]
76             sorted_list.append((max_value_key, data[max_value_key]))
77             del data[max_value_key]
78         # iteratively removing item from data dictionary with minimum key
79         # and appending to a list as tuple, it gives ascending items list by
80         value
81         elif order == "asc":
82             min_value_key = min_value_dict_key(data)
83             sorted_dict[min_value_key] = data[min_value_key]
84             sorted_list.append((min_value_key, data[min_value_key]))
85             del data[min_value_key]
#return sorted_dict # returns sorted dictionary, for python version >=
3.7
85     return sorted_list # returns sorted list
```

Problem 5

Question

Among a group of students, some play sport **a**, some play **b**, and some play **c**.

Id	Description	Set Symbol	Count of Students
1	played a	a	$n(a)$
2	played b	b	$n(b)$
3	played c	c	$n(c)$
4	played both a and b	a_and_b	$n(a_and_b)$
5	played both b and c	a_and_c	$n(a_and_b)$
6	played a and c	b_and_c	$n(b_and_c)$
7	played a and b and c	a_and_b_and_c	$n(a_and_b_and_c)$
8	played any three sports (a set of all students)	a_or_b_or_c	$n(a_or_b_or_c)$
9	played only a and b not c	only_a_and_b_not_c	$n(only_a_and_b_not_c)$
10	played only a and c not b	only_a_and_c_not_b	$n(only_a_and_c_not_b)$
11	played only b and c not a	only_b_and_c_not_a	$n(only_b_and_c_not_a)$
12	played only a	only_a	$n(only_a)$
13	played only b	only_b	$n(only_b)$
14	played only c	only_c	$n(only_c)$



The probability of an event **e** on given sample space **s** is defined as :

$$P(e) = \frac{n(e)}{n(s)}$$

where `n(a)` is the count of elements in the set `a` and `n(s)` is the total number of elements in the sample space `s`. In above table, the sample space `s` is represented by the set `a_or_b_or_c`. It gives the total number of students in the group.

Write a program to print the count of elements and probability of all 14 sets given in the above table in the sequence they are listed. The input is three sequence, one for each set `a`, `b` and `c` in the order, see the **suffix** code block. The values in each sequence is separated by comma.

The **output** is the element count and the probability separated by comma for a set. The output for each of the 14 sets are given on the newline. The probability value should be rounded up to 1 decimal point. Please see a public test case for example.

You can use library functions / operators or write custom set functions to achieve the task.

Note: Elements in each set represents a unique student, it can be a number or a string.

Prefix Code Block - Hidden

```
1 # accepts two sets and returns its intersection
2 def set_intersection(x, y):
3     common_set = set()
4     # iterate through the elements in the smaller set and check common
5     # elements in both set x and y
6     if len(x) <= len(y):
7         for item in x:
8             if item in y:
9                 common_set.add(item)
10    else:
11        for item in y:
12            if item in x:
13                common_set.add(item)
14    return common_set
15
16 # # accepts two sets and returns its intersection using union of set
17 # def set_intersection(x, y):
18 # # iterate throught the elements in union set and check common elements in
19 # both set x and y
20 #     union_set = set(list(x) + list(y))
21 #     common_set = set()
22 #     for item in union_set:
23 #         if (item in x) and (item in y):
24 #             common_set.add(item)
25 #     return common_set
26
27 # accepts two sets and returns its union
28 def set_union(x, y):
29     # converts set to list, appends and returns as a set
30     return set(list(x) + list(y))
31
32 # accepts two sets and returns difference of second set from first set
33 def set_difference(x, y):
34     asymmetric_diff_set = set()
35     # adds elements which are in set x and not in y
36     for item in x:
37         if item not in y:
38             asymmetric_diff_set.add(item)
39     return asymmetric_diff_set
```

```

39 # accepts two sets and returns its symmetric set difference
40 def set_symmetric_difference(x, y):
41     # create a union set from input sets
42     union_set = set(list(x) + list(y))
43     symmetric_diff_set = set()
44     for item in union_set:
45         # adds elements which are in set x and not in y
46         if item in x and item not in y:
47             symmetric_diff_set.add(item)
48         # adds elements which are not in set x but in y
49         elif item not in x and item in y:
50             symmetric_diff_set.add(item)
51     return symmetric_diff_set
52
53 # custom functions defined for set operations, you may use library functions
54 # instead of these
55 # accepts multiple parameters and returns its intersection
56 def set_intersection_generic(*args): # *args used to accept a list of
parameters of variable length
57     union_set = set_union_generic(*args) # args[0] -> first param, *args[1]
-> second param and so on
58     common_set = set()
59     for item in union_set:
60         flag = True
61         # if an element is present in all sets, add it to common set
62         for argument in args:
63             if item not in argument:
64                 flag = False
65         if flag:
66             common_set.add(item)
67     return common_set
68
69 # accepts multiple parameters and returns its union
70 def set_union_generic(*args): # *args used to accept a list of parameters of
variable length
71     union_list = []
72     # make a large list with elements from all input sets
73     for argument in args: # args[0] -> first param, *args[1] -> second param
and so on
74         union_list = union_list + list(argument)
75     # return a set of this large list
76     return set(union_list)

```

Suffix Code Block - Visible

```

1 a = set(input().split(","))
2 b = set(input().split(","))
3 c = set(input().split(","))

```

Test Cases

type	Input	Output
Public	1,4,7,10 1,3,5,7,9,10 1,2,4,6,8,10	4,0.4 6,0.6 6,0.6 3,0.3 2,0.2 3,0.3 2,0.2 10,1,0 1,0.1 1,0.1 0,0.0 0,0.0 3,0.3 3,0.3
Private	a,b,c,f,h b,c,f,g,h,j a,c,e,f,g,k	5,0.6 6,0.7 6,0.7 4,0.4 3,0.3 3,0.3 2,0.2 9,1,0 2,0.2 1,0.1 1,0.1 0,0.0 1,0.1 2,0.2
Private	1,1,4,4,5,5 1,1,5,5,9,10 1,1,4,4,5,5	3,0.6 4,0.8 3,0.6 2,0.4 2,0.4 3,0.6 2,0.4 5,1,0 0,0.0 1,0.2 0,0.0 0,0.0 2,0.4 0,0.0

Answer#1

```
1 # using library functions
2 # finding the expected set using set operations on sets a, b, and c
3 a_and_b = a & b
4 b_and_c = b & c
5 a_and_c = a & c
6 a_and_b_and_c = a & b & c
7 a_or_b_or_c = a | b | c
8 only_a_and_b_not_c = (a & b) - (a & b & c)
9 only_a_and_c_not_b = (a & c) - (a & b & c)
10 only_b_and_c_not_a = (b & c) - (a & b & c)
11 only_a = ((a - b) - c)
12 only_b = ((b - a) - c)
13 only_c = ((c - a) - b)
14
15 # dictionary of all sets
16 set_dict = {
17     "a": a,
18     "b": b,
19     "c": c,
20     "a_and_b": a_and_b,
21     "b_and_c": b_and_c,
22     "a_and_c": a_and_c,
23     "a_and_b_and_c": a_and_b_and_c,
24     "a_or_b_or_c": a_or_b_or_c,
25     "only_a_and_b_not_c": only_a_and_b_not_c ,
26     "only_a_and_c_not_b": only_a_and_c_not_b ,
27     "only_b_and_c_not_a": only_b_and_c_not_a,
28     "only_a": only_a,
29     "only_b": only_b,
30     "only_c": only_c
31 }
32
33 # printing the length and probability of event represented by each set in
34 # set_dict
35 for set_name, set_val in set_dict.items():
36     #print(f"\n{set_name} = {len(set_val)}, prob({set_name}) = "
37     #      f"{len(set_val)/len(a_or_b_or_c):.1f}")
38     print(f"\n{len(set_val)},{len(set_val)/len(a_or_b_or_c):.1f}")
```

Alternative Answer#2

```
1 # using library functions
2 # finding the expected set using set operations on sets a, b, and c
3 a_and_b = a & b
4 b_and_c = b & c
5 a_and_c = a & c
6 a_and_b_and_c = a & b & c
7 a_or_b_or_c = a | b | c
8 only_a_and_b_not_c = (a & b) - (a & b & c)
9 only_a_and_c_not_b = (a & c) - (a & b & c)
10 only_b_and_c_not_a = (b & c) - (a & b & c)
11 only_a = ((a - b) - c)
12 only_b = ((b - a) - c)
13 only_c = ((c - a) - b)
```

```

14
15 # dictionary of length of each set
16 count_dict = {
17     "a": len(a),
18     "b": len(b),
19     "c": len(c),
20     "a_and_b": len(a_and_b),
21     "b_and_c": len(b_and_c),
22     "a_and_c": len(a_and_c),
23     "a_and_b_and_c": len(a_and_b_and_c),
24     "a_or_b_or_c": len(a) + len(b) + len(c) - len(a_and_b) - len(b_and_c) -
len(a_and_c) + len(a_and_b_and_c),
25     "only_a_and_b_not_c": len(a_and_b) - len(a_and_b_and_c) ,
26     "only_a_and_c_not_b": len(a_and_c) - len(a_and_b_and_c) ,
27     "only_b_and_c_not_a": len(b_and_c) - len(a_and_b_and_c) ,
28     "only_a": len(a) - len(a_and_b) - len(a_and_c) + len(a_and_b_and_c) ,
29     "only_b": len(b) - len(a_and_b) - len(b_and_c) + len(a_and_b_and_c) ,
30     "only_c": len(c) - len(a_and_c) - len(b_and_c) + len(a_and_b_and_c)
31 }
32
33 # printing the length and probability of event represented by each set in
34 count_dict
35 for set_name, set_size in count_dict.items():
36     #print(f"\n{set_name} = {set_size}, prob({set_name}) = "
37     {set_size/len(a_or_b_or_c):.1f}")
38     print(f"\n{set_size},{set_size/len(a_or_b_or_c):.1f}")

```

Alternative Answer#3

```

1 # using custom functions defined in the suffix code block
2 # finding the expected set using custom set operations on sets a, b, and c
3 a_and_b = set_intersection(a, b)
4 b_and_c = set_intersection(b, c)
5 a_and_c = set_intersection(a, c)
6 a_and_b_and_c = set_intersection_generic(a, b, c)
7 a_or_b_or_c = set_union_generic(a, b, c)
8 only_a_and_b_not_c = set_difference(a_and_b, a_and_b_and_c)
9 only_a_and_c_not_b = set_difference(a_and_c, a_and_b_and_c)
10 only_b_and_c_not_a = set_difference(b_and_c, a_and_b_and_c)
11 only_a = set_difference(set_difference(a, b), c)
12 only_b = set_difference(set_difference(b, a), c)
13 only_c = set_difference(set_difference(c, a), b)
14
15 # dictionary of all sets
16 set_dict = {
17     "a": a,
18     "b": b,
19     "c": c,
20     "a_and_b": a_and_b,
21     "b_and_c": b_and_c,
22     "a_and_c": a_and_c,
23     "a_and_b_and_c": a_and_b_and_c,
24     "a_or_b_or_c": a_or_b_or_c,
25     "only_a_and_b_not_c": only_a_and_b_not_c ,
26     "only_a_and_c_not_b": only_a_and_c_not_b ,
27     "only_b_and_c_not_a": only_b_and_c_not_a,
28     "only_a": only_a,

```

```

29     "only_b": only_b,
30     "only_c": only_c
31 }
32
33 # printing the length and probability of event represented by each set in
34 # set_dict
35 for set_name, set_val in set_dict.items():
36     #print(f"\n{set_name} = {len(set_val)}, prob({set_name}) = "
37     #{len(set_val)/len(a_or_b_or_c):.1f}")
38     print(f"\n{len(set_val)},{len(set_val)/len(a_or_b_or_c):.1f}")

```

Alternative Answer#4

```

1 # Using custom functions defined in the suffix code block
2 # finding the expected set using custom set operations on sets a, b, and c
3 a_and_b = set_intersection(a, b)
4 b_and_c = set_intersection(b, c)
5 a_and_c = set_intersection(a, c)
6 a_and_b_and_c = set_intersection_generic(a, b, c)
7 a_or_b_or_c = set_union_generic(a, b, c)
8 only_a_and_b_not_c = set_difference(a_and_b, a_and_b_and_c)
9 only_a_and_c_not_b = set_difference(a_and_c, a_and_b_and_c)
10 only_b_and_c_not_a = set_difference(b_and_c, a_and_b_and_c)
11 only_a = set_difference(set_difference(a, b), c)
12 only_b = set_difference(set_difference(b, a), c)
13 only_c = set_difference(set_difference(c, a), b)
14
15 # dictionary of length of each set
16 count_dict = {
17     "a": len(a),
18     "b": len(b),
19     "c": len(c),
20     "a_and_b": len(a_and_b),
21     "b_and_c": len(b_and_c),
22     "a_and_c": len(a_and_c),
23     "a_and_b_and_c": len(a_and_b_and_c),
24     "a_or_b_or_c": len(a) + len(b) + len(c) - len(a_and_b) - len(b_and_c) -
len(a_and_c) + len(a_and_b_and_c),
25     "only_a_and_b_not_c": len(a_and_b) - len(a_and_b_and_c) ,
26     "only_a_and_c_not_b": len(a_and_c) - len(a_and_b_and_c) ,
27     "only_b_and_c_not_a": len(b_and_c) - len(a_and_b_and_c),
28     "only_a": len(a) - len(a_and_b) - len(a_and_c) + len(a_and_b_and_c),
29     "only_b": len(b) - len(a_and_b) - len(b_and_c) + len(a_and_b_and_c),
30     "only_c": len(c) - len(a_and_c) - len(b_and_c) + len(a_and_b_and_c)
31 }
32
33 # printing the length and probability of event represented by each set in
34 count_dict
35 for set_name, set_size in count_dict.items():
36     #print(f"\n{set_name} = {set_size}, prob({set_name}) = "
37     #{set_size/len(a_or_b_or_c):.1f}")
38     print(f"\n{set_size},{set_size/len(a_or_b_or_c):.1f}")

```


Week-6 Graded Assignment (Programming)

Week-6 Graded Assignment (Programming)

Problem 1

Question

Prefix (Shown)

Answer

Suffix (Hidden)

Test cases

 Public

 Private

Problem 2

Question

Prefix (Shown)

Answer

Suffix (Hidden)

Test cases

 Public

 Private

Tags

Problem 3

Question

Prefix (Shown)

Answer

Suffix (Hidden)

Test cases

 Public

 Private

Problem 1

Question

Write a function `freqWords` which accepts a list of words as a parameter and returns a dictionary which has the following structure:

- key: frequency of words in the list
- value: list of all words that have the above frequency

For example, consider this input:

```
1 | ['No', 'sentence', 'can', 'begin', 'with', 'because', 'because', 'because',  
  |   'is', 'a', 'conjunction.']}
```

The output dictionary corresponding to this is:

Key	Value
1	['no', 'sentence', 'can', 'begin', 'with', 'is', 'a', 'conjunction.']}
3	['because']

```
1 | def freqWords(words):  
2 |     '''  
3 |         Input: list of strings  
4 |         Output: dictionary  
5 |             key: integer  
6 |             value: list of strings  
7 |     '''
```

Note

- All keys of the returned dictionary should be in lowercase. This means that words like 'It' and 'it' in the input list are equivalent.
- Remove any occurrence of these characters from all the words present in the list `,;.:!.`. These characters correspond to: comma, semi-colon, colon, full-stop, exclamation mark.
- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Prefix (Shown)

```

1 malgudi = ['It', 'was', 'Monday', 'morning.', 'Swaminathan', 'was',
2   'reluctant', 'to', 'open', 'his',
3   'eyes.', 'He', 'considered', 'Monday', 'specially', 'unpleasant', 'in',
4   'the', 'calendar.', 'After',
5   'the', 'delicious', 'freedom', 'of', 'Saturday', 'And', 'Sunday,', 'it',
6   'was', 'difficult', 'to',
7   'get', 'into', 'the', 'Monday', 'mood', 'of', 'work', 'and', 'discipline.',
8   'He', 'shuddered', 'at',
9   'the', 'very', 'thought', 'of', 'school:', 'the', 'dismal', 'yellow',
10  'building;', 'the',
11  'fire-eyed', 'Vedanayagam,', 'his', 'class', 'teacher,', 'and',
12  'headmaster', 'with', 'his',
13  'thin', 'long', 'cane...']

```

Answer

```

1 def freqWords(wordList):
2     freq = {} # dictionary to store word as key and frequenct as value
3     for wd in wordList:
4         for i in ',;.:!': # iterating over the string contains the
5             characters to be removed
6             while i in wd: # do until no removeable left
7                 wd = wd.replace(i,'')
8             if wd.lower() not in freq.keys(): # initialization of new words as
9                 keys
10                freq[wd.lower()] = 0
11                freq[wd.lower()] += 1
12    freqWords_ = {} # dictionary to store frequency as key and word as value
13    for wd in freq.keys():
14        if freq[wd] not in freqWords_.keys(): # initialization for new
15            frequency as key with empty list
16            freqWords_[freq[wd]] = []
17            freqWords_[freq[wd]].append(wd) # dictionary inversion
18    return freqWords_

```

Suffix (Hidden)

```

1 dataset = {}
2 dataset['malgudi'] = malgudi
3 dataset['A Red, Red Rose'] = '''O my Luve's like a red, red rose
4 That's newly sprung in June;
5 O my Luve's like the melodie
6 That's sweetly play'd in tune.
7
8 As fair art thou, my bonnie lass,
9 So deep in luve am I:
10 And I will luve thee still, my dear,
11 Till a' the seas gang dry:
12
13 Till a' the seas gang dry, my dear,
14 And the rocks melt wi' the sun:
15 I will luve thee still, my dear,
16 While the sands o' life shall run.
17
18 And fare thee well, my only Luve

```

```

19 And fare thee well, a while!
20 And I will come again, my Luve,
21 Tho' it were ten thousand mile.'''.strip().split()
22
23 dataset['Song of Myself'] = '''I celebrate myself, and sing myself,
24 And what I assume you shall assume,
25 For every atom belonging to me as good belongs to you.
26
27 I loafe and invite my soul,
28 I lean and loafe at my ease observing a spear of summer grass.
29
30 My tongue, every atom of my blood, form'd from this soil, this air,
31 Born here of parents born here from parents the same, and their
32     parents the same,
33 I, now thirty-seven years old in perfect health begin,
34 Hoping to cease not till death.
35
36 Creeds and schools in abeyance,
37 Retiring back a while sufficed at what they are, but never forgotten,
38 I harbor for good or bad, I permit to speak at every hazard,
39 Nature without check with original energy.'''.strip().split()
40
41 fw = freqWords(dataset[input().strip()])
42 for i in sorted(list(fw.keys())):
43     print(i)
44     print(*sorted(fw[i]))

```

Test cases

Public

Input	Output
malgudi	1 after at building calendar cane class considered delicious difficult discipline dismal eyes fire-eyed freedom get headmaster in into long mood morning open reluctant saturday school shuddered specially sunday swaminathan teacher thin thought unpleasant vedanayagam very with work yellow 2 he it to 3 and his monday of was 6 the

Private

Input	Output
A Red, Red Rose	1 again am art as bonnie come deep fair it june lass life melody melt mile newly only o' play'd rocks rose run sands shall so sprung sun sweetly ten thou thousand tho' tune were wi' 2 a a' dry fare gang like luve's o red seas still that's till well while 3 dear in will 4 i thee 5 and luve 6 the 8 my
Song of Myself	1 abeyance air are as back bad begin belonging belongs blood but cease celebrate check creeds death ease energy forgotten form'd grass harbor hazard health hoping invite lean me nature never not now observing old or original perfect permit retiring schools shall sing soil soul speak spear sufficed summer their they thirty-seven till tongue while with without years 2 a assume atom born for from good here in loafe myself same the this what you 3 at every of parents 4 my to 6 and 7 i

Problem 2

Question

An institution decides to allow students to create student groups for each subject where students with similar marks can help each other. But it draws up a set of constraints for creating student groups:

- A group is associated with a particular `subject`.
- The group can consist of any number of students.
- The maximum difference of marks in this `subject` between the highest and lowest scorer within the group should be `markLimit`.

The `Scores Dataset` from the CT course will have the student details. Write a function called `crowdedGroup` that accepts three parameters as input:

- `scores`
- `subject`
- `markLimit`

The function should return a list of groups, where each group is a list of `SeqNo` of students of the largest possible group in that subject.

```
1 def crowdedGroup(scores, subject, markLimit):
2     ...
3     Input:
4         scores: list of dictionaries; Scores Dataset
5         subject: string
6         markLimit: integer
7     Output:
8         inner list: list of integers (SeqNo)
9         each inner list represents a group
10    ...
```

Note

- `scores` is a list of dictionaries that represents the `Scores Dataset`. Look at the prefix code shown below to get an idea of the structure of `scores`.
- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Prefix (Shown)

```
1 scores = [ { 'Chemistry': 78,
2             'CityTown': 'Erode',
3             'DateOfBirth': '7 Nov',
4             'Gender': 'M',
5             'Mathematics': 68,
6             'Name': 'Bhuvanesh',
7             'Physics': 64,
8             'SeqNo': 0,
9             'Total': 210},
```

```
10
11 { 'Chemistry': 91,
12   'CityTown': 'Salem',
13   'DateOfBirth': '3 Jun',
14   'Gender': 'M',
15   'Mathematics': 62,
16   'Name': 'Harish',
17   'Physics': 45,
18   'SeqNo': 1,
19   'Total': 198},
20
21 { 'Chemistry': 77,
22   'CityTown': 'Chennai',
23   'DateOfBirth': '4 Jan',
24   'Gender': 'M',
25   'Mathematics': 57,
26   'Name': 'Shashank',
27   'Physics': 54,
28   'SeqNo': 2,
29   'Total': 188},
30
31 { 'Chemistry': 78,
32   'CityTown': 'Chennai',
33   'DateOfBirth': '5 May',
34   'Gender': 'F',
35   'Mathematics': 42,
36   'Name': 'Rida',
37   'Physics': 53,
38   'SeqNo': 3,
39   'Total': 173},
40
41 { 'Chemistry': 89,
42   'CityTown': 'Madurai',
43   'DateOfBirth': '17 Nov',
44   'Gender': 'F',
45   'Mathematics': 87,
46   'Name': 'Ritika',
47   'Physics': 64,
48   'SeqNo': 4,
49   'Total': 240},
50
51 { 'Chemistry': 84,
52   'CityTown': 'Chennai',
53   'DateOfBirth': '8 Feb',
54   'Gender': 'F',
55   'Mathematics': 71,
56   'Name': 'Akshaya',
57   'Physics': 92,
58   'SeqNo': 5,
59   'Total': 247},
60
61 { 'Chemistry': 87,
62   'CityTown': 'Ambur',
63   'DateOfBirth': '23 Mar',
64   'Gender': 'M',
65   'Mathematics': 81,
66   'Name': 'Sameer',
67   'Physics': 82,
```

```
68     'SeqNo': 6,
69     'Total': 250},
70
71 { 'Chemistry': 76,
72   'CityTown': 'Vellore',
73   'DateOfBirth': '15 Mar',
74   'Gender': 'M',
75   'Mathematics': 84,
76   'Name': 'Aditya',
77   'Physics': 92,
78   'SeqNo': 7,
79   'Total': 252},
80
81 { 'Chemistry': 51,
82   'CityTown': 'Bengaluru',
83   'DateOfBirth': '28 Feb',
84   'Gender': 'M',
85   'Mathematics': 74,
86   'Name': 'Surya',
87   'Physics': 64,
88   'SeqNo': 8,
89   'Total': 189},
90
91 { 'Chemistry': 73,
92   'CityTown': 'Bengaluru',
93   'DateOfBirth': '6 Dec',
94   'Gender': 'M',
95   'Mathematics': 63,
96   'Name': 'Clarence',
97   'Physics': 88,
98   'SeqNo': 9,
99   'Total': 224},
100
101 { 'Chemistry': 68,
102   'CityTown': 'Chennai',
103   'DateOfBirth': '12 Jan',
104   'Gender': 'F',
105   'Mathematics': 64,
106   'Name': 'Kavya',
107   'Physics': 72,
108   'SeqNo': 10,
109   'Total': 204},
110
111 { 'Chemistry': 92,
112   'CityTown': 'Bengaluru',
113   'DateOfBirth': '30 Apr',
114   'Gender': 'M',
115   'Mathematics': 97,
116   'Name': 'Rahul',
117   'Physics': 92,
118   'SeqNo': 11,
119   'Total': 281},
120
121 { 'Chemistry': 71,
122   'CityTown': 'Chennai',
123   'DateOfBirth': '14 Jan',
124   'Gender': 'F',
125   'Mathematics': 52,
```

```
126     'Name': 'Srinidhi',
127     'Physics': 64,
128     'SeqNo': 12,
129     'Total': 187},
130
131 { 'Chemistry': 89,
132   'CityTown': 'Madurai',
133   'DateOfBirth': '6 May',
134   'Gender': 'M',
135   'Mathematics': 65,
136   'Name': 'Gopi',
137   'Physics': 73,
138   'SeqNo': 13,
139   'Total': 227},
140
141 { 'Chemistry': 93,
142   'CityTown': 'Trichy',
143   'DateOfBirth': '23 July',
144   'Gender': 'F',
145   'Mathematics': 89,
146   'Name': 'Sophia',
147   'Physics': 62,
148   'SeqNo': 14,
149   'Total': 244},
150
151 { 'Chemistry': 90,
152   'CityTown': 'Theni',
153   'DateOfBirth': '22 Sep',
154   'Gender': 'F',
155   'Mathematics': 76,
156   'Name': 'Goutami',
157   'Physics': 58,
158   'SeqNo': 15,
159   'Total': 224},
160
161 { 'Chemistry': 43,
162   'CityTown': 'Trichy',
163   'DateOfBirth': '30 Dec',
164   'Gender': 'M',
165   'Mathematics': 87,
166   'Name': 'Tauseef',
167   'Physics': 86,
168   'SeqNo': 16,
169   'Total': 216},
170
171 { 'Chemistry': 67,
172   'CityTown': 'Chennai',
173   'DateOfBirth': '14 Dec',
174   'Gender': 'M',
175   'Mathematics': 62,
176   'Name': 'Arshad',
177   'Physics': 81,
178   'SeqNo': 17,
179   'Total': 210},
180
181 { 'Chemistry': 97,
182   'CityTown': 'Erode',
183   'DateOfBirth': '9 Oct',
```

```
184     'Gender': 'F',
185     'Mathematics': 72,
186     'Name': 'Abirami',
187     'Physics': 92,
188     'SeqNo': 18,
189     'Total': 261},
190
191 { 'Chemistry': 62,
192   'CityTown': 'Trichy',
193   'DateOfBirth': '30 Aug',
194   'Gender': 'M',
195   'Mathematics': 56,
196   'Name': 'Vetrivel',
197   'Physics': 78,
198   'SeqNo': 19,
199   'Total': 196},
200
201 { 'Chemistry': 91,
202   'CityTown': 'Vellore',
203   'DateOfBirth': '17 Sep',
204   'Gender': 'M',
205   'Mathematics': 93,
206   'Name': 'Kalyan',
207   'Physics': 68,
208   'SeqNo': 20,
209   'Total': 252},
210
211 { 'Chemistry': 74,
212   'CityTown': 'Bengaluru',
213   'DateOfBirth': '15 Mar',
214   'Gender': 'F',
215   'Mathematics': 78,
216   'Name': 'Monika',
217   'Physics': 69,
218   'SeqNo': 21,
219   'Total': 221},
220
221 { 'Chemistry': 57,
222   'CityTown': 'Nagercoil',
223   'DateOfBirth': '17 Jul',
224   'Gender': 'F',
225   'Mathematics': 62,
226   'Name': 'Priya',
227   'Physics': 62,
228   'SeqNo': 22,
229   'Total': 181},
230
231 { 'Chemistry': 88,
232   'CityTown': 'Bengaluru',
233   'DateOfBirth': '13 May',
234   'Gender': 'F',
235   'Mathematics': 97,
236   'Name': 'Deepika',
237   'Physics': 91,
238   'SeqNo': 23,
239   'Total': 276},
240
241 { 'Chemistry': 58,
```

```
242     'CityTown': 'Madurai',
243     'DateOfBirth': '26 Dec',
244     'Gender': 'M',
245     'Mathematics': 44,
246     'Name': 'Siddharth',
247     'Physics': 72,
248     'SeqNo': 24,
249     'Total': 174},
250
251 { 'Chemistry': 92,
252   'CityTown': 'Chennai',
253   'DateOfBirth': '16 May',
254   'Gender': 'F',
255   'Mathematics': 87,
256   'Name': 'Geeta',
257   'Physics': 75,
258   'SeqNo': 25,
259   'Total': 254},
260
261 { 'Chemistry': 82,
262   'CityTown': 'Chennai',
263   'DateOfBirth': '22 Jul',
264   'Gender': 'M',
265   'Mathematics': 74,
266   'Name': 'JK',
267   'Physics': 71,
268   'SeqNo': 26,
269   'Total': 227},
270
271 { 'Chemistry': 52,
272   'CityTown': 'Madurai',
273   'DateOfBirth': '4 Mar',
274   'Gender': 'M',
275   'Mathematics': 81,
276   'Name': 'Jagan',
277   'Physics': 76,
278   'SeqNo': 27,
279   'Total': 209},
280
281 { 'Chemistry': 83,
282   'CityTown': 'Madurai',
283   'DateOfBirth': '10 Sep',
284   'Gender': 'F',
285   'Mathematics': 74,
286   'Name': 'Nisha',
287   'Physics': 83,
288   'SeqNo': 28,
289   'Total': 240},
290
291 { 'Chemistry': 81,
292   'CityTown': 'Vellore',
293   'DateOfBirth': '13 Oct',
294   'Gender': 'M',
295   'Mathematics': 72,
296   'Name': 'Naveen',
297   'Physics': 66,
298   'SeqNo': 29,
299   'Total': 219}]]
```

Answer

```

1 def crowdedGroup(scores, sub, markLimit):
2     subDict = {} # dictionary to store subject mark as key and list of
      students' ID as value
3     for sDict in scores:
4         if sDict[sub] not in subDict.keys():
5             subDict[sDict[sub]] = [] # Initialization for new keys
6             subDict[sDict[sub]].append(sDict['SeqNo']) # Append the student
      ID inside the list in key of subject marks
7         else:
8             subDict[sDict[sub]].append(sDict['SeqNo'])
9     maxNumber, crowdedGroup_ = 0, [] # variable to hold maximum group count
      and list of IDs of group
10    for i in range(0, 101-markLimit): # Iterating from zero to maximum mark
      - limit
11        group = []
12        for m in subDict.keys(): # Iterating through each marks
13            if i <= m <= i+markLimit:
14                group.extend(subDict[m]) # Append all the IDs if it is in
      the valid limit
15            if len(group) > maxNumber:
16                maxNumber = len(group) # Updating the maximum group count so far
17                crowdedGroup_ = [group] # Updating the maximum group list so far
18            if len(group) == maxNumber and group not in crowdedGroup_:
19                crowdedGroup_.append(group) # adding another group having same
      count
20    return crowdedGroup_

```

Suffix (Hidden)

```

1 sub, markLimit = input().strip().split()
2 markLimit = int(markLimit)
3 members = crowdedGroup(scores, sub, markLimit)
4 for m in members:
5     print(*sorted(m))

```

Test cases

Public

Input	Output
Physics 25	0 4 6 8 10 12 13 14 15 17 19 20 21 22 24 25 26 27 28 29 0 4 6 8 10 12 13 14 16 17 19 20 21 22 24 25 26 27 28 29

Private

Input	Output
Chemistry 30	0 1 2 3 4 5 6 7 9 10 11 12 13 14 15 17 18 20 21 23 25 26 28 29
Mathematics 30	0 1 2 4 5 6 7 8 9 10 13 15 16 17 18 21 22 25 26 27 28 29 0 1 4 5 6 7 8 9 10 13 14 15 16 17 18 21 22 25 26 27 28 29
Physics 32	0 4 5 6 7 8 9 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29

Tags

Problem 3

Question

The **Scores Dataset** from the CT course has the details of a class of students. A student **X** can mentor another student **Y** in **subject** if the following conditions are satisfied:

- **X** has scored more than **Y** in **subject**.
- The difference in marks between **X** and **Y** in **subject** lies in the range [10, 20], both end points included.

Write a function `topMentors` that accepts the following parameters as input:

- `scores`
- `subject`

The function should return a dictionary having the following structure as output:

- key: `SeqNo` of the student who can mentor the largest number of students in `subject`
- value: list of `SeqNo` of students who can be mentored by the above student

```
1 def topMentors(scores, subject):  
2     ...  
3     Input:  
4         scores: list of dictionaries; Scores dataset  
5         subject: string  
6     Output:  
7         dictionary:  
8             key: integer (SeqNo)  
9             value: list of integers (SeqNo)  
10            ...
```

Note

- `scores` is a list of dictionaries that represents the **Scores Dataset**. Look at the prefix code shown below to get an idea of the structure of `scores`.
- You don't need to accept the input from the user or print the output to the console. This will be processed internally.
- You only need to fill the details in the body of the function.

Prefix (Shown)

```
1 scores = [ { 'Chemistry': 78,  
2             'CityTown': 'Erode',  
3             'DateOfBirth': '7 Nov',  
4             'Gender': 'M',  
5             'Mathematics': 68,  
6             'Name': 'Bhuvanesh',  
7             'Physics': 64,  
8             'SeqNo': 0,  
9             'Total': 210},  
10            { 'Chemistry': 91,  
11              'CityTown': 'Salem',
```

```
13     'DateOfBirth': '3 Jun',
14     'Gender': 'M',
15     'Mathematics': 62,
16     'Name': 'Harish',
17     'Physics': 45,
18     'SeqNo': 1,
19     'Total': 198},
20
21 { 'Chemistry': 77,
22   'CityTown': 'Chennai',
23   'DateOfBirth': '4 Jan',
24   'Gender': 'M',
25   'Mathematics': 57,
26   'Name': 'Shashank',
27   'Physics': 54,
28   'SeqNo': 2,
29   'Total': 188},
30
31 { 'Chemistry': 78,
32   'CityTown': 'Chennai',
33   'DateOfBirth': '5 May',
34   'Gender': 'F',
35   'Mathematics': 42,
36   'Name': 'Rida',
37   'Physics': 53,
38   'SeqNo': 3,
39   'Total': 173},
40
41 { 'Chemistry': 89,
42   'CityTown': 'Madurai',
43   'DateOfBirth': '17 Nov',
44   'Gender': 'F',
45   'Mathematics': 87,
46   'Name': 'Ritika',
47   'Physics': 64,
48   'SeqNo': 4,
49   'Total': 240},
50
51 { 'Chemistry': 84,
52   'CityTown': 'Chennai',
53   'DateOfBirth': '8 Feb',
54   'Gender': 'F',
55   'Mathematics': 71,
56   'Name': 'Akshaya',
57   'Physics': 92,
58   'SeqNo': 5,
59   'Total': 247},
60
61 { 'Chemistry': 87,
62   'CityTown': 'Ambur',
63   'DateOfBirth': '23 Mar',
64   'Gender': 'M',
65   'Mathematics': 81,
66   'Name': 'Sameer',
67   'Physics': 82,
68   'SeqNo': 6,
69   'Total': 250},
70
```

```
71 { 'Chemistry': 76,
72   'CityTown': 'Vellore',
73   'DateOfBirth': '15 Mar',
74   'Gender': 'M',
75   'Mathematics': 84,
76   'Name': 'Aditya',
77   'Physics': 92,
78   'SeqNo': 7,
79   'Total': 252},
80
81 { 'Chemistry': 51,
82   'CityTown': 'Bengaluru',
83   'DateOfBirth': '28 Feb',
84   'Gender': 'M',
85   'Mathematics': 74,
86   'Name': 'Surya',
87   'Physics': 64,
88   'SeqNo': 8,
89   'Total': 189},
90
91 { 'Chemistry': 73,
92   'CityTown': 'Bengaluru',
93   'DateOfBirth': '6 Dec',
94   'Gender': 'M',
95   'Mathematics': 63,
96   'Name': 'Clarence',
97   'Physics': 88,
98   'SeqNo': 9,
99   'Total': 224},
100
101 { 'Chemistry': 68,
102   'CityTown': 'Chennai',
103   'DateOfBirth': '12 Jan',
104   'Gender': 'F',
105   'Mathematics': 64,
106   'Name': 'Kavya',
107   'Physics': 72,
108   'SeqNo': 10,
109   'Total': 204},
110
111 { 'Chemistry': 92,
112   'CityTown': 'Bengaluru',
113   'DateOfBirth': '30 Apr',
114   'Gender': 'M',
115   'Mathematics': 97,
116   'Name': 'Rahul',
117   'Physics': 92,
118   'SeqNo': 11,
119   'Total': 281},
120
121 { 'Chemistry': 71,
122   'CityTown': 'Chennai',
123   'DateOfBirth': '14 Jan',
124   'Gender': 'F',
125   'Mathematics': 52,
126   'Name': 'Srinidhi',
127   'Physics': 64,
128   'SeqNo': 12,
```

```
129     'Total': 187},
130
131     { 'Chemistry': 89,
132      'CityTown': 'Madurai',
133      'DateOfBirth': '6 May',
134      'Gender': 'M',
135      'Mathematics': 65,
136      'Name': 'Gopi',
137      'Physics': 73,
138      'SeqNo': 13,
139      'Total': 227},
140
141     { 'Chemistry': 93,
142      'CityTown': 'Trichy',
143      'DateOfBirth': '23 July',
144      'Gender': 'F',
145      'Mathematics': 89,
146      'Name': 'Sophia',
147      'Physics': 62,
148      'SeqNo': 14,
149      'Total': 244},
150
151     { 'Chemistry': 90,
152      'CityTown': 'Theni',
153      'DateOfBirth': '22 Sep',
154      'Gender': 'F',
155      'Mathematics': 76,
156      'Name': 'Goutami',
157      'Physics': 58,
158      'SeqNo': 15,
159      'Total': 224},
160
161     { 'Chemistry': 43,
162      'CityTown': 'Trichy',
163      'DateOfBirth': '30 Dec',
164      'Gender': 'M',
165      'Mathematics': 87,
166      'Name': 'Tauseef',
167      'Physics': 86,
168      'SeqNo': 16,
169      'Total': 216},
170
171     { 'Chemistry': 67,
172      'CityTown': 'Chennai',
173      'DateOfBirth': '14 Dec',
174      'Gender': 'M',
175      'Mathematics': 62,
176      'Name': 'Arshad',
177      'Physics': 81,
178      'SeqNo': 17,
179      'Total': 210},
180
181     { 'Chemistry': 97,
182      'CityTown': 'Erode',
183      'DateOfBirth': '9 Oct',
184      'Gender': 'F',
185      'Mathematics': 72,
186      'Name': 'Abirami',
```

```
187     'Physics': 92,
188     'SeqNo': 18,
189     'Total': 261},
190
191 { 'Chemistry': 62,
192   'CityTown': 'Trichy',
193   'DateOfBirth': '30 Aug',
194   'Gender': 'M',
195   'Mathematics': 56,
196   'Name': 'Vetrivel',
197   'Physics': 78,
198   'SeqNo': 19,
199   'Total': 196},
200
201 { 'Chemistry': 91,
202   'CityTown': 'Vellore',
203   'DateOfBirth': '17 Sep',
204   'Gender': 'M',
205   'Mathematics': 93,
206   'Name': 'Kalyan',
207   'Physics': 68,
208   'SeqNo': 20,
209   'Total': 252},
210
211 { 'Chemistry': 74,
212   'CityTown': 'Bengaluru',
213   'DateOfBirth': '15 Mar',
214   'Gender': 'F',
215   'Mathematics': 78,
216   'Name': 'Monika',
217   'Physics': 69,
218   'SeqNo': 21,
219   'Total': 221},
220
221 { 'Chemistry': 57,
222   'CityTown': 'Nagercoil',
223   'DateOfBirth': '17 Jul',
224   'Gender': 'F',
225   'Mathematics': 62,
226   'Name': 'Priya',
227   'Physics': 62,
228   'SeqNo': 22,
229   'Total': 181},
230
231 { 'Chemistry': 88,
232   'CityTown': 'Bengaluru',
233   'DateOfBirth': '13 May',
234   'Gender': 'F',
235   'Mathematics': 97,
236   'Name': 'Deepika',
237   'Physics': 91,
238   'SeqNo': 23,
239   'Total': 276},
240
241 { 'Chemistry': 58,
242   'CityTown': 'Madurai',
243   'DateOfBirth': '26 Dec',
244   'Gender': 'M',
```

```
245     'Mathematics': 44,
246     'Name': 'Siddharth',
247     'Physics': 72,
248     'SeqNo': 24,
249     'Total': 174},
250
251 { 'Chemistry': 92,
252   'CityTown': 'Chennai',
253   'DateOfBirth': '16 May',
254   'Gender': 'F',
255   'Mathematics': 87,
256   'Name': 'Geeta',
257   'Physics': 75,
258   'SeqNo': 25,
259   'Total': 254},
260
261 { 'Chemistry': 82,
262   'CityTown': 'Chennai',
263   'DateOfBirth': '22 Jul',
264   'Gender': 'M',
265   'Mathematics': 74,
266   'Name': 'JK',
267   'Physics': 71,
268   'SeqNo': 26,
269   'Total': 227},
270
271 { 'Chemistry': 52,
272   'CityTown': 'Madurai',
273   'DateOfBirth': '4 Mar',
274   'Gender': 'M',
275   'Mathematics': 81,
276   'Name': 'Jagan',
277   'Physics': 76,
278   'SeqNo': 27,
279   'Total': 209},
280
281 { 'Chemistry': 83,
282   'CityTown': 'Madurai',
283   'DateOfBirth': '10 Sep',
284   'Gender': 'F',
285   'Mathematics': 74,
286   'Name': 'Nisha',
287   'Physics': 83,
288   'SeqNo': 28,
289   'Total': 240},
290
291 { 'Chemistry': 81,
292   'CityTown': 'Vellore',
293   'DateOfBirth': '13 Oct',
294   'Gender': 'M',
295   'Mathematics': 72,
296   'Name': 'Naveen',
297   'Physics': 66,
298   'SeqNo': 29,
299   'Total': 219}]
```

Answer

```
1 def adjMat(scores, sub): # Function to find adjacency matrix
2     adj = {} # adjacency matrix as nested dictionary
3     for mentor in scores: # Outer dictionary / Row for mentor
4         adj[mentor['SeqNo']] = {} # initializing with an empty dictionary
5     for all SeqNo
6         for mentored in scores: # Inner dictionary / column for mentored
7             # mentoring condition and prevention of same student comparison
8             if 10 <= mentor[sub] - mentored[sub] <= 20 and mentor['SeqNo']
9                 != mentored['SeqNo']:
10                 adj[mentor['SeqNo']][mentored['SeqNo']] = 1
11     return adj
12
13 def topMentors(scores, sub):
14     adj = adjMat(scores, sub) # getting an adjacency matrix
15     topMentors_ = {} # dictionary to store mentor as key and mentored SeqNo
16     list as value
17     maxMentored = 0 # value to hold the maximum mentored students
18     for i in adj.keys():
19         if len(adj[i]) > maxMentored:
20             maxMentored = len(adj[i]) # replacing the maximum mentored count
21             topMentors_ = {} # reinitialization with empty dictionary
22             topMentors_[i] = list(adj[i].keys()) # adding entry with mentor
23             as key and mentored list value
24             if len(adj[i]) == maxMentored:
25                 topMentors_[i] = list(adj[i].keys()) # adding entry with mentor
26             as key and mentored list value
27     return topMentors_
```

Suffix (Hidden)

```
1 tm = topMentors(scores, input())
2 for i in sorted(list(tm.keys())):
3     print(i)
4     print(*sorted(tm[i]))
```

Test cases

Public

Input	Output
Physics	6 0 4 8 10 12 14 20 21 22 24 26 29

Private

Input	Output
Chemistry	<p>14</p> <p>0 2 3 7 9 21 26 28 29</p>
Mathematics	<p>7</p> <p>0 5 8 10 13 18 26 28 29</p>

Week - 7, Graded, Theory

Week - 7, Graded, Theory

Problem-1

[Question-1](#)

[Answer](#)

[Solution](#)

[Question-2](#)

[Answer](#)

[Solution](#)

[Question-3](#)

[Answer](#)

[Solution](#)

[Question-4](#)

[Answer](#)

[Solution](#)

Problem-2

[Question-5](#)

[Answer](#)

[Solution](#)

[Question-6](#)

[Answer](#)

[Solution](#)

[Question-7](#)

[Answer](#)

[Solution](#)

[Question-8](#)

[Answer](#)

[Solution](#)

[Question-9](#)

[Answer](#)

[Solution](#)

[Question-10](#)

[Answer](#)

[Solution](#)

Problem 3

[Question-11](#)

[Answer](#)

[Solution](#)

[Question-12](#)

[Answer](#)

[Solution](#)

[Question-13](#)

[Answer](#)

[Solution](#)

Problem-4

[Question-14](#)

[Answer](#)

[Solution](#)

[Question-15](#)

[Answer](#)

[Solution](#)

[Question-16](#)

[Answer](#)

[Solution](#)

[Question-17](#)

[Answer](#)

[Solution](#)

Problem-1

Questions 1 to 4 are based on common theme

Use the data block given below to answer questions 1 to 4.

```
1 org_role_list = [("dev", "developer"),
2                   ("ba", "business analyst"),
3                   ("qa", "quality analyst"),
4                   ("lead", "program manager"),
5                   ("ra", "research associate")]
6 org_work = {0: "code",
7             1: "requirement",
8             2: "validation",
9             3: "management",
10            4: "research" }
11 org_sal_band = {"sal-band-1": 1000000.0,
12                  "sal-band-2": 1500000.0,
13                  "sal-band-3": 2000000.0}
```

Question-1

Which of the following code fragments will create a dictionary `org_role_dict` from the list `org_role_list`. [MSQ]

The output dictionary should be equivalent to `org_role_list` and is given below.

Expected Output

```
1 | {
2 |     'dev': 'developer',
3 |     'ba': 'business analyst',
4 |     'qa': 'quality analyst',
5 |     'lead': 'program manager',
6 |     'ra': 'research associate'
7 | }
```

(a)

```
1 | org_role_dict = {}
2 | for i in range(len(org_role_list)):
3 |     org_role_dict.update({org_role_list[i][0] : org_role_list[i][1]})
4 | print(org_role_dict)
```

(b)

```
1 | org_role_dict = {}
2 | for i in range(len(org_role_list)):
3 |     org_role_dict.update([org_role_list[i]])
4 | print(org_role_dict)
```

(c)

```
1 | org_role_dict = {}
2 | for i in range(len(org_role_list)):
3 |     org_role_dict.update([(org_role_list[i][0], org_role_list[i][1])])
4 | print(org_role_dict)
```

(d)

```
1 | org_role_dict = {}
2 | for unit, role in org_role_list:
3 |     org_role_dict[unit] = role
4 | print(org_role_dict)
```

Answer

(a), (b), (c), (d)

Solution

All the above options are correct. The function `update()` is used to add/update elements in a dictionary. Option (a) and (b) adds a `dict` of `key, value`. Option (b) adds a list of `key, value` tuple. Option (c) is another way to add/update am item in the dictionary.

Question-2

The list `emp_work` represents the work area of each employee on `emp_name` (given below). An employee at `i-th` position in `emp_name` is assigned the `(i%5)-th` item value of `org_work`. If the employee is at 4th position in `emp_name`, it is assigned $4 \% 5 = 4$ which means "research" work area. Select the piece of code that gives expected `emp_work`. [MSQ]

```
1 | emp_name = ["Manohar", "Rajeev", "Savita", "Daizy", "Alok"]
```

Expected Output

```
1 | ['code', 'requirement', 'validation', 'management', 'research']
```

(a)

```
1 | emp_work = []
2 | for i in range(len(emp_name)):
3 |     emp_work.append(org_work[i%5])
4 | print(emp_work)
```

(b)

```
1 | emp_work = []
2 | for i in range(len(emp_name)):
3 |     emp_work = emp_work + [org_work[i%5]]
4 | print(emp_work)
```

(c)

```
1 | emp_work = []
2 | for i in range(len(emp_name)):
3 |     emp_work.append(org_work[i%5])
4 | print(emp_work)
```

(d)

```
1 | emp_work = []
2 | for i in range(len(emp_name)):
3 |     emp_work + [org_work[i%5]]
4 | print(emp_work)
```

Answer

(a), (b)

Solution

Option (a) and (b) are correct. The function `append()` is used to add a new element to a list in the end. This function updates the original list itself and does not require return statement. Another way to concatenate two list is to use `+` operator. Option (c) is incorrect, it missed out the `len()` function inside the `range()` to get the length of the list. Option (d) requires `emp_work` to be

assigned the value of concatenate list at Line-3 to work. Hence, it is incorrect.

Question-3

Employee details and their salary bands are given below in the same order. The list `emp_sal` is holding annual salaries corresponding to the employees on the `emp_name`. Which of the following code fragment populates `emp_sal` correctly. [MSQ]

```
1 | emp_name = ["Manohar", "Rajeev", "Savita", "Daizy", "Alok"]
2 | emp_band = ["sal-band-1", "sal-band-2", "sal-band-1", "sal-band-3", "sal-
band-2"]
```

(a)

```
1 | emp_sal = []
2 | for emp_id in range(len(emp_band)):
3 |     emp_sal.append(org_sal_band[emp_band[emp_id]])
4 | print(emp_sal)
```

(b)

```
1 | emp_sal = []
2 | for emp_id in range(len(emp_band)):
3 |     emp_sal.append(org_sal_band.get(emp_band[emp_id]))
4 | print(emp_sal)
```

(c)

```
1 | emp_sal = []
2 | for emp in emp_name:
3 |     emp_sal.append(org_sal_band[emp_band[emp]])
4 | print(emp_sal)
```

(d)

```
1 | emp_sal = []
2 | for emp_id in range(len(emp_name)):
3 |     emp_sal.append(org_sal_band[emp_band[emp_id]])
4 | print(emp_sal)
```

Answer

(a), (b), (d)

Solution

An element at a an index from a list can be retrieved using `list_var[index]` or `list_var.get(index)`. Option (a), (b) and (d) are correct. In option (c), the loop variable is a string representing element of `emp_name` rather than an `int`, it is not a valid list index. Therefore, (c) is incorrect.

Question-4

The salary of each employee in `emp_sal` (created in the last question) is hiked by say x percent due to good company performance. Which of the following code can be used to print the hiked salary of employee in the `emp_list`. The output should be employee name, old salary and new salary separated by comma. Each employee details is printed on a new line. [MSQ]

Example:

Old Salary	Hike %	New Salary
10000	10	$10000 + 1000 = 11000$

(a)

```
1 x = int(input())
2 for emp_id in range(len(emp_name)):
3     print(emp_name[emp_id], emp_sal[emp_id], emp_sal[emp_id]*x, sep=",")
```

(b)

```
1 x = int(input())
2 for emp_id in range(len(emp_name)):
3     print(emp_name[emp_id], emp_sal[emp_id], emp_sal[emp_id]*(1+0.01*x),
      sep=",")
```

(c)

```
1 x = int(input())
2 for emp_id in range(len(emp_name)):
3     print(emp_name[emp_id], emp_sal[emp_id], emp_sal[emp_id]*(1+x), sep=",")
```

(d)

```
1 x = int(input())
2 for emp_id in range(len(emp_name)):
3     print(emp_name[emp_id], emp_sal[emp_id], emp_sal[emp_id]*1.01*x, sep=",")
```

Answer

(b)

Solution

After hike, employee salary is $\text{New Salary} = \text{old salary} + \text{old salary} * x\% = \text{old salary} * (1+0.01*x)$.

Option (b) performs the correct calculation of `New salary`.

Problem-2

Questions 5 to 10 are based on a common theme

Use below code fragments to answer questions.

```
1 a = int(input())
2 b = int(input())
3 def v(a):
4     return(a + w(b, a))
5
6 def w(a, b = 1):
7     a += b
8     return(a)
9
10 def x(a = 1, b = 1):
11     a += b
12     return(a)
13
14 def y(a = 1):
15     global b
16     a += b
17     b += 1
18     return(a)
19
20 def z(a):
21     if a:
22         a += z(a-1)
23     return a
```

Question-5

Which of the following functions group uses default arguments partially or completely? [MCQ]

- (a) v, w, x
- (b) v, w, z
- (c) w, x, y
- (d) x, y, z

Answer

- (c) w, x, y

Solution

By definition, function w, x, y uses default arguments. The **default** value is assigned by using the assignment (=) operator of the form keyword_name = value inside the parenthesis.

Question-6

Which of the following functions uses recursion? [MCQ]

(a) v

(b) w

(c) x

(d) y

(e) z

Answer

(e) z

Solution

A function is called `recursive` when it calls itself inside the function definition. Option (c) is the correct answer.

Question-7

What value function `z` returns when the argument is a positive integer `n`? [MCQ]

- (a) `(n + 1)`
- (b) `n(n + 1)/2`
- (c) `2n`
- (d) `n`

Answer

- (b) `n(n + 1)/2`

Solution

For a positive integer argument, the function `z` recursively itself with a new argument which is one less than the value of last argument. $z(n) \rightarrow n+z(n) \rightarrow n+(n-1)+z(n-2) \rightarrow n+(n-1)+(n-2)+\dots+2+1+z(0)=n*(n+1)/2$, where `z(0) = 0`. This is a sum of a series of the first `n` natural numbers.

Question-8

What should be the expected output of the below code block, when `a = b = 10`. [MCQ]

```
1 | print(v(a), a, b)
2 | print(w(a), a, b)
3 | print(x(a), a, b)
```

(a)

```
1 | 30 10 10
2 | 11 10 10
3 | 11 10 10
```

(b)

```
1 | 11 10 10
2 | 11 10 10
3 | 11 10 10
```

(c)

```
1 | 30 20 10
2 | 30 20 10
3 | 30 20 10
```

(d)

```
1 | 10 10 10
2 | 10 10 10
3 | 10 10 10
```

Answer

(a)

Solution

Both function `w` and `x` accepts two parameter `a` and `b` and returns sum of it. In this case for `a=10, b=10`, the output is `10`. Both uses `b` as default argument in the function call, which is 1. The function `x`, adds global variable to `a` to the output of `w(b, a)` which is 30. Hence, (a) is the correct answer.

Question-9

Given `a = 10, b = 20`, what should be the output of `print(y(a), a, b)`? [MCQ]

- (a) `31 10 21`
- (b) `30 11 21`
- (c) `30 10 20`
- (d) `30 10 21`

Answer

- (d)

Solution

The function `y` accepts two parameters `a` and `b` and returns sum of it. Inside the function value of global variable `b` is also incremented by 1. For `a=10, b=20`, the output of `print(y(a), a, b)` will be `30 10 21`.

Question-10

Given `a = 10, b = 20`, what is the output of the following snippet of code?

```
1 | y(a)
2 | print(z(a), a, b)
```

- (a) 50 10 21
- (b) 50 10 21
- (c) 51 11 21
- (d) 55 10 21

Answer

(d)

Solution

The function `y` accepts two parameters `a` and `b` and returns sum of it. Inside the function value of global variable `b` is also incremented by 1. Function `z`, accepts a positive integer `n` as a parameter and returns sum of the first `n` natural numbers.

For `a=10, b=20`, the output of `print(y(a), a, b)` will be 55 10 21.

Problem 3

Questions 11 to 13 are based on common theme

```
1 def doSomething(M, r, c):
2     if len(M)*len(M[0]) == r * c:
3         l = []
4         for i in M:
5             for j in i:
6                 l.append(j)
7         M_ = []
8         for i in range(r):
9             M_.append([])
10            for j in range(c):
11                M_[-1].append(l.pop(0))
12
13 return M_
```

Question-11

`dosomething` always returns a list for any given parameters.

- (a) True
- (b) False

Answer

- (b) False

Solution

The return statement is defined inside the if-statement. Hence `dosomething` will return a list only when the `if` condition is satisfied.

Question-12

What will `dosomething(M, r, c)` return? `M` is a non-empty matrix (nested list), `r` and `c` are integers.

- (a) Transpose of matrix `M`
- (b) Reshaped matrix `M` of `r` rows and `c` columns for any integer `r` and `c`.
- (c) Reshaped matrix `M` of `r` rows and `c` columns for specific `r` and `c`.
- (d) Reshaped matrix `M` of `c` rows and `r` columns for specific `r` and `c`.

Answer

(c)

Solution

The function will return a list only if the condition `len(M)*len(M[0]) == r * c` is satisfied else `None` will be returned, hence it will return a list only when the product of `r` and `c` is equal to the product of number of rows `len(M)` and columns `len(M[0])`.

From the below code-snippet all the elements are made into a single list `l` with row wise ordered.

```
1 | def dosomething(M, r, c):  
2 |     if len(M)*len(M[0]) == r * c:  
3 |         l = []  
4 |         for i in M:  
5 |             for j in i:  
6 |                 l.append(j)
```

A new nested `M_` is created populated with the elements of list where the number of rows and columns will be `r` and `c` respectively.

```
1 |     M_ = []  
2 |     for i in range(r):  
3 |         M_.append([])  
4 |         for j in range(c):  
5 |             M_[ -1 ].append(l.pop(0))  
6 |     return M_
```

Question-13

For what values of `r` and `c` will `dosomething(M, r, c)` return a matrix? `len(M)` is equal to 12 and `len(M[0])` is 14. [MSQ]

- (a) 24, 7
- (b) 6, 28
- (c) 2, 82
- (d) 3, 56

Answer

(a), (b), (d)

Solution

The code results in the reshaping of the matrix. The number of elements before and after the operations should be equal. `len(M)` and `len(M[0])` gives number of rows and number of columns of the original matrix respectively. The total number of elements in the original matrix is `12*14=168`. Therefore, Option (a), (b), (d) are valid answers.

Problem-4

Questions 14 to 17 are based on common theme

Question-14

In the following code block, what should be the value of `a, b, c, d, e, f, g` to produce text given in the output.

```
1 x = "Running contests are of mostly three type {p}, {q} and {r}.".format(p =
2     a, q = b, r = c)
3 y = f"Marathon an {a} sport, have two popular format {d} and {e}"
4 z = "The fastest human footspeed on record is %.2f km/h seen during a %d
5     meter sprint by %s."%(g, h, f)
6 print(x, y, z, sep="\n")
```

Expected Output

- 1 Running contests are of mostly three type Endurance, Sprinting and Hurdling.
- 2 Marathon is an Endurance sport, there are two popular format Half Marathon and Full Marathon
- 3 The fastest human footspeed on record is 44.72 km/h seen during a 100 meter sprint by Usain Bolt.

Select the option that correctly matches items on Variable column to Value column and print above output. [MCQ]

Variable	Value
1. a	A. "Hurdling"
2. b	B. "Half Marathon"
3. c	C. 44.72
4. d	D. "Full Marathon"
5. e	E. 100
6. f	F. "Usain Bolt"
7. g	G. "Endurance"
8. h	H. "Sprinting"

- (a) 1-G, 3-H, 2-A, 4-B, 6-D, 5-F, 7-C, 8-E
(b) 2-G, 1-H, 4-A, 3-B, 5-D, 6-F, 7-C, 8-E
(c) 3-G, 2-H, 1-A, 5-B, 4-D, 7-F, 6-C, 8-E
(d) 1-G, 2-H, 3-A, 4-B, 5-D, 6-F, 7-C, 8-E

Answer

(d) 1-G, 2-H, 3-A, 4-B, 5-D, 6-F, 7-C, 8-E

Solution

Below table shows the correct match of the options.

Variable	Value
1. a	G. "Endurance"
2. b	H. "Sprinting"
3. c	A. "Hurdling"
4. d	B. "Half Marathon"
5. e	D. "Full Marathon"
6. f	F. "Usain Bolt"
7. g	C. 44.72
8. h	E. 100

Question-15

Which of the following code will create a list `s` from three string variables `x`, `y`, `z` used in previous question. Hint: These variables represent a line in the below text in the same order.
[MSQ]

- 1 Running contests are of mostly three type Endurance, Sprinting and Hurdling.
- 2 Marathon is an Endurance sport, there are two popular format Half Marathon and Full Marathon
- 3 The fastest human footspeed on record is 44.72 km/h seen during a 100 meter sprint by Usain Bolt.

(a)

```
1 s = [0]*3
2 s[0] = x
3 s[1] = y
4 s[2] = z
```

(b)

```
1 s = [0]
2 s[0] = x
3 s[1] = y
4 s[2] = z
```

(c)

```
1 s = [x, y, z]
```

(d)

```
1 s = [x] + [y] + [z]
```

Answer

(a), (c), (d)

Solution

A list can be concatenated using multiple ways. Options (a), (c), (d) are valid ways. Option (b) is not correct, because `s` is a list of only one element but we are trying to access 2nd and 3rd element. This should result in an error.

Question-16

The variable `s` used in the previous question holds a list of strings and is given below.

```
1 s = [
2 'Running contests are of mostly three type Endurance, Sprinting and
Hurdling.',
3 'Marathon an Endurance sport, have two popular format Half Marathon and Full
Marathon',
4 'The fastest human footspeed on record is 44.72 km/h seen during a 100 meter
sprint by Usain Bolt.'
5 ]
```

Which of the following options create a new nested list `t` given below (by splitting each element of `s` into a list of strings by space)? [MSQ]

```
1 [
2 ['Running', 'contests', 'are', 'of', 'mostly', 'three', 'type', 'Endurance',
'Sprinting', 'and', 'Hurdling.'],
3 ['Marathon', 'an', 'Endurance', 'sport', 'have', 'two', 'popular', 'format',
'Half', 'Marathon', 'and', 'Full', 'Marathon'],
4 ['The', 'fastest', 'human', 'footspeed', 'on', 'record', 'is', '44.72',
'km/h', 'seen', 'during', 'a', '100', 'meter', 'sprint', 'by', 'Usain',
'Bolt.']
5 ]
```

(a)

```
1 t = []
2 for i in [0,1,2]:
3     t.append(s[i].split())
```

(b)

```
1 t = []
2 for i in range(3):
3     t.append(s[i].split())
```

(c)

```
1 t = []
2 for i in range(len(s)):
3     t.append(s[i].split(" ")) # a single space enclosed between double quotes
```

(d)

```
1 t = []
2 for q in s:
3     t.append(q.split(" ")) # a single space enclosed between double quotes
```

Answer

(a), (b), (c), (d)

Solution

All options are valid. The string method `split()` breaks up a string at the specified separator and returns a list of string. By default the separator is a blank space string `' '`.

Question-17

The variable `t` used in the previous question holds a nested list of strings and is given below.

```
1 [  
2 ['Running', 'contests', 'are', 'of', 'mostly', 'three', 'type', 'Endurance',  
  'Sprinting', 'and', 'Hurdling.'],  
3 ['Marathon', 'an', 'Endurance', 'sport', 'have', 'two', 'popular', 'format',  
  'Half', 'Marathon', 'and', 'Full', 'Marathon'],  
4 ['The', 'fastest', 'human', 'footspeed', 'on', 'record', 'is', '44.72',  
  'km/h', 'seen', 'during', 'a', '100', 'meter', 'sprint', 'by', 'Usain',  
  'Bolt.'][  
5 ]
```

```
1 print(t[0][a:b])  
2 print(t[1][:c])  
3 print(t[2][::-c])
```

The output of the code fragment is given below. [MCQ]

```
1 ['Running', 'contests']  
2 ['Marathon', 'have', 'Half', 'Marathon']  
3 ['Bolt.', 'meter', 'seen', 'record', 'fastest']
```

What should be the value of `a`, `b`, `c` in order to get this output?

- (a) `a = 0, b = 2, c = 4`
- (b) `a = 1, b = 3, c = 4`
- (c) `a = 0, b = 2, c = 5`
- (d) `a = 0, b = 3, c = 4`

Answer

- (a) `a = 0, b = 2, c = 4`

Solution

The first line is a slice (a sub list) of the first element of given nested list. The sub list contains only first two elements , hence `a=0, b=2` is acceptable.

The second line and third line of the output is printed when `c` is 4.

Hence, option (a) is the correct answer.

Week-8 Practice Programming

Week-8 Practice Programming

Problem 1

 Question

 Answer

 Testcases

 Public

 Private

 Tags

 Comments

Problem 2

 Question

 Answer

 Suffix (Hidden)

 Testcases

 Public

 Private

 Tags

 Comments

Problem 3

 Question

 Answer

 Suffix Visible

 Testcases

 Public

 Private

 Tags

 Comments

Problem 4

 Question

 Answer

 Suffix Hidden

 Testcases

 Public

 Private

 Tags

 Comments

Problem 5

 Question

 Answer

 Suffix Hidden

 Testcases

 Public

 Private

 Tags

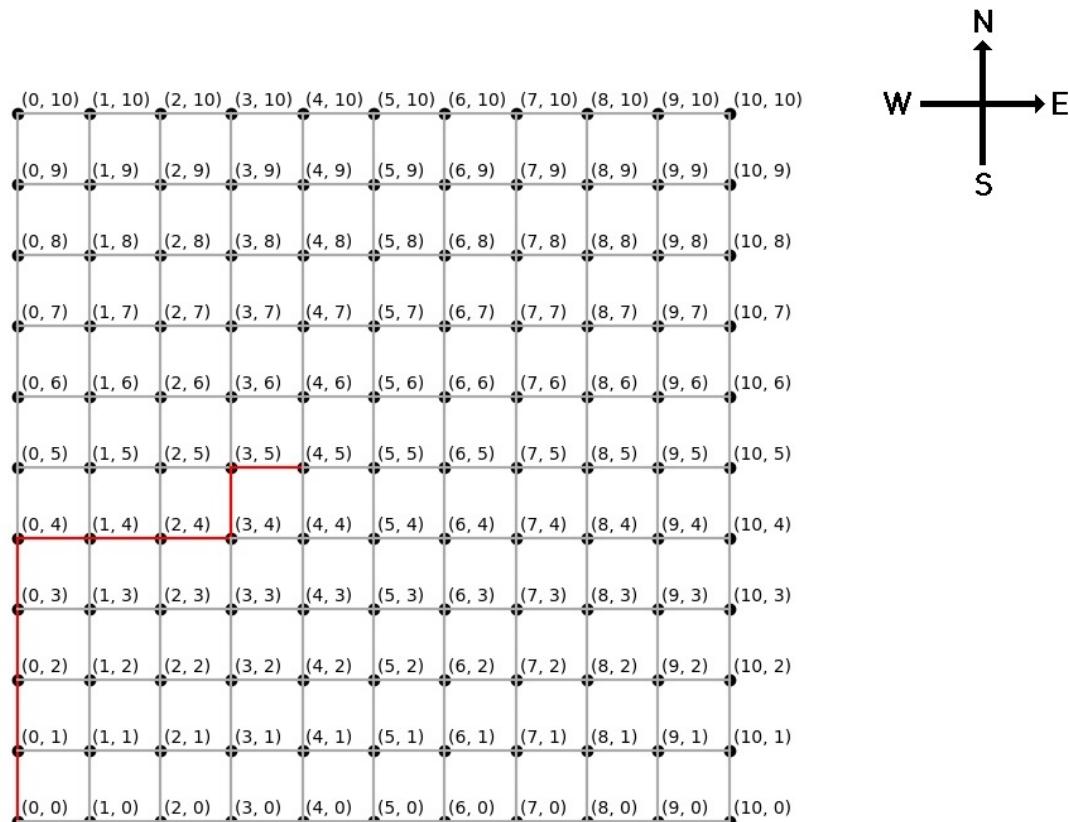
 Comments

Problem 1

The below image represents a grid having 11×11 nodes numbered from 0 to 10.

- Distance between one node to next connected node is 1 unit.
- One can go in any direction, each letter counts as 1 unit in respective direction.
 - N North
 - S South
 - E East
 - W West

The below graph shows the path for `NNNNEEENE` starting from $(0, 0)$.



Question

Write a python program to take a string as input from user and print the total distance traveled.

Answer

```
1 | p = input()
2 | print(p.count('N') + p.count('E') + p.count('S') + p.count('W')) # count
   every step and print it
```

Testcases

Public

Input	Output
NNNNEEENE	9
NEWS	4

Private

Input	Output
NEEWWSSEWEWWS	15
EEEEEEEEEEEE	13
N	1

Tags

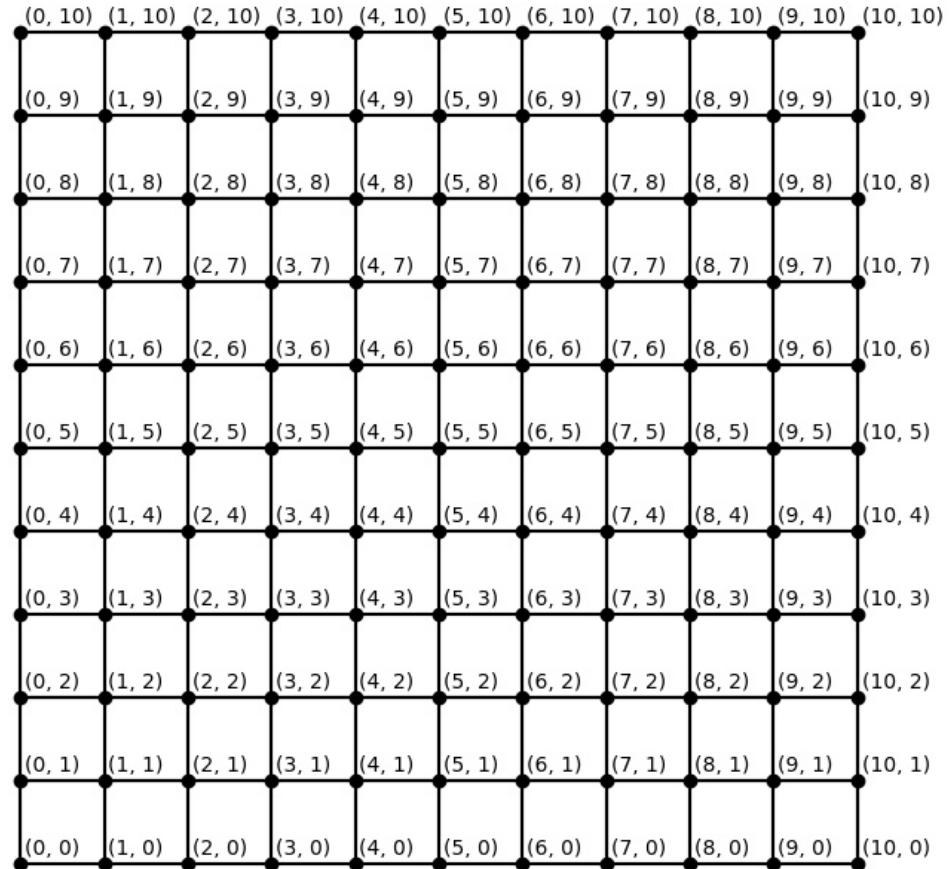
TAGS SEPERATED BY COMMAS

Comments

COMMENTS IN TEXT/ CODE

Problem 2

The below image represents a grid having 11×11 nodes numbered from 0 to 10.



Question

A block is a closed loop of distinct adjacent connected nodes.

(0, 0), (1, 0), (1, 1) and (0, 1) will form a block.

(0, 0), (1, 0), (1, 1) and (1, 2) will not form a block.

Write a function `isBlock` to take a list of four nodes as tuples and check whether they form block or not. Return True if they form a block and False otherwise.

- The function is only required.
- No need to take any input or print any output.

Answer

```
1 def isBlock(l):
2     l.sort() # sorted in order of Bottom-Left, Top-Left Bottom-Right, and
3     # Top-Right node
4     b = 0
5     if (l[0][0], l[0][1]+1) == l[1]: # Checking of Bottom-Left and Top-
6     # Left
7         b += 1
8     if (l[0][0]+1, l[0][1] ) == l[2]: # Checking of Bottom-Left and Bottom-
9     # Right
10    b += 1
11    if (l[0][0]+1, l[0][1]+1) == l[3]: # Checking of Bottom-Left and Top-
12    # Right
13    b += 1
14 return b == 3
```

Suffix (Hidden)

```
1 # suffix
2 import ast
3 def parse(inp):
4     inp = ast.literal_eval(inp)
5     return inp
6
7 fncall = input()
8 lparen = fncall.find("(")
9 rparen = fncall.rfind(")")
10 fname = fncall[:lparen]
11 farg = fncall[lparen+1:rparen]
12
13 if fname == "isBlock":
14     arg = parse(farg)
15     print(isBlock(arg))
16 else:
17     print("Function", fname, "unknown")
```

Testcases

Public

Input	Output
isBlock([(0, 0), (1, 0), (1, 1), (0, 1)])	True
isBlock([(0, 0), (1, 0), (1, 1), (1, 4)])	False

Private

Input	Output
<code>isBlock([(0, 0), (1, 0), (1, 1), (1, 0)])</code>	<code>False</code>
<code>isBlock([(2, 1), (3, 1), (3, 2), (2, 2)])</code>	<code>True</code>
<code>isBlock([(5, 5), (6, 6), (5, 6), (6, 5)])</code>	<code>True</code>
<code>isBlock([(1, 1), (1, 1), (1, 1), (1, 1)])</code>	<code>False</code>

Tags

TAGS SEPERATED BY COMMAS

Comments

COMMENTS IN TEXT/ CODE

Problem 3

Question

Write a Python program for a ticket reservation system. The operational details are given below.

- Available tickets are 100
- Booking time starts at 10:00 and closes at 17:00 ends inclusive.
- Any booking that falls outside the booking time should be rejected.
- One person can book tickets for multiple persons, hence that reservation should be completely reserved or completely rejected.
- Assume that multiple bookings do not happen at the same time.
- `HH:MM S 10 C 25 W 20 O 5` is an example line from the input.
 - `'S'`, `'C'`, `'W'` and `'O'` are the identifiers that denote Senior citizen, Child, Woman and Others respectively.
 - The line always starts with `HH:MM` which are the time in 24 hours.
 - `S 10` denotes the 10 tickets of Senior citizens, similarly for other identifiers.
 - The ticket type and number can be in any order, where the number is always followed by the identifier. The same example is valid and is equivalent to `HH:MM C 25 S 10 O 5 W 20 .`
- Update the dictionary `log` accordingly. Refer the suffix part of the code.
- You don't need to print anything.
- The last line of input will be an empty line.

Answer

```
1 availableTickets = 100 # variable to store the available tickets
2 log = {} # initialization of log
3 for i in ['S', 'C', 'W', 'O']: # creating a key of all type of tickets
4     log[i] = 0
5 line = ' '
6 while line:
7     line = input().strip() # read input
8     if availableTickets > 0: # continue if there is atleast one ticket
9         t = line[:5] # time
10        if '10:00' <= t <= '17:00': # checking for the time limit
11            tk = line[5:].strip().split() # ticket type and numbers in list
12            d = {} # dictionary for storing ticket type and numbers of a line
13            for i in range(0, len(tk), 2): # storing the ticket type and
number
14                d[tk[i].strip()] = int(tk[i+1])
15            tkCount = 0 # variable to store total tickets in the line
16            for k in d:
17                tkCount += d[k] # total tickets in the line
18                if tkCount <= availableTickets: # is enough tickets available
19                    availableTickets -= tkCount # reduce the number of tickets
from available tickets
20                    for k in d: # insert in to log
21                        log[k] += d[k]
```

Suffix Visible

```
1 | sold = 0
2 | for i in log:
3 |     sold += log[i]
4 | print(f'Tickets sold: {sold}')
5 | print(f'Tickets remaining: {100-sold}')
6 | print(f'Senior citizens: {log["S"]}')
7 | print(f'Children: {log["C"]}')
8 | print(f'Women: {log["W"]}')
9 | print(f'Other: {log["O"]}')
```

Testcases

Public

Input	Output
09:15 S 1 C 1 O 1 10:00 S 12 O 21 12:00 W 19 13:01 O 12 14:15 O 15 W 12 C 1 ⋮	Tickets sold: 92 Tickets remaining: 8 Senior citizens: 12 Children: 1 Women: 31 Other: 48
09:15 S 1 C 1 O 1 10:00 S 12 O 21 12:00 W 19 13:01 O 12 14:15 O 15 W 12 C 1 14:16 O 25 16:17 O 5 ⋮	Tickets sold: 97 Tickets remaining: 3 Senior citizens: 12 Children: 1 Women: 31 Other: 53

Private

Input	Output
12:00 W 19 13:01 O 12 14:15 O 15 W 12 C 1 14:16 O 25 16:17 O 5 	Tickets sold: 89 Tickets remaining: 11 Senior citizens: 0 Children: 1 Women: 31 Other: 57
09:15 S 1 C 1 O 1 10:00 S 12 O 21 12:00 W 19 13:01 O 12 14:15 O 15 W 12 C 1 14:16 O 25 16:17 O 5 17:00 W 3 	Tickets sold: 100 Tickets remaining: 0 Senior citizens: 12 Children: 1 Women: 34 Other: 53
09:15 S 1 C 1 O 1 10:00 S 12 O 21 12:00 W 19 13:01 O 12 14:15 O 15 W 12 C 1 14:16 O 25 16:17 O 5 17:00 W 2 17:01 O 1 	Tickets sold: 99 Tickets remaining: 1 Senior citizens: 12 Children: 1 Women: 33 Other: 53

Tags

TAGS SEPERATED BY COMMAS

Comments

COMMENTS IN TEXT/ CODE

Problem 4

Question

Write a function `findMe`, it returns `True` if a `*` is present anywhere in the nested list and `False` otherwise.

Answer

```
1 def findMe(l):
2     found = False # variable to store whether the '*' is found
3     for i in l: # iterating through each list elements of passed list into the
4         function
5             if i == '*': # if '*' is found return True
6                 return True
7             if type(i) == type([]): # if the element type is list then search for
8                 '*' in that list before moving on to the next element
9                 found = found or findMe(i) # found will become True if atleast one
10                '*' is present inside the list
11
12 return found
```

Suffix Hidden

```
1 # Suffix
2 import ast
3
4 def parse(inp):
5     inp = ast.literal_eval(inp)
6     return inp
7
8 fncall = input()
9 lparen = fncall.find("(")
10 rparen = fncall.rfind(")")
11 fname = fncall[:lparen]
12 farg = fncall[lparen+1:rparen]
13
14 if fname == "findMe":
15     arg = parse(farg)
16     print(findMe(arg))
17 else:
18     print("Function", fname, "unknown")
```

Testcases

Public

Input	Output
<code>findMe([['a', 'z'], ['b'], ['c'], [['d']], [[5], [*], [], []], [], ['j']]])</code>	True
<code>findMe([['a', 'z'], ['b'], ['c'], [['d']], [[5], [], [], []], [], ['j']]])</code>	False
<code>findMe([['a', 'z'], ['b'], ['c'], [['d']], [[5], [], [], []], [], ['j']], [*])</code>	True

Private

Input	Output
<code>findMe([[1,2,3],[1,2,*]])</code>	True
<code>findMe([*])</code>	True
<code>findMe([[],[],[],[],[],[],[],[],[[[[[[[]]]]]]]])</code>	False
<code>findMe([[],[],[],[],[],[],[],[],[[[[[[[*]]]]]]])</code>	True
<code>findMe([[1,2,3],[1,2]])</code>	False
<code>findMe([[1,2,3],,[[[[[[]]]]], [1,2]])</code>	False

Tags

TAGS SEPERATED BY COMMAS

Comments

COMMENTS IN TEXT/ CODE

Problem 5

Question

Write a function `listToDict` to convert a nested list (two level) into dictionary where the keys of the dictionary be the index of the nested list.

- Let `l` be the passed nested list and `d` be the returned dictionary, where `l[i][j] == d[i][j]` should return True for all valid index / key `i` and `j`.

Write a `dictToList` to convert a dictionary (two level) into nested list where the index of the list be the keys of the dictionary.

- Let `d` be the passed dictionary and `l` be the returned nested list, where `d[i][j] == l[i][j]` should return True for all valid index / key `i` and `j`.
- The keys of the dictionaries are always an integer from 0 to 100.
- 0 is placed in the position where there is no key.
- The size of the list should be in a way that accommodates all the keys in the dictionary as list index.

Answer

```
1 def listToDict(l):
2     d = {} # initializing the dictionary to be returned
3     for i in range(len(l)):
4         d[i] = {} # inner dictionary initialization
5         for j in range(len(l[i])):
6             d[i][j] = l[i][j] # storing the list element of index [i][j] to
7             # dictionary of key [i][j]
8     return d
9
10
11 def dictToList(d):
12     l = [] # initializing the list to be returned
13     maxKeys1 = max(d.keys()) # length of outer list/row is the maximum of the
14     # key in outer dictionary
15     maxKeys2 = 0 # length of inner list/column is the maximum of the key in
16     # inner dictionary
17     for i in d:
18         for j in d[i]:
19             if j > maxKeys2: #finding the maximum of the key in inner
20             # dictionary
21                 maxKeys2 = j
22
23     for i in range(maxKeys1+1): # iteraing for number of rows required
24         l.append([]) # appending an empty list for each column
25         for j in range(maxKeys2+1): # iteraing for number of rows required
26             if i in d.keys() and j in d[i].keys(): # append to list element
27                 [i][j] if i and j are in the keys of outer and inner dictionary respectively
28                 l[-1].append(d[i][j])
29             else: # otherwise zero is appended
30                 l[-1].append(0)
31
32     return l
```

Suffix Hidden

```
1 import ast
2
3 def parse(inp):
4     inp = ast.literal_eval(inp)
5     return inp
6
7 fncall = input()
8 lparen = fncall.find("(")
9 rparen = fncall.rfind(")")
10 fname = fncall[:lparen]
11 farg = fncall[lparen+1:rparen]
12
13 if fname == "listToDict":
14     arg = parse(farg)
15     print(listToDict(arg))
16 elif fname == "dictToList":
17     arg = parse(farg)
18     print(dictToList(arg))
19 else:
20     print("Function", fname, "unknown")
```

Testcases

Public

Input	Output
<code>listToDict([[1], [1,2]])</code>	<code>{0: {0: 1}, 1: {0: 1, 1: 2}}</code>
<code>dictToList({1:{5:1, 2:9}, 3:{0:1}})</code>	<code>[[0, 0, 0, 0, 0, 0], [0, 0, 9, 0, 0, 1], [0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0]]</code>

Private

Input	Output
<code>listToDict([[1,2,2,2,2,2,2], [1,2,2]])</code>	<code>{0: {0: 1, 1: 2, 2: 2, 3: 2, 4: 2, 5: 2, 6: 2}, 1: {0: 1, 1: 2, 2: 2}}</code>
<code>listToDict([[1,2,3]])</code>	<code>{0: {0: 1, 1: 2, 2: 3}}</code>
<code>listToDict([[1,2],[3,4]])</code>	<code>{0: {0: 1, 1: 2}, 1: {0: 3, 1: 4}}</code>
<code>dictToList({10:{1:0},1:[3:12]})</code>	<code>[[0, 0, 0, 0], [0, 0, 0, 12], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]</code>
<code>dictToList({10:{0:0}})</code>	<code>[[0], [0], [0], [0], [0], [0], [0], [0], [0], [0]]</code>
<code>dictToList({0:{0:0}})</code>	<code>[[0]]</code>

Tags

Comments

Week-7, Graded, Programming

Week-7, Graded, Programming

Problem-1

 Question

 Test Cases

 Answer

Problem-2

 Question

 Test Cases

 Answer

Problem-3

 Question

 Answer

 Test Cases

Problem 4

 Question

 Answer

 Test Cases

Problem 5

 Question

 Answer

 Suffix Code Block (hidden)

 Test Cases

Problem-1

Question

Accept an integer as input and print the digits (as lower case words) present in it from left to right. Each digit should be printed on a separate line. At the end, print all the digits (as words) in a single line without space in camel case.

Test Cases

Type	Input	Output
public	123	one two three oneTwoThree
public	5440	five four four zero fiveFourFourZero
private	10101	one zero one zero one oneZeroOneZeroOne
private	123456789	one two three four five six seven eight nine oneTwoThreeFourFiveSixSevenEightNine
private	66666	six six six six six sixSixSixSixSix

Answer

```
1 num = input()  
2 # use a dictionary to store the mapping between digits and words  
3 D = {'0': 'zero', '1': 'one', '2': 'two', '3': 'three', '4': 'four',  
4     '5': 'five', '6': 'six', '7': 'seven', '8': 'eight', '9': 'nine'}  
5 # treat num as a string  
6 # loop through each digit and print the corresponding word  
7 # key is digit, value is corresponding word
```

```
8  for digit in num:
9      print(D[digit])
10
11 # go through (index, digit) pair in the number
12 # here, index is the position of the digit in num
13 # starting from 0 to len(num) - 1
14 # you can look up the enumerate function online
15 for index, digit in enumerate(num):
16     if index != 0:
17         # capitalize the word corresponding to the first digit
18         word = D[digit].capitalize()
19     else:
20         # every other word remains in lower case
21         word = D[digit]
22     # if it is not the last word, add a comma at the end
23     if index != len(num) - 1:
24         print(word, end = '')
25     #
26     else:
27         print(word)
```

Problem-2

Question

Accept details of a class of students as input from the user and create a dictionary named `students` to store the details.

Input Format

You have to accept input from the user in this problem.

- The first line of input will be a string that denotes the test case number. Store it in a string variable called `TEST_CASE`. This will be used by us for processing your solution.
- The second line of input will have the number of students in the class. Call this n .
- The third line of input will be a sequence of $s + 1$ comma-separated words. The first word will always be `id`. The remaining s words will be names of the subjects taken up by the class.
- The next n lines will have $s + 1$ comma-separated numbers on each line. The first number will be the student id. The remaining s numbers on this line will have the marks scored by this student in the corresponding subjects.

Sample Input

```
1 | TEST CASE 1
2 |
3 | 2
4 | id,sub1,sub2,sub3
5 | 1,85,65,90
6 | 2,100,56,34
```

Evaluation Format

You are not expected to print anything to the console. You only have to create a dictionary named `students` that is structured as follows.

- Keys will be student ids. It has to be an integer.
- Value for each key will be another dictionary which is structured as follows:
 - Keys will be subject names and should be strings.
 - Values will be the marks scored by the student in that subject. All values will be integers.

Sample Output

`students` is the dictionary's name:

```
1 | {
2 |   1: {
3 |     'sub1': 85,
4 |     'sub2': 65,
5 |     'sub3': 90
6 |   }
7 |   2: {
8 |     'sub1': 100,
9 |     'sub2': 56,
10 |    'sub3': 34
11 |   }
12 | }
```

Note

- You have to accept input from the user in this problem.
- You do not have to print the output to the console.
- You will be tested based on the dictionary `students` that you create. Do NOT use a different name for the dictionary. If `students` matches the exact format given in the problem specification, you will get the actual output as `CORRECT`.
- Do NOT forget to store the test case string in the variable `TEST_CASE`.
- `TEST_CASE` and `students` should have global scope so that we can call them from anywhere from within the code.

Test Cases

Type	Input	Output
Public	TEST CASE 1 2 id,sub1,sub2,sub3 1,85,65,90 2,100,56,34	CORRECT
Public	TEST CASE 2 5 id,maths,physics,chemistry,biology,compsci 1,10,20,30,40,50 2,60,70,80,90,100 3,100,98,38,49,29 4,100,100,100,100,100 5,78,39,98,29,98	CORRECT
Private	TEST CASE 3 1 id,s1,s2 1,78,45	CORRECT
Private	TEST CASE 4 10 id,Maths 1,10 2,20 3,30 4,40 5,50 6,60 7,70 8,80 9,90 10,100	CORRECT

Answer

```
1 # accept test case
2 TEST_CASE = input()
3 # accept the number of students as input
4 n = int(input())
5 # names of subjects to be stored as a list
6 subjects = [ ]
7 # get the subject names
8 for word in input().split(','):
9     if word == 'id':
10         continue    # ignore the id word
```

```
11     subjects.append(word)
12 # student details to be stored in students dict
13 students = dict()
14 for i in range(n):
15     # input sequence
16     inp_seq = input().split(',')
17     # first entry is student id
18     sid = int(inp_seq[0])
19     # create a key for this student
20     students[sid] = dict()
21     # go through all marks
22     for i in range(len(inp_seq[1:])):
23         # this is the subject name
24         subject = subjects[i]
25         # get the mark for this subject
26         mark = int(inp_seq[i + 1])
27         # for student sid, and for subject, add the value mark
28         students[sid][subject] = mark
```

Problem-3

Question

Write a function named `merge` that accepts two dictionaries as input, merges them into a single dictionary, and returns merged dictionary as output. This is what we mean by merging:

- The key-value pairs in both dictionaries should be present in the returned dictionary.
- If a particular key is common to both dictionaries, the value corresponding to this key in one of the two dictionaries should be retained.

Input Format

- The only input for this problem corresponds to the test case number. You do not have to accept this from the user. We will take care of this.
- The input dictionaries will not be shown to you. They will be passed as arguments to your functions. We will take care of this.

Evaluation Format

You have to write the function `merge`

- The function `merge` accepts three arguments.
 - `D1` : first dictionary
 - `D2` : second dictionary
 - `priority` : This is a string variable that denotes the priority given to common keys while merging. That is, if both `D1` and `D2` have a key in common, then this variable will determine which value needs to be retained. More specifically, `priority` can take one of these two values:
 - `first` : retain the value corresponding to the common key present in the first dictionary
 - `second` : retain the value corresponding to the common key present in the second dictionary

```
1 def merge(D1, D2, priority):
2     ...
3     Arguments:
4         - D1: first dictionary
5         - D2: second dictionary
6         - priority: string
7     Returns: D; merged dictionary
8     ...
```

- You do not need to print output to the console.
- The dictionary returned by your function will be evaluated against our solution. If it matches, then the expected output is `CORRECT`.

Note

- You do NOT have to call the function `merge`. The function call will be our responsibility.

Answer

```
1 def merge(D1, D2, priority):
2     # basic idea is to write code only for priority being first
3     # when priority is second, we can just have a recursive call
4     # with priority being first
5     if priority == 'second':
6         return merge(D2, D1, 'first')
7     # merged dict will be stored in D
8     D = dict()
9     # first add all key-value pairs in D1 into D
10    for key, value in D1.items():
11        D[key] = value
12    for key, value in D2.items():
13        # if this is not a common key, only then add it to D
14        if key not in D:
15            D[key] = value
16    return D
```

Test Cases

```
1 data = dict()
2 data['TEST CASE 1'] = {
3     'D1': {1: 2, 2: 3, 3: 4},
4     'D2': {1: 10, 4: 15, 5:10},
5     'priority': 'first'
6 }
7 data['TEST CASE 2'] = {
8     'D1': {1: 2, 2: 3, 3: 4},
9     'D2': {1: 10, 4: 15, 5:10},
10    'priority': 'second'
11 }
12 data['TEST CASE 3'] = {
13     'D1': {'a': 1, 'b': 100, 'c': 1000},
14     'D2': {'d': 20, 'e': 1000, 'c': 1234},
15     'priority': 'first'
16 }
17 data['TEST CASE 4'] = {
18     'D1': {'a': 1, 'b': 100, 'c': 1000},
19     'D2': {'d': 20, 'e': 1000, 'c': 1234},
20     'priority': 'second'
21 }
22 data['TEST CASE 5'] = {
23     'D1': {({0, 1}: [1, 2, 3], (1, 2): [4, 5, 6], (3, 4): [1000])},
24     'D2': {({0, 1}: [5, 6, 7], (1, 2): [10, 11, 12], (5, 6): [1234])},
25     'priority': 'first'
26 }
27 data['TEST CASE 6'] = {
28     'D1': {({0, 1}: [1, 2, 3], (1, 2): [4, 5, 6], (3, 4): [1000])},
29     'D2': {({0, 1}: [5, 6, 7], (1, 2): [10, 11, 12], (5, 6): [1234])},
30     'priority': 'second'
31 }
```

Problem 4

Question

Suppose you have an $m \times n$ matrix, where m represents number of rows and n represents number of columns and $2 < m, n < 12$, that consists of integer values. Write a program that creates a new $m \times n$ matrix in which all the elements are replaced by zeros except the border elements.

Input Format

- The first line contains a positive integer m to represent the number of rows.
- The second line contains a positive integer n represent to number of columns.
- Accept m rows in which each row contain n elements separated by space.

Evaluation Format

Print the matrix elements row wise and separated by space after replacing all elements by zeros except the border elements.

Sample Input

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

Sample Output

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

Answer

```
1 # This function accepts data row-wise from the user and returns two-
2 # dimension matrix as a list
3 def create_matrix(r,c):
4     l1 = []
5     i = 0;
6     while i < r:
7         l2 = input().split(' ')
8         l1.append(l2)
9         i += 1
10    return l1
11 # This function replaces all elements by zero except the border elements
12 def replacebyzero(l,r,c):
13     for i in range(r):
14         for j in range(c):
15             # Below conditional expression target to non-border elements of
the matrix
16             if i!=0 and j!=0 and i!=r-1 and j!=c-1:
```

```

16         l[i][j]=0;
17
18 # Accept the number of row and column from user
19 m = int(input())
20 n = int(input())
21 # Call create_matrix() function to accept data row-wise from user and return
22 # two-dimension matrix as a list
22 l = create_matrix(m,n)
23 # Call replacebyzero() function that replace all element by zero except the
24 # border elements
24 replacebyzero(l,m,n)
25
26 # Print the matrix elements row-wise and separated them by space
27 for i in range(m):
28     for j in range(n):
29         if j != n-1:
30             print(l[i][j],end = ' ')
31         else:
32             print(l[i][j])

```

Test Cases

Public

Input-1

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

Output-1

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

Input-2

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

Output-2

1	2 3 4
2	2 0 4
3	4 0 7
4	8 7 5

Private

Input-1

```
1 5
2 5
3 1 2 3 4 5
4 -1 -2 -3 -4 -5
5 7 8 9 4 5
6 2 1 3 4 5
7 1 4 7 8 9
```

Output-1

```
1 1 2 3 4 5
2 -1 0 0 0 -5
3 7 0 0 0 5
4 2 0 0 0 5
5 1 4 7 8 9
```

Input-2

```
1 6
2 4
3 0 0 0 0
4 0 1 2 0
5 0 2 5 0
6 0 1 5 0
7 0 2 5 0
8 0 0 0 0
```

Output-2

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

Input-3

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

Output-3

1	1	2	3	4
2	3	0	0	5
3	4	0	0	6
4	2	3	4	5

Problem 5

Question

The `Collatz` function is defined for a positive integer n as follows.

- $f(n) = 3n + 1$ if n is odd
- $f(n) = \frac{n}{2}$ if n is even

We consider the repeated application of the `Collatz` function starting with a given integer n , as follows:

$f(n), f(f(n)), f(f(f(n))), \dots$

It is conjectured that no matter which positive integer n you start from, the sequence will always reach 1. [Source: Wikipedia]

For example, If $n = 10$, the sequence is:

Seq No.	x	f(x)
1	10	5
2	5	16
3	16	8
4	8	4
5	4	2
6	2	1

Thus if you start from $n = 10$, you need to apply the function f six times in order to first reach 1.

Write a function `collatz_repeat(n)`, where $1 < n \leq 32,000$, and return the number of times f has to be applied repeatedly in order to first reach 1.

Input Format

- The input to the program will be an integer in the range $1 < n \leq 32000$. You do NOT have to accept input from the user.

Evaluation Format

Complete the following function:

```
1 | def collatz_repeat(n):
2 |     ...
3 |     Argument: n -> integer
4 |     Returns: integer
5 |     ...
```

- You do NOT have to print the output to the console. This will be done by us internally.
- You do not have to call the function. It is sufficient if you complete the body of the function.

Answer

Recursive Solution

```
1 def collatz_repeat(n):
2     # When n becomes 1, 0 will be added in the return value and this is the
3     # terminate condition for collatz_repeat(n)
4     if n == 1:
5         return 0
6     else:
7         # For each recursive call of collatz_repeat(n), 1 will be added in
8         # return value
9         if n % 2==0:
10            return 1 + collatz_repeat(n//2)
11        else:
12            return 1 + collatz_repeat(3*n+1)
```

Non-Recursive Solution

```
1 def collatz_repeat(n):
2     # Initialize counter count to 0
3     count=0
4     while n != 1:
5         if n % 2 == 0:
6             # if n is even then n//2 will be assign to n
7             n = n // 2
8             # Increase the count value by 1
9             count += 1
10        else:
11            # if n is odd then 3 * n + 1 will be assign to n
12            n = 3 * n + 1
13            # Increase the count value by 1
14            count += 1
15    # Return count after completion of the while loop
16    return count
```

Suffix Code Block (hidden)

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

Test Cases

Public Test Cases

Input	Output
101	25
100	25
2463	208
7	16

Private Test Cases

Input	Output
30000	178
31999	98
125	108
999	49

Week-8, Practice Theory

Week-8, Practice Theory

Problem 1

Question-1

Answer

Solution

Problem 2

Question-2

Answer

Solution

Question-3

Answer

Solution

Problem 3

Question-4

Answers

Solution

Problem 4

Question-5

Answers

Solution

Problem 5

Question-6

Answer

Solution

Problem 6

Question-7

Answer

Solution

Problem 7

Question-8

Answer

Solution

Problem 1

Question-1

`l` is a non-empty sorted list, `first` is the starting index and `last` is the end index of `l`. Which of these statements are correct regarding the function `search()`. It is MSQ type question.

```
1 def search(l, first, last, k):
2     mid = int((first + last)/2)
3     if k > l[mid]:
4         search(l, mid+1, last, k)
5     elif k < l[mid]:
6         search(l, first, mid-1, k)
7     elif k == l[mid]:
8         print("Number found at", mid, 'index')
9     else:
10        print("Number does not exist in the list")
```

- (a) `Number does not exist in the list`, is printed if `k` lies between minimum and maximum element of the list, but does not exist in the list `l`
- (b) `RecursionError` occurs if `k` does not exist in the list `l`
- (c) `IndexError` occurs if `k` does not exist in the list `l`
- (d) line 10 never gets executed
- (e) `Number does not exist in the list`, is printed if `k` is less than the minimum or greater than the maximum element of the list `l`.

Answer

(b) and (d)

Solution

Option (b) is correct because If element `k` does not exist in the list then for each recursive call, `mid` value will be calculated and after some recursive call it becomes constant (`0 <= mid <= len(l)-1`). By default python recursion limit is 1000, so after 1000 recursion, `RecursionError` will occur.

Option (d) is correct because control flow of execution will never reach the `else` statement block.

Problem 2

Common data for questions 2 and 3

`l1` and `l2` are two non-empty lists each of which has distinct integer elements in sorted order. Function `merge_sorted_list()` returns a list `l3` which has elements in non-descending order after merging the elements of `l1` and `l2`.

```
1 def merge_sorted_list(l1,l2):
2     l3 = []
3     i,j = 0,0
4     while(i < len(l1) and j < len(l2)):
5         if l1[len(l1)-i-1] <= l2[j]:
6             l3.append(l1[len(l1)-i-1])
7             i += 1
8         else:
9             l3.append(l2[j])
10            j += 1
11     while(i < len(l1)):
12         l3.append(l1[len(l1)-i-1])
13         i += 1
14     while(j < len(l2)):
15         l3.append(l2[j])
16         j += 1
17 return l3
```

Question-2

Which statement is correct about `l1` and `l2` to always produce correct output by the function `merge_sorted_list()`?

- (a) `l1` and `l2` should be in descending order.
- (b) `l1` and `l2` should be in ascending order.
- (c) `l1` should be in ascending order and `l2` should be in descending order.
- (d) `l1` should be in descending order and `l2` should be in ascending order.

Answer

(d)

Solution

In Code line 5 `if l1[len(l1)-i-1] <= l2[j]:` traverse the list `l1` from the last index and `l2` from the start index for comparison so `l1` should be in descending order and `l2` should be in ascending order for merging.

Question-3

The function `merge_sorted_list1()` given below (after removing line 11 to 16 from the above function `merge_sorted_list()`).

```

1 def merge_sorted_list1(l1,l2):
2     l3 = []
3     i,j = 0,0
4     while(i < len(l1) and j < len(l2)):
5         if l1[len(l1)-i-1] <= l2[j]:
6             l3.append(l1[len(l1)-i-1])
7             i += 1
8         else:
9             l3.append(l2[j])
10            j += 1
11    return l3

```

Assume x_1 is the maximum value of the list $l1$ and x_2 is the maximum value of $l2$. Which of these statements are correct about the `merge_sorted_list1()`. It is MSQ type question.

- (a) If x_2 is larger than x_1 , then elements of $l2$ that are greater than or equal to x_1 will not be included in $l3$.
- (b) If x_1 is equal to x_2 then both will not be included in the returned list $l3$.
- (c) If x_1 is larger than x_2 , then elements of $l1$ that are greater than x_2 will not be included in $l3$
- (d) If x_1 is not equal to x_2 , then elements of any one list ($l1$ or $l2$) that are less than the largest element of the other list will not be included in $l3$
- (e) If x_2 is larger than x_1 then only elements of list $l2$ that are greater than x_1 will not be included in $l3$

Answer

(a) and (c)

Solution

Due to `if l1[len(l1)-i-1] <= l2[j]:` condition in code, for each element of $l1$ that is less than or equal to the comparing element of $l2$ will be copied in $l3$ first. because of this option (a) and (c) are correct. we can easily observe this by taking some examples.

For option (a) example :

```

1 l1 = [9,8,7,6]
2 l2 = [3,4,6,8,9,10,11]
3 Here x_2(11) > x_1(9)
4 function return [3, 4, 6, 6, 7, 8, 8, 9], you can see 9,10,11 of l2 are not
   included.

```

For option (c) example :

```

1 l1 = [9,8,7,6,5,4]
2 l2 = [3,4,6]
3 Here x_1(9) > x_2(6)
4 function return [3, 4, 4, 5, 6, 6], you can see 9,8,7 of l1 are not included.

```

Problem 3

Question-4

```
1 def fun1(x):
2     if x == []:
3         return 0
4     if x[0] % 2 == 0:
5         return 1 + fun1(x[1:])
6     else:
7         return fun1(x[1:])
```

`l` is a non-empty list of positive integers, which is already defined. Which statement is correct about the recursive function `fun1(l)`?

- (a) It returns total number of odd elements in the list `l`
- (b) It returns total number of even elements in the list `l`
- (c) It returns sum of the even elements in the list `l`
- (d) It returns sum of the odd elements in the list `l`

Answers

(b)

Solution

For each call of `fun1` if the first element of the list is even then add 1 in return value and recursively call `fun1` again with parameter list after removing the first element until the list becomes empty. So finally the function returns the total number of even elements in the list `l`.

Problem 4

Question-5

```
1 | def fun2(l,d):  
2 |     if l == []:  
3 |         return 0  
4 |     if l[-1] % d == 0:  
5 |         return l[-1] + fun2(l[:-1],d)  
6 |     else:  
7 |         return fun2(l[:-1],d)
```

`l` is a non-empty list of positive integers, which is already defined. What does the following statement print?

```
1 | print(fun2(l,1)-fun2(l,2))
```

- (a) Total number of odd elements in the list `l`
- (b) Total number of even elements in the list `l`
- (c) Total number of elements in the list `l`
- (d) Sum of odd elements in the list `l`
- (e) Sum of even elements in the list `l`

Answers

(d)

Solution

`fun2(l,1)` returns the sum of all elements in the list `l` and `fun2(l,2)` returns the sum of all even elements in the list `l` then `print(fun2(l,1)-fun2(l,2))` prints sum of all odd elements in the list `l`

Problem 5

Question-6

Consider the following function:

```
1 def countVowels(s):
2     count = 0
3     for i in s:
4         if i.lower() in 'aeiou':
5             count += 1
6     return count
```

Which of the following functions is the recursive equivalent of the code given above, where `s` is a string? It is MSQ type question.

(a)

```
1 def countVowels(s):
2     if s == '':
3         return 0
4     #only lowercase vowels will be count, due to this it is an incorrect code
5     if s[len(s)-1] in 'aeiou':
6         return 1 + countVowels(s[:len(s)-1])
7     else:
8         return 0 + countVowels(s[:len(s)-1])
```

(b)

```
1 def countVowels(s):
2     if s == '':
3         return 0
4     if s[0].lower() in 'aeiou':
5         return 1 + countVowels(s[1:])
6     else:
7         return 0 + countVowels(s[1:])
```

(c)

```
1 def countVowels(s):
2     if s == '':
3         return 0
4     if s[-1].lower() in 'aeiou':
5         return 1 + countVowels(s[:-1])
6     else:
7         return 0 + countVowels(s[:-1])
```

(d)

```
1 def isVowel(char):
2     char = char.lower()
3     vowels = "aeiou"
4     if char in vowels :
5         return 1
6     else:
7         return 0
8 def countVowels(s):
9     n = len(s)
10    if n == 1:
11        return isVowel(s[0])
12    return countVowels(s[:-1]) + isVowel(s[n-1])
```

Answer

(b), (c) and (d)

Solution

Option (a) is not correct because vowels in the upper-case will not be covered in counting by `countVowels(s)`. Other option (b), (c) and (d) are correct.

Problem 6

Question-7

```
1 i = 0
2 s = 0
3 while i < len(l):
4     for i in range(i,l[i]):
5         s += l[i]
6         i += 1
7     i += 1
```

Assume `l` is a list `[1, 2, 3, 4, 5, 6, 7, 8, 9]`. What will be the value of `s` after completion of 4th iteration of the while loop?

- (a) 17
- (b) 9
- (c) 16
- (d) 25
- (d) 50

Answer

(c)

Solution

After first iteration of while:

```
for i in range(0,1)

s = 0 + 1 because l[0] = 1

i = 2, s = 1
```

After second iteration of while:

```
for i in range(2,3)

s = 1 + 3 because l[2] = 3

i = 4, s = 4
```

After third iteration of while:

```
for i in range(4,5)

s = 4 + 5 because l[4] = 5

i = 6, s = 9
```

After fourth iteration of while:

```
for i in range(6,7)

s = 9 + 7 because l[6] = 7
```

i = 8, s = 16

Hence correct option is (c)

Problem 7

Question-8

```
1 def BinToDec(n):
2     if n == 0:
3         return 0
4     else:
5         return ((n % 10 + 2 * BinToDec(n // 10)))
```

Which of these are equivalent iterative code of the above recursive code where `n` is a binary number? It is MSQ type question.

Note- Binary to Decimal conversion method example:-

Binary = 11101

Decimal = $1 \times (2^4) + 1 \times (2^3) + 1 \times (2^2) + 0 \times (2^1) + 1 \times (2^0)$

Decimal = 29

(a)

```
1 def BinToDec(n):
2     n = str(n)
3     q = 0
4     for i in range (len(n)-1,-1,-1):
5         #(len(n)-i) generate value from 1.It should be (len(n)-1-i) for
correct answer
6         q += (int(n[i]) * (2 ** (len(n)-i)))
7     return q
```

(b)

```
1 def BinToDec(n):
2     n = str(n)
3     d = 0
4     for i in range (len(n)-1,-1,-1):
5         d += (int(n[i]) * (2 ** (len(n)-1-i)))
6     return d
```

(c)

```
1 def BinToDec(n):
2     decimal, i = 0, 0
3     while(n != 0):
4         # operator // should be replace by % for correct answer
5         dec = n // 10
6         decimal = decimal + dec * pow(2, i)
7         n = n // 10
8         i += 1
9     return(decimal)
```

(d)

```
1 def BinToDec(n):
2     decimal, i = 0, 0
3     while(n != 0):
4         dec = n % 10
5         decimal = decimal + dec * pow(2, i)
6         n = n // 10
7         i += 1
8     return(decimal)
```

Answer

(b) and (d)

Solution

Option (a) is not correct because in this code exponent value will start from 1 (right to left). It should be started from 0 for the correct answer. for example:-

Binary = 11101

Decimal = $1 \times (2^5) + 1 \times (2^4) + 1 \times (2^3) + 0 \times (2^2) + 1 \times (2^1)$

Decimal = 58 (it is wrong)

Option (c) is not correct because in line number 5 operator `//` should be replaced by `%` for the correct answer.

Other options (b) and (d) are correct.