

Question Number : 65 Question Id : 640653565432 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following denotes the column space of A?(More than one option may be correct)

Options :

6406531889949. ✓ $\mathbb{R}^3$

6406531889950. ✗ $\text{Span}\{(-1, -1, 1)\}$

6406531889951. ✗ $\text{Span}\{(3, 6, 2), (3, -2, 2)\}$

6406531889952. ✓ $\text{Span}\{(3, 6, 2), (3, -2, 2), (-1, -1, 1)\}$

Sem2 Intro to Python

Section Id :	64065338385
Section Number :	4
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	26
Number of Questions to be attempted :	26
Section Marks :	100
Display Number Panel :	Yes
Group All Questions :	No

Enable Mark as Answered Mark for Review and
Clear Response : Yes
Maximum Instruction Time : 0
Sub-Section Number : 1
Sub-Section Id : 64065380791
Question Shuffling Allowed : No
Is Section Default? : null

Question Number : 66 Question Id : 640653565433 Question Type : MCQ Is Question
Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction
Time : 0
Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "FOUNDATION LEVEL : SEMESTER 2:
INTRODUCTION TO PYTHON (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE [TOP](#) FOR THE SUBJECTS
REGISTERED BY YOU)

Options :

6406531889953.  YES

6406531889954.  NO

Sub-Section Number : 2
Sub-Section Id : 64065380792
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 67 Question Id : 640653565434 Question Type : MCQ Is Question
Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What is the output of the following snippet of code?

```
1 def welcome(age):  
2     if 0 < age < 18:  
3         print('welcome boys and girls')  
4     else:  
5         print('welcome ladies and gentlemen')  
6  
7 print(welcome(50))
```

Options :

6406531889955. ✖ 1 | welcome boys and girls

6406531889956. ✖ 1 | welcome ladies and gentlemen

6406531889957. ✖ 1 | welcome boys and girls
2 | None

6406531889958. ✔ 1 | welcome ladies and gentlemen
2 | None

Question Number : 68 Question Id : 640653565435 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Which of the following code blocks prints the product of the digits for the given number?

Options :

6406531889959. ✓

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

6406531889960. ✖

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

6406531889961. ✖

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

6406531889962. ✖

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

Question Number : 69 Question Id : 640653565436 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What will be the output of the following Python code?

```
1 string1 = 'programming'
2 string2 = 'python'
3 L = []
4 for i in range(0, len(string1)):
5     for j in range(0, len(string2)):
6         if (string1[i] == string2[j]):
7             L.append(string1[i])
8             break
9         else:
10            continue
11 print(L)
```

Options :

6406531889963. ✖

1 | ['p', 'o', 'n', 'p', 'o', 'n']

6406531889964. ✖

1 | ['p', 'p', 'o', 'o', 'n', 'n']

6406531889965. ✔

1 | ['p', 'o', 'n']

6406531889966. ✖ None of these

Question Number : 70 Question Id : 640653565437 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following snippet of code.

```
1 def func(L):  
2     if L == []:  
3         return 0  
4     if L[-1] % 2 == 1:  
5         return L[-1] + func(L[:-1])  
6     else:  
7         return func(L[:-1])
```

If `L` is a non-empty list of positive integers, which of the following statements is correct about the recursive function `func(L)` ?

Options :

6406531889967. ✖ It returns total number of odd elements in the list `L`

6406531889968. ✖ It returns total number of even elements in the list `L`

6406531889969. ✖ It returns sum of the even elements in the list `L`

6406531889970. ✔ It returns sum of the odd elements in the list `L`

Question Number : 71 Question Id : 640653565438 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What will be the output of the following Python code?

```
1 A = [[2, 2, 3], [3, 4, 5], [1, 0, 7]]
2 B = [[1, 1, 4], [4, 5, 2], [1, 4, 1]]
3 n = len(A)
4 C = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
5 for i in range(n):
6     for j in range(n):
7         C[i][j] = B[i][j] - A[i][j]
8         if j != n - 1:
9             print(C[i][j], end = ',')
10        else:
11            print(C[i][j])
```

Options :

1	3,3,-7
2	-7,-9,7
3	2,-4,8

6406531889971. ✖

1	3,3,7
2	7,9,7
3	2,4,8

6406531889972. ✖

1	1,1,1
2	1,1,3
3	0,4,6

6406531889973. ✖

1	-1,-1,1
2	1,1,-3
3	0,4,-6

6406531889974. ✔

Question Number : 72 Question Id : 640653565441 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Select the correct implementation of a program that creates a text file named `pattern.txt` with the following contents:

```
1 | az
2 | by
3 | cx
4 | dw
5 | ev
6 | fu
7 | gt
8 | hs
9 | ir
10 | jq
11 | kp
12 | lo
13 | mn
```

Options :

```
1 | f = open('pattern.txt', 'r')
2 | letters = 'abcdefghijklmnopqrstuvwxyz'
3 | n = len(letters) // 2
4 | for i in range(n):
5 |     line = letters[i] + letters[-1 - i]
6 |     if i != n - 1:
7 |         line = line + '\n'
8 |     f.write(line)
9 | f.close()
```

6406531889984. ✖

```
1 | f = open('pattern.txt', 'w')
2 | letters = 'abcdefghijklmnopqrstuvwxyz'
3 | n = len(letters) // 2
4 | for i in range(n):
5 |     line = letters[i] + letters[-1 - i]
6 |     f.write(line)
7 | f.close()
```

6406531889985. ✖

6406531889986. ✔

```

1 f = open('pattern.txt', 'w')
2 letters = 'abcdefghijklmnopqrstuvwxyz'
3 n = len(letters) // 2
4 for i in range(n):
5     line = letters[i] + letters[-1 - i]
6     if i != n - 1:
7         line = line + '\n'
8     f.write(line)
9 f.close()

```

```

1 f = open('pattern.txt', 'w')
2 letters = 'abcdefghijklmnopqrstuvwxyz'
3 n = len(letters)
4 for i in range(n):
5     line = letters[i] + letters[-1 - i]
6     if i != n - 1:
7         line = line + '\n'
8     f.write(line)
9 f.close()

```

6406531889987. ✖

Sub-Section Number :	3
Sub-Section Id :	64065380793
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 73 Question Id : 640653565439 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the following functions:

```
1 def f(n):
2     if n < 10:
3         return 1
4     return 1 + f(n // 10)
5
6 def g(n):
7     if n < 10:
8         return n
9     return n % 10 + g(n // 10)
10
11 def h(n):
12     if n <= 1:
13         return n
14     return n * h(n - 1)
```

Which of the following function calls returns the number of digits in 100!, where, $n!$ is the product of the first n positive integers?

Options :

6406531889975. ✖ `f(g(100))`

6406531889976. ✔ `f(h(100))`

6406531889977. ✖ `g(h(100))`

6406531889978. ✖ `h(g(100))`

6406531889979. ✖ `h(f(100))`

Question Number : 74 Question Id : 640653565440 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

What will be the output of the following Python code?

```
1 def new_fun(a):
2     s_L = []
3     while a != []:
4         x = a[0]
5         for i in a:
6             if i < x:
7                 x = i
8         a.remove(x)
9         s_L.append(x)
10    return s_L
11
12 L = [6, 2, 9, 8]
13 print(new_fun(L))
```

Options :

6406531889980. ✓

1 | [2, 6, 8, 9]

6406531889981. ✗

1 | [9, 8, 6, 2]

6406531889982. ✗

1 | [2, 6, 9, 8]

6406531889983. ✗

1 | [9, 8, 2, 6]

Question Number : 75 Question Id : 640653565442 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

What is the output of the following snippet of code?

```
1 L = [1, 3, -1, 4, -2, 5, 3]
2 try:
3     n = 10
4     for i in range(n):
5         if L[i] < 0:
6             L[i] = 0
7 except IndexError:
8     for i in range(n - len(L)):
9         L.append(0)
10 finally:
11     print(L)
```

Options :

6406531889988. ✖

1 | [1, 3, -1, 4, -2, 5, 3]

6406531889989. ✖

1 | [1, 3, 0, 4, 0, 5, 3]

6406531889990. ✔

1 | [1, 3, 0, 4, 0, 5, 3, 0, 0, 0]

6406531889991. ✖ This code doesn't print anything to the console

Question Number : 76 Question Id : 640653565443 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the following snippet of code:

```
1 | M = [[0 for j in range(3)] for i in range(3)]
```

Select the snippet of code that is equivalent to this.

Options :

```
1 | M = []
2 | for i in range(3):
3 |     M.append([])
4 |     for j in range(3):
5 |         M[-1].append(0)
```

6406531889992. ✓

```
1 | M = []
2 | for i in range(3):
3 |     M.append([])
4 |     for j in range(3):
5 |         M[j].append(0)
```

6406531889993. ✗

```
1 | M = []
2 | for i in range(3):
3 |     M.append([])
4 |     for j in range(3):
5 |         M[0].append(0)
```

6406531889994. ✗

```
1 | M = []
2 | for i in range(3):
3 |     for j in range(3):
4 |         M[i][j].append(0)
```

6406531889995. ✗

Question Number : 77 Question Id : 640653565445 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

People from four different cities have come for an interview: Amethi, Trichy, Goa and Cochin. They are all seated on a long row of chairs waiting for their turn. A seating plan is termed **good** if no person finds himself sandwiched between two persons from the same city that is different from his.

For example, ['C', 'A', 'T', 'G', 'T'] is a **bad** seating plan as the person from Goa finds two persons from Trichy seated on either side. ['A', 'T', 'T', 'T', 'G', 'A', 'T'] is a **good** plan. Even though the person from Trichy is sitting between two persons from Trichy, they are still from his city.

This seating plan is represented by a list **L**. Write a function named **plan_type** that accepts **L** as argument. It should return the string **'good'** if it is a good seating plan, and **'bad'** otherwise. You can assume that $\text{len}(L) \geq 3$. Sample input-output behavior is given below.

L	plan_type(L)
['A', 'T', 'G', 'T', 'C', 'A']	'bad'
['A', 'T', 'T', 'T', 'G', 'A', 'T']	'good'
['A', 'C', 'G', 'T', 'A', 'C', 'T', 'G', 'C']	'good'
['A', 'T', 'A', 'C']	'bad'
['A', 'A', 'A', 'A', 'T', 'G', 'C']	'good'

Select the correct implementations of this function.

Options :

```
1 def plan_type(L):
2     for i in range(1, len(L) - 1):
3         if (L[i - 1] == L[i + 1]) and (L[i] != L[i - 1]):
4             return 'bad'
5     return 'good'
```

6406531890000. ✓

```
1 def plan_type(L):
2     for i in range(1, len(L) - 1):
3         if (L[i - 1] == L[i + 1]) or (L[i] != L[i - 1]):
4             return 'bad'
5     return 'good'
```

6406531890001. ✖

```

1 def plan_type(L):
2     for i in range(1, len(L) - 1):
3         if (L[i - 1] == L[i + 1]) and (L[i] != L[i - 1]):
4             return 'good'
5     return 'bad'

```

6406531890002. ✖

```

1 def plan_type(L):
2     for i in range(1, len(L) - 1):
3         if (L[i - 1] == L[i + 1]):
4             return 'bad'
5     return 'good'

```

6406531890003. ✖

```

1 def plan_type(L):
2     for i in range(1, len(L) - 1):
3         if (L[i - 1] == L[i] == L[i + 1]):
4             return 'bad'
5     return 'good'

```

6406531890004. ✖

Sub-Section Number :

4

Sub-Section Id :

64065380794

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 78 Question Id : 640653565444 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5

Question Label : Multiple Choice Question

The variable `number` represents a positive integer. Which of the following options are equivalent to the below code block?

```
1 temp = number
2 total = 0
3 times = len(str(number))
4 while temp > 0:
5     remainder = temp % 10
6     total = total + (remainder ** times)
7     temp = temp // 10
8 if number == total:
9     print(True)
10 else:
11     print(False)
```

Options :

```
1 L1 = list(map(int, str(number)))
2 L2 = list(map(lambda x: x ** len(str(number)), number))
3 if sum(L2) == number:
4     print(True)
5 else:
6     print(False)
```

6406531889996. ✖

```
1 L1 = list(map(int, str(number)))
2 L2 = list(map(lambda x: x ** len(str(number)), L1))
3 print(sum(L2) == number)
```

6406531889997. ✔

```
1 L1 = list(map(int, number))
2 L2 = list(map(lambda x: x ** len(str(number)), L1))
3 print(sum(L2) == number)
```

6406531889998. ✖

```
1 L1 = list(map(int, str(number)))
2 L2 = list(map(lambda x: x ** len(number), L1))
3 if sum(L2) == number:
4     print(True)
5 else:
6     print(False)
```

6406531889999. ✖

Sub-Section Number :	5
Sub-Section Id :	64065380795
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 79 Question Id : 640653565448 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

The variable `s` is a set in which of the following options?

Options :

6406531890015. ✓

1 | `s = set()`

6406531890016. ✗

1 | `s = {}`

6406531890017. ✓

1 | `s = {1, 2, 3}`

6406531890018. ✓

1 | `s = set([1, 2, 3, 4, 3, 2, 1])`

Question Number : 80 Question Id : 640653565453 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Select the correct code block(s) that perform the following task:

1. Accept two strings as input
2. Print `Ascending` if the first string comes before the second in alphabetical order
3. `Descending` otherwise

For example, if the input is as follows:

```
1 java
2 python
```

then the expected output is `Ascending` because `python` comes after `java`. Assume that all the strings consist of only one word and will be entered in lowercase during the time of input.

Options :

```
1 str1 = input()
2 str2 = input()
3 if(str1 < str2):
4     print("Ascending")
5 else:
6     print("Descending")
```

6406531890036. ✓

```
1 str1 = input()
2 str2 = input()
3 if(str1 > str2):
4     print("Ascending")
5 else:
6     print("Descending")
```

6406531890037. ✗

```
1 if(input() < input()):
2     print("Ascending")
3 else:
4     print("Descending")
```

6406531890038. ✓

```
1 if(input() > input()):
2     print("Ascending")
3 else:
4     print("Descending")
```

6406531890039. ✗

6406531890040. ✓

```

1 str1 = input()
2 str2 = input()
3 result = str1 < str2
4 if(result == True):
5     print("Ascending")
6 else:
7     print("Descending")

```

```

1 str1 = input()
2 str2 = input()
3 result = str1 > str2
4 if(result == True):
5     print("Ascending")
6 else:
7     print("Descending")

```

6406531890041. ✖

Sub-Section Number :

6

Sub-Section Id :

64065380796

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 81 Question Id : 640653565446 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question

Select all correct implementations of a function named `unique` that accepts a non-empty list `L` of integers as an argument. The function should remove all duplicate elements from the list `L`. Specifically, It should return a list `L_uniq` that retains the first occurrence (from the left) of each distinct element in the input list `L`. For example,

L	unique(L)
[1, 1, 2, 3, 5, 5]	[1, 2, 3, 5]
[1, 2, 3, 4, 5, 7, 7]	[1, 2, 3, 4, 5, 7]

Options :

```

1 def unique(L):
2     L_uniq = []
3     for elem in L:
4         if elem not in L_uniq:
5             L_uniq.append(elem)
6     return L_uniq

```

6406531890005. ✓

```

1 def unique(L):
2     L_uniq = []
3     for elem in L:
4         if elem in L_uniq:
5             L_uniq.append(elem)
6     return L_uniq

```

6406531890006. ✖

```

1 def unique(L):
2     L_uniq = [L[0]]
3     for i in range(1, len(L)):
4         if not(L[i] in L[:i]):
5             L_uniq.append(L[i])
6     return L_uniq

```

6406531890007. ✓

```

1 def unique(L):
2     L_uniq = []
3     for i in range(1, len(L)):
4         if not(L[i] in L[:i]):
5             L_uniq.append(L[i])
6     return L_uniq

```

6406531890008. ✖

```

1 def unique(L):
2     L_uniq = []
3     for i in range(0, len(L)):
4         if not(L[i] in L[i + 1: ]):
5             L_uniq.append(L[i])
6     return L_uniq

```

6406531890009. ✖

Question Number : 82 Question Id : 640653565449 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question

Write a function named `length_analysis` that accepts a list of distinct words `L` as argument. It should return a dictionary `D` that has the following structure:

- **keys:** words in the list
- **values:** length of the word (key)

Select all correct implementations of this function.

Options :

```
1 def length_analysis(L):
2     D = dict()
3     for word in L:
4         D[word] = len(word)
5     return D
```

6406531890019. ✓

```
1 def length_analysis(L):
2     D = {}
3     for word in L:
4         D[word] = len(word)
5     return D
```

6406531890020. ✓

```
1 def length_analysis(L):
2     D = dict()
3     for word in L:
4         D[word] = D[word] + len(word)
5     return D
```

6406531890021. ✗

```
1 def length_analysis(L):
2     D = {}
3     for word in L:
4         D[word] = D[word] + len(word)
5     return D
```

6406531890022. ✗

Question Number : 83 Question Id : 640653565450 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question

Select the correct code snippet(s) which can be used to reverse the given integer `num`. For example, if the input is 3456 then the output should be 6543.

Options :

6406531890023. ✓

```
1 reversed_num = 0
2 while num != 0:
3     digit = num % 10
4     reversed_num = reversed_num * 10 + digit
5     num //= 10
6 print(reversed_num)
```

6406531890024. ✓

```
1 print(str(num)[::-1])
```

6406531890025. ✗

```
1 reversed_num = 0
2 while num != 0:
3     digit = num % 10
4     num /= 10
5     reversed_num = reversed_num * 10 + digit
6 print(reversed_num)
```

6406531890026. ✗

```
1 print(str(num)[-1::])
```

Sub-Section Number :

7

Sub-Section Id :

64065380797

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 84 Question Id : 640653565447 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Consider the following snippet of code:

```
1 def f(word, n):
2     P = dict()
3     for c in word:
4         if c not in P:
5             P[c] = 0
6             P[c] += 1
7     for c in P:
8         if P[c] == n:
9             return True
10    return False
11
12 if some_word.isalpha():
13     print(True)
14 print(f(some_word, 5))
```

`some_word` is a string variable that has already been defined. The above code runs without any errors.

The output when the code given above is executed is as follows:

```
1 True
2 True
```

Which of the following statements are True? Note that your answer should hold for any value of the string `some_word` that results in the above output.

Options :

6406531890010. ✓ `some_word` is only made up of alphabets

6406531890011. ✗ `some_word` could also contain numbers

6406531890012. ✓ `len(some_word)` is at least 5

6406531890013. ✓ At least one letter in the English alphabet occurs exactly five times in `some_word`

6406531890014. ✗ Exactly one letter in the English alphabet occurs at least five times in `some_word`

Question Number : 85 Question Id : 640653565451 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

We are not sure if the file `file.txt` exists in the current directory. Select all error-free implementations of a program that reads this file and prints one line of the file at a time. If the file doesn't exist, the code should terminate gracefully.

Options :

```
1 try:
2     f = open('file.txt', 'r')
3     for line in f:
4         print(line.strip())
5 except FileNotFoundError:
6     print('This file does not exist')
```

6406531890027. ✓

```
1 try:
2     f = open('file.txt', 'r')
3     for line in f:
4         print(line.strip())
5 except:
6     print('This file does not exist')
```

6406531890028. ✓

```
1 try:
2     f = open('file.txt', 'r')
3     for line in f:
4         print(line.strip())
5 except ValueError:
6     print('This file does not exist')
```

6406531890029. ✗

```

1  try:
2      f = open('file.txt', 'r')
3      for line in f:
4          print(line.strip())
5  except FileNotFoundError:
6      print('This file does not exist')
7  except:
8      print('This is for all other errors that might come up')

```

6406531890030. ✓

Question Number : 86 Question Id : 640653565452 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

Let L be a $m \times m$ 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 `rowsum`.

Rows are indexed from 0 to $m - 1$. For the j^{th} row, `rowsum[j]` should be the sum of all elements in that row. Select all correct code snippets that achieve this.

Options :

```

1  rowsum = []
2  m = len(L)
3  row, col = 0, 0
4  while col < m:
5      rowsum.append(0)
6      while row < m:
7          rowsum[col] += L[row][col]
8          row += 1

```

6406531890031. ✖

6406531890032. ✖

```

1 rowsum = []
2 m = len(L)
3 row, col = 0, 0
4 while col < m:
5     rowsum.append(0)
6     while row < m:
7         rowsum[col] += L[col][row]
8         row += 1
9     col += 1

```

```

1 rowsum = []
2 m = len(L)
3 row, col = 0, 0
4 while col < m:
5     row = 0
6     rowsum.append(0)
7     while row < m:
8         rowsum[col] += L[col][row]
9         row += 1
10    col += 1

```

6406531890033. ✓

```

1 rowsum = []
2 m = len(L)
3 for j in range(m):
4     val = 0
5     for i in range(m):
6         val = val + L[i][j]
7     rowsum.append(val)

```

6406531890034. ✖

```

1 rowsum = []
2 m = len(L)
3 for j in range(m):
4     val = 0
5     for i in range(m):
6         val = val + L[j][i]
7     rowsum.append(val)

```

6406531890035. ✓

Sub-Section Number :

8

Sub-Section Id :

64065380798

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 87 Question Id : 640653565454 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Short Answer Question

`words.txt` is a text file with the following contents:

```
1 no
2 on
3 one
4 two
5 a
6 an
7 the
8 six
9 fin
10 out
```

What is the output of the following snippet of code?

```
1 def do_something(filename):
2     f = open(filename, 'r')
3     maxword = f.readline().strip()
4     count = 1
5     for line in f:
6         word = line.strip()
7         if len(word) > len(maxword):
8             maxword = word
9             count = 1
10        elif len(word) == len(maxword):
11            count += 1
12    f.close()
13    return count
14
15 print(do_something('words.txt'))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

6

Question Number : 88 Question Id : 640653565455 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Short Answer Question

Consider the following snippet of code:

```
1 L = [1, 2, 3, 4, 5]
2 S = []
3 T = 0
4 i = 0
5 while i < len(L):
6     S += L[:i] + L[i:]
7     for j in S:
8         T += j
9     i += 1
```

What will be the value of `T` at the end of execution of the above code?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

225

Sub-Section Number : 9

Sub-Section Id : 64065380799

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 89 Question Id : 640653565456 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Short Answer Question

What is the output of the following snippet of code?

```
1 def some_fun(x):  
2     if x < 3:  
3         return 0  
4     return 1 + some_fun(x // 3)  
5  
6 print(some_fun(59049))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

10

Sub-Section Number : 10

Sub-Section Id : 64065380800

Question Shuffling Allowed : No

Is Section Default? : null

Question Id : 640653565457 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Question Numbers : (90 to 93)

Question Label : Comprehension

Assume the values of Boolean variables **A** and **B** are **True** and **False** respectively.

What will be output of the given code snippets?

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 90 Question Id : 640653565458 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

```
1 | print(not (A and B), not(A) and B)
```

Options :

6406531890045. ✖ 1 | False False

6406531890046. ✔ 1 | True False

6406531890047. ✖ 1 | False True

6406531890048. ✖ 1 | True True

Question Number : 91 Question Id : 640653565459 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

```
1 | print(A or B and A or not(B))
```

Options :

6406531890049. ✖ 1 | False

6406531890050. ✖ 1 | True False

6406531890051.



1 | False True

6406531890052. ✓

1 | True

Question Number : 92 Question Id : 640653565460 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

```
1 | print(not(A or B), not(A) or B)
```

Options :

6406531890053. ✓

1 | False False

6406531890054. ✗

1 | True False

6406531890055. ✗

1 | False True

6406531890056. ✗

1 | True True

Question Number : 93 Question Id : 640653565461 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

```
1 | print(not A and B or A and B)
```

Options :

6406531890057. ✓

1 | False

6406531890058. ✖

1 | True False

6406531890059. ✖

1 | False True

6406531890060. ✖

1 | True

Sub-Section Number :	11
Sub-Section Id :	64065380801
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653565462 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0 Question Numbers : (94 to 95) Question Label : Comprehension

`Rational` is a class that represents positive rational numbers. Recall that a rational number x is of the form $\frac{p}{q}$, where the greatest common divisor of (p, q) is equal to 1, with $q \neq 0$. In this case, we will be modeling only positive rationals as a class.

```
1 class Rational:
2     def __init__(self, num, den):
3         self.num = num
4         self.den = den
5         self.reduce()
6
7     def print_info(self):
8         if self.den == 1:
9             print(f'{self.num}')
10        else:
11            print(f'{self.num}/{self.den}')
12
13    def reduce(self):
14        for i in range(min(self.num, self.den), 1, -1):
15            if self.num % i == 0 and self.den % i == 0:
16                self.num = self.num // i
17                self.den = self.den // i
18                break
19
20    def divide(self, k):
21        self.den = self.den * k
22        self.reduce()
```

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 94 Question Id : 640653565463 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

What is the output of the following snippet of code?

```
1 x = Rational(15, 24)
2 x.divide(5)
3 x.print_info()
```

Options :

6406531890061. ✖

1 | 3/24

6406531890062. ✔

1 | 1/8

6406531890063. ✖

1 | 15/24

6406531890064. ✖

1 | 5/24

Question Number : 95 Question Id : 640653565464 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5

Question Label : Multiple Choice Question

A new method `add` should be introduced into the class to add two rational numbers. This method should accept another rational number number, say `rat`, as argument and return the sum of the current rational number and `rat`.

Select the correct implementation of this method.

Sample behavior of this method is given below:

Input:

```
1 | a = Rational(10, 2)
2 | b = Rational(10, 3)
3 | c = a.add(b)
4 | c.print_info()
```

Output:

1 | 25/3

Options :

6406531890065. ✖

```
1 def add(self, rat):
2     out = Rational(1, 1)
3     out.num = self.num + rat.num
4     out.den = self.den + rat.den
5     return out
```

6406531890066. ✖

```
1 def add(self, rat):
2     out = Rational(1, 1)
3     out.num = self.num + rat.num
4     out.den = self.den + rat.den
5     out.reduce()
6     return out
```

6406531890067. ✖

```
1 def add(rat):
2     out = Rational(1, 1)
3     out.den = self.den * rat.den
4     out.num = self.num * rat.den + rat.num * self.den
5     out.reduce()
6     return out
```

6406531890068. ✔

```
1 def add(self, rat):
2     out = Rational(1, 1)
3     out.den = self.den * rat.den
4     out.num = self.num * rat.den + rat.num * self.den
5     out.reduce()
6     return out
```

Sem2 English2

Section Id :	64065338386
Section Number :	5
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	36
Number of Questions to be attempted :	36