

tes62.py X tes63.py tes64.py tes65.py tes66.py tes67.py tes68.py

D: > sam > python > tes62.py > ...

```
2 class RomanNumerals:
23     @staticmethod
24     def from_roman(roman):
25         i = 0
26         num = 0
27         while i < len(roman):
28             if i+1 < len(roman) and roman[i:i+2] in RomanNumerals.roman_to_int_map:
29                 num += RomanNumerals.roman_to_int_map[roman[i:i+2]]
30                 i += 2
31             else:
32                 num += RomanNumerals.roman_to_int_map[roman[i]]
33                 i += 1
34         return num
35
36 converter = RomanNumerals()
37 print(converter.to_roman(54))
38 print(converter.from_roman('CIV'))
```

tes62.py

tes63.py X

tes64.py

tes65.py

tes66.py

tes67.py

D: &gt; sam &gt; python &gt; tes63.py &gt; ...

```
1  #ass-7 2nd question
2  class py_solution:
3      def is_valid_parenthesis(self, str1):
4          stack, pchar = [], {"(": ")", "{": "}", "[": "]" }
5          for parenthesis in str1:
6              if parenthesis in pchar:
7                  stack.append(parenthesis)
8              elif len(stack) == 0 or pchar[stack.pop()] != parenthesis:
9                  return False
10         return len(stack) == 0
11
12 print(py_solution().is_valid_parenthesis("(){}[]"))
13 print(py_solution().is_valid_parenthesis("()[{}]()"))
14 print(py_solution().is_valid_parenthesis("()"))
15
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
True
False
True
```

tes62.py tes63.py tes64.py X tes65.py tes66.py tes67.py tes68.py tes69.py

D: > sam > python > tes64.py > ...

```
1  #ass-7 3rd question
2  class py_solution:
3      def sub_sets(self, sset):
4          return self.subsetsRecur([], sorted(sset))
5
6      def subsetsRecur(self, current, sset):
7          if sset:
8              return self.subsetsRecur(current, sset[1:]) + self.subsetsRecur(current + [sset[0]], sset[1:])
9          return [current]
10
11 print(py_solution().sub_sets([4,5,6]))
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

True  
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes64.py  
[[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

tes62.py

tes63.py

tes64.py

tes65.py

X

tes66.py

tes67.py

tes68.py

D: &gt; sam &gt; python &gt; tes65.py &gt; ...

```
1  #ass-7 4th question
2  class PairFinder:
3      def find_pair_indices(self, nums, target):
4          lookup = {}
5          for i, num in enumerate(nums):
6              if target - num in lookup:
7                  return lookup[target - num], i
8              lookup[num] = i
9
10 # Example usage
11 numbers = [90, 20, 10, 40, 50, 60, 70]
12 target = 50
13 pair_finder = PairFinder()
14 index1, index2 = pair_finder.find_pair_indices(numbers, target)
15 print(f"Indices: {index1}, {index2}")
16
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes65.py
Indices: 2, 3
```

tes62.py

tes63.py

tes64.py

tes65.py

tes66.py X

tes67.py

D: &gt; sam &gt; python &gt; tes66.py &gt; ...

```
1  #ass-7 5th question
2  def findTriplets(arr, n):
3
4      found = False
5      for i in range(0, n-2):
6
7          for j in range(i+1, n-1):
8
9              for k in range(j+1, n):
10
11                  if (arr[i] + arr[j] + arr[k] == 0):
12                      print(arr[i], arr[j], arr[k])
13                      found = True
14
15      if (found == False):
16          print(" not exist ")
17
18  arr = [-25, -10, -7, -3, 2, 4, 8, 10]
19  n = len(arr)
20  findTriplets(arr, n)
21
22
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
-10 2 8
-7 -3 10
```

 tes62.py tes63.py tes64.py tes65.py tes66.py tes67.py

✕

 tes68.pyD: > sam > python >  tes67.py > ...

```
1  #ass-7 6th question
2  class Power:
3      def __init__(self, x, n):
4          self.x = x
5          self.n = n
6
7      def power(self):
8          return pow(self.x, self.n)
9
10 p = Power(8, 2)
11 print(p.power())
12
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes67.py
64
```

tes62.py

tes63.py

tes64.py

tes65.py

tes66.py

tes67.py

tes68.py

X

D: > sam > python > tes68.py > ...

```
1 #ass-7 7th question
2 class py_solution3:
3     def reverse_words(self, s):
4         return ' '.join(reversed(s.split()))
5
6
7 print(py_solution3().reverse_words('hello .py'))
8
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes68.py
.py hello
PS C:\Users\hp>
```

[tes62.py](#)[tes63.py](#)[tes64.py](#)[tes65.py](#)[tes66.py](#)[tes67.py](#)[tes68.py](#)D: > sam > python > [tes69.py](#) > ...

```
1  #ass-7 8th question
2  class StringManipulator:
3      def __init__(self):
4          self.user_string = ""
5
6      def get_string(self):
7          self.user_string = input("Enter a string: ")
8
9      def print_string(self):
10         print("Reversed string:", self.user_string[::-1])
11
12  # Example usage:
13  if __name__ == "__main__":
14      string_manipulator = StringManipulator()
15      string_manipulator.get_string()
16      string_manipulator.print_string()
17
18
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes69.py
Enter a string: Mangalore
Reversed string: erolagnaM
```



tes63.py

tes64.py

tes65.py

tes66.py

tes67.py

tes68.py

tes69.py

D: &gt; sam &gt; python &gt; tes70.py &gt; ...

```
1  #ass-7 9th question
2  class Circle():
3      def __init__(self, r):
4          self.radius = r
5
6      def area(self):
7          return self.radius**2*3.14
8
9      def perimeter(self):
10         return 2*self.radius*3.14
11
12     NewCircle = Circle(4)
13     print(NewCircle.area())
14     print(NewCircle.perimeter())
15
```

PROBLEMS




OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes70.py
50.24
25.12
```

 tes64.py tes65.py tes66.py tes67.py tes68.py tes69.py tes70.pyD: > sam > python >  tes71.py > ...

```
1  #ass-7 10th question
2  class food:
3      def items():
4          pass
5
6  c = food()
7  print(c.__class__)
8  classes = c.__class__
9  print(classes.__name__)
10 |
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes71.py
<class '__main__.food'>
food
```

tes65.py

tes66.py

tes67.py

tes68.py

tes69.py

tes70.py

tes71.py

D: > sam > python > tes72.py > ...

```
1 #ass-7 lambda 1st question
2 r=lambda a:a+15
3 print(r(10))
4 r=lambda x,y:x*y
5 print(r(12,4))
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes72.py
25
48
```

 tes66.py tes67.py tes68.py tes69.py tes70.py tes71.py tes72.pyD: > sam > python >  tes73.py > ...

```
1 #ass-7 lambda 2nd question
2 subject_marks = [('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]
3 print("Original list of tuples are:")
4 print(subject_marks)
5 subject_marks.sort(key=lambda x: x[1])
6 print("\nSorted List of Tuples are:")
7 print(subject_marks)
8
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
Sorted List of Tuples are:
[('Social sciences', 82), ('English', 88), ('Science', 90), ('Maths', 97)]
```

[tes67.py](#)[tes68.py](#)[tes69.py](#)[tes70.py](#)[tes71.py](#)[tes72.py](#)[tes73.py](#)[tes74.py](#)

D: > sam > python > tes74.py > ...


```
1  #ass-7 lambda 3rd question
2  models = [
3      {'make': 'Nokia', 'model': 216, 'color': 'Black'},
4      {'make': 'Mi Max', 'model': '2', 'color': 'Gold'},
5      {'make': 'Samsung', 'model': 7, 'color': 'Blue'}
6  ]
7
8  print("Original list of dictionaries are:")
9  print(models)
10 sorted_models = sorted(models, key=lambda x: x['color'])
11 print("\nSorted List of dictionaries:")
12 print(sorted_models)
13
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

Sorted List of dictionaries:

```
[{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}, {'make': 'Mi Ma
x', 'model': '2', 'color': 'Gold'}]
```

 tes68.py tes69.py tes70.py tes71.py tes72.py tes73.py tes74.py

D: > sam > python >  tes75.py > ...

```
1  #ass-7 lambda 4th question
2  starts_with = lambda x: True if x.startswith('S') else False
3  print(starts_with('Script'))
4  print([starts_with('Poem')])
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes75.py
True
False
```

[tes69.py](#)[tes70.py](#)[tes71.py](#)[tes72.py](#)[tes73.py](#)[tes74.py](#)[tes75.py](#)

D: > sam > python > [tes76.py](#) > ...

```
1  #ass-7 lambda 5th question
2  is_num = lambda q: q.replace('.', '', 1).isdigit()
3  print(is_num('26587'))
4  print(is_num('4.2365'))
5  print(is_num('abc'))
6  print("\nPrint checked numbers:")
7
8  is_num1 = lambda r: is_num(r[1:]) if r[0] == '-' else is_num(r)
9
10 print(is_num1('-16.4'))
11 print(is_num1('-24587.11')) |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
False
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes76.py
True
True
False

Print checked numbers:
True
True
```

tes70.py

tes71.py

tes72.py

tes73.py

tes74.py

tes75.py

tes76.py

D: > sam > python > tes77.py > ...

```
1 #ass-7 lambda 6th question
2 nums = [19, 65, 57, 39, 152, 639, 121, 44, 90, 190]
3 print("Original list:")
4 print(nums)
5 result = list(filter(lambda x: (x % 19 == 0 or x % 13 == 0), nums))
6 print("\nNumbers of the above list which are divisible by nineteen or thirteen:")
7 print(result)
8 |
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes77.py

Original list:

[19, 65, 57, 39, 152, 639, 121, 44, 90, 190]

Numbers of the above list which are divisible by nineteen or thirteen:

[19, 65, 57, 39, 152, 190]



[tes71.py](#)[tes72.py](#)[tes73.py](#)[tes74.py](#)[tes75.py](#)[tes76.py](#)[tes77.py](#)

D: > sam > python > [tes78.py](#) > ...

```
1  #ass-7 lambda 7th question
2  def sort_matrix(M):
3      result = sorted(M, key=lambda matrix_row: sum(matrix_row))
4      return result
5  matrix1 = [[1, 2, 3], [2, 4, 5], [1, 1, 1]]
6  matrix2 = [[1, 2, 3], [-2, 4, -5], [1, -1, 1]]
7
8  print("Original Matrix:")
9  print(matrix1)
10 print("\nSort the said matrix in ascending order according to the sum of its rows")
11 print(sort_matrix(matrix1))
12 print("\nOriginal Matrix:")
13 print(matrix2)
14 print("\nSort the said matrix in ascending order according to the sum of its rows")
15 print(sort_matrix(matrix2))
16
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

Original Matrix:

```
[[1, 2, 3], [2, 4, 5], [1, 1, 1]]
```

Sort the said matrix in ascending order according to the sum of its rows

```
[[1, 1, 1], [1, 2, 3], [2, 4, 5]]
```

Original Matrix:

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

Sort the said matrix in ascending order according to the sum of its rows

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

tes72.py tes73.py tes74.py tes75.py tes76.py tes77.py tes78.py tes79.py

D: > sam > python > tes79.py > ...

```
1  #ass-7 lambda 8th question
2  def check_string(str1):
3      messg = [
4          lambda str1: any(x.isupper() for x in str1) or 'String must have 1 upper case character.',
5          lambda str1: any(x.islower() for x in str1) or 'String must have 1 lower case character.',
6          lambda str1: any(x.isdigit() for x in str1) or 'String must have 1 number.',
7          lambda str1: len(str1) >= 7 or 'String length should be at least 8.',
8      ]
9      result = [x for x in [i(str1) for i in messg] if x != True]
10     if not result:
11         result.append('Valid string.')
12     return result
13 s = input("Input the string: ")
14 print(check_string(s))
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes79.py
Input the string: PaceWisdom
['Valid string.']
```

[tes73.py](#)[tes74.py](#)[tes75.py](#)[tes76.py](#)[tes77.py](#)[tes78.py](#)[tes79.py](#)D: > sam > python > [tes80.py](#) > ...

```
1  #ass-7 lambda 9th question
2  def find_substring(str1, sub_str):
3      result = list(filter(lambda x: sub_str in x, str1))
4      return result
5  colors = ["red", "black", "white", "green", "orange"]
6  print("Original list:")
7  print(colors)
8  sub_str = "ack"
9  print("\nSubstring to search:")
10 print(sub_str)
11 print("Elements of the said list that contain the specific substring:")
12 print(find_substring(colors, sub_str))
13 sub_str = "abc"
14 print("\nSubstring to search:")
15 print(sub_str)
16 print("Elements of the said list that contain the specific substring:")
17 print(find_substring(colors, sub_str))
18
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

Original list:  
['red', 'black', 'white', 'green', 'orange']

Substring to search:  
ack  
Elements of the said list that contain the specific substring:  
['black']

Substring to search:

tes62.py



tes63.py

tes64.py

tes65.py

tes66.py

tes67.py

tes68.py

D: &gt; sam &gt; python &gt; tes62.py &gt; ...

```
1  #ass-7 1st question
2  class RomanNumerals:
3      roman_to_int_map = {
4          'I': 1, 'IV': 4, 'V': 5, 'IX': 9, 'X': 10,
5          'XL': 40, 'L': 50, 'XC': 90, 'C': 100,
6          'CD': 400, 'D': 500, 'CM': 900, 'M': 1000
7      }
8      int_to_roman_map = [
9          (1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'),
10         (100, 'C'), (90, 'XC'), (50, 'L'), (40, 'XL'),
11         (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'), (1, 'I')
12     ]
13
14     @staticmethod
15     def to_roman(num):
16         roman = ''
17         for value, numeral in RomanNumerals.int_to_roman_map:
18             while num >= value:
19                 roman += numeral
20                 num -= value
21         return roman
22
23     @staticmethod
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

LIV

104

[tes74.py](#)[tes75.py](#)[tes76.py](#)[tes77.py](#)[tes78.py](#)[tes79.py](#)[tes80.py](#)[tes81.py](#)

D: > sam > python > [tes81.py](#) > ...

```
1  #ass-7 lambda 10th question
2  def sort_mixed_list(mixed_list):
3      mixed_list.sort(key=lambda e: (isinstance(e, str), e))
4      return mixed_list
5  mixed_list = [19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]
6  print("Original list:")
7  print(mixed_list)
8  print("\nSort the said mixed list of integers and strings:")
9  print(sort_mixed_list(mixed_list))
10
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

PS C:\Users\hp> & C:/Users/hp/AppData/Local/Programs/Python/Python312/python.exe d:/sam/python/tes81.py

Original list:

[19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]

Sort the said mixed list of integers and strings:

[1, 10, 12, 19, 'blue', 'green', 'green', 'red', 'white']

[tes73.py](#)[tes74.py](#)[tes75.py](#)[tes76.py](#)[tes77.py](#)[tes78.py](#)[tes79.py](#)[tes80.py](#)

D: > sam > python > [tes80.py](#) > ...

```
1  #ass-7 lambda 9th question
2  def find_substring(str1, sub_str):
3      result = list(filter(lambda x: sub_str in x, str1))
4      return result
5  colors = ["red", "black", "white", "green", "orange"]
6  print("Original list:")
7  print(colors)
8  sub_str = "ack"
9  print("\nSubstring to search:")
10 print(sub_str)
11 print("Elements of the said list that contain the specific substring:")
12 print(find_substring(colors, sub_str))
13 sub_str = "abc"
14 print("\nSubstring to search:")
15 print(sub_str)
16 print("Elements of the said list that contain the specific substring:")
17 print(find_substring(colors, sub_str))
18
```

[PROBLEMS](#)[OUTPUT](#)[DEBUG CONSOLE](#)[TERMINAL](#)[PORTS](#)

Substring to search:

ack

Elements of the said list that contain the specific substring:

['black']

Substring to search:

abc

Elements of the said list that contain the specific substring:

[]