



COMP 1023 Introduction to Python Programming
Exercises on Collections
COMP 1023 Teaching Team

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China



Exercise 1

Write a program that reads a list of scores and then assigns grades based on the following scheme:

- The grade is A if score is $\geq$ best - 10.
- The grade is B if score is $\geq$ best - 20.
- The grade is C if score is $\geq$ best - 30.
- The grade is D if score is $\geq$ best - 40.

Enter scores separated by spaces from one line: 0 55 70 58

Student 0 score is 40.0 and grade is C

Student 1 score is 55.0 and grade is B

Student 2 score is 70.0 and grade is A

Student 3 score is 58.0 and grade is B

Exercise 1 Solutions

```
input_scores = input("Enter scores separated by spaces from one line: ")
score_strings = input_scores.split()
scores = []; best = 0

for score_str in score_strings:
    score = float(score_str)
    scores.append(score)
    if score > best:
        best = score

i = 0
for score in scores:
    if score >= best - 10:
        grade = 'A'
    elif score >= best - 20:
        grade = 'B'
    elif score >= best - 30:
        grade = 'C'
    elif score >= best - 40:
        grade = 'D'
    else:
        grade = 'F'

    print("Student", i, "score is", score, "and grade is", grade)
    i += 1
```

Exercise 2

Write a program that reads a list of scores and then assigns pass if the score is equal or above the average, other assigns fail.

Enter scores separated by spaces from one line: 67 87 56 76 81 83 54 79

Student 0 score is 67 and result is Fail

Student 1 score is 87 and result is Pass

Student 2 score is 56 and result is Fail

Student 3 score is 76 and result is Pass

Student 4 score is 81 and result is Pass

Student 5 score is 83 and result is Pass

Student 6 score is 54 and result is Fail

Student 7 score is 79 and result is Pass

Exercise 2 Solutions

```
# Read scores from input
input_scores = input("Enter scores separated by spaces from one line: ")
score_strings = input_scores.split()

# Convert to integers using list comprehension
scores = [int(score) for score in score_strings]

# Calculate average
total = 0
for score in scores:
    total += score
average = total / len(scores)

# Assign and display pass/fail results
for i in range(len(scores)):
    score = scores[i]
    if score >= average:
        result = "Pass"
    else:
        result = "Fail"

    print(f"Student {i} score is {score} and result is {result}")
```

Exercise 3

Write a program that reads some integers between 1 and 100 in one line and counts the occurrences of each.

Enter integers between 1 and 100: 2 5 6 5 4 3 23 43 2

2 occurs 2 times

3 occurs 1 time

4 occurs 1 time

5 occurs 2 times

6 occurs 1 time

23 occurs 1 time

43 occurs 1 time

Exercise 3 Solutions

```
# Read integers from input
input_numbers = input("Enter integers between 1 and 100: ")
number_strings = input_numbers.split()

# Convert to integers using list comprehension
numbers = [int(num) for num in number_strings]

# Count occurrences
count_dict = {}
for number in numbers:
    if number in count_dict:
        count_dict[number] += 1
    else:
        count_dict[number] = 1

# Display results in sorted order
for number in sorted(count_dict.keys()):
    count = count_dict[number]
    if count == 1:
        print(f"{number} occurs 1 time")
    else:
        print(f"{number} occurs {count} times")
```

Exercise 4

Write a program that reads in integers separated by a space in one line and displays distinct numbers in their input order and separated by exactly one space (i.e., if a number appears multiple times, it is displayed only once.)
Hint: Read all the numbers and store them in `list1`. Create a new list `list2`. Add a number in `list1` to `list2`. If the number is already in the list, ignore it.

Enter 10 numbers: 1 2 3 2 1 6 3 4 5 2

The number of distinct numbers is 6

The distinct numbers are: 1 2 3 6 4 5

Exercise 4 Solutions

```
# Read integers from input
input_numbers = input("Enter 10 numbers: ")
number_strings = input_numbers.split()

# Convert to integers
list1 = [int(num) for num in number_strings]

# Create list for distinct numbers
list2 = []

# Add numbers to list2 if not already present
for number in list1:
    if number not in list2:
        list2.append(number)

# Display results
print(f"The number of distinct numbers is {len(list2)}")
print("The distinct numbers are:", end=" ")
for i in range(len(list2)):
    print(list2[i], end="")
    if i < len(list2) - 1:
        print(" ", end="")
print()
```

Exercise 5

Write a program that generates 1,000 random integers between 0 and 9 and displays the count for each number.

Hint: Use a list of 10 integers, say `counts`, to store the counts for the number of 0s, 1s, ..., 9s.

Exercise 5 Solutions

```
import random

# Initialize counts list with 10 zeros
counts = [0] * 10

# Generate 1000 random integers and count them
for i in range(1000):
    number = random.randint(0, 9)
    counts[number] += 1

# Display the counts for each number
for i in range(10):
    print(f"Count for {i}: {counts[i]}")
```

Exercise 6

Write a function that returns the index of the largest element in a list of integers. If the number of such elements is greater than 1, return the largest index. Use the following header:

```
def index_of_largest_element(my_list):
```

Write a test program that prompts the user to enter a list of numbers, invokes this function to return the index of the largest element, and displays the index.

```
Enter 10 numbers: 1 3 5 7 5 3 2
```

```
The index of the largest element is 3
```

Exercise 6 Solutions

```
def index_of_largest_element(my_list):
    max_index = 0
    for i in range(1, len(my_list)):
        if my_list[i] >= my_list[max_index]:
            max_index = i
    return max_index

def main():
    # Read numbers from input
    input_numbers = input("Enter numbers: ")
    number_strings = input_numbers.split()

    # Convert to integers
    numbers = [int(num) for num in number_strings]

    # Find and display the index of largest element
    largest_index = index_of_largest_element(numbers)
    print(f"The index of the largest element is {largest_index}")

if __name__ == "__main__":
    main()
```

Exercise 7

Write a function that returns the Least Common Multiple (LCM) of integers in a list. Use the following header:

```
def lcm(numbers):
```

Write a test program that prompts the user to enter a list of numbers, invokes this function to find the LCM of these numbers, and displays the LCM.

Enter 10 numbers: 43 65 34

The LCM of 43, 65, 34 is 95030

Exercise 7 Solutions

```
def gcd(a, b):  
    """Helper function to find Greatest Common Divisor using Euclidean algorithm"""  
    while b != 0:  
        a, b = b, a % b  
    return a  
  
def lcm(a, b):  
    """Helper function to find LCM of two numbers"""  
    return abs(a * b) // gcd(a, b)  
  
def lcm(numbers):  
    """Function to find LCM of a list of numbers"""  
    if not numbers:  
        return 0  
  
    result = numbers[0]  
    for i in range(1, len(numbers)):  
        result = lcm(result, numbers[i])  
  
    return result
```

Exercise 7 Solutions (Cont'd)

```
def main():  
    # Read numbers from input  
    input_numbers = input("Enter numbers: ")  
    number_strings = input_numbers.split()  
  
    # Convert to integers  
    numbers = [int(num) for num in number_strings]  
  
    # Find and display the LCM  
    lcm_result = lcm(numbers)  
    print(f"The LCM of {numbers} is {lcm_result}")  
  
if __name__ == "__main__":  
    main()
```

Exercise 8

Write a function that returns a new list by swapping the smallest element with the largest element in the list. Use the following function header:

```
def smallest_largest_swap(my_list):
```

Write a test program that reads in a list of integers and display the result.

Enter 10 numbers: 65 78 23 58 85 71 94 47 68 39

List after swapping smallest and largest: 65, 78, 94, 58, 85, 71, 23, 47, 68, 39

Exercise 8 Solutions

```
def smallest_largest_swap(my_list):  
    if len(my_list) == 0:  
        return my_list  
  
    # Find index of smallest and largest elements  
    min_index = 0  
    max_index = 0  
  
    for i in range(1, len(my_list)):  
        if my_list[i] < my_list[min_index]:  
            min_index = i  
        if my_list[i] > my_list[max_index]:  
            max_index = i  
  
    # Create a new list and swap the elements  
    new_list = my_list.copy()  
    new_list[min_index], new_list[max_index] = new_list[max_index], new_list[min_index]  
  
    return new_list
```

Exercise 8 Solutions (Cont'd)

```
def main():  
    # Read numbers from input  
    input_numbers = input("Enter numbers: ")  
    number_strings = input_numbers.split()  
  
    # Convert to integers  
    numbers = [int(num) for num in number_strings]  
  
    # Swap smallest and largest  
    result = smallest_largest_swap(numbers)  
    print(f"List after swapping smallest and largest: {result}")  
  
if __name__ == "__main__":  
    main()
```

Exercise 9

Write the following function that returns true if all the elements in the list are even, otherwise returns false.

```
def isEven(lst):
```

Write a test program that prompts the user user to enter a list and displays whether the list is even or not.

```
Enter numbers: 64 78 4 70 54
```

```
The list is even
```

```
Enter numbers: 52 76 91 44 806 22
```

```
The list is not even
```

Exercise 9 Solutions

```
def isEven(lst):  
    for number in lst:  
        if number % 2 != 0:  
            return False  
    return True  
  
def main():  
    # Read numbers from input  
    input_numbers = input("Enter numbers: ")  
    number_strings = input_numbers.split()  
  
    # Convert to integers  
    numbers = [int(num) for num in number_strings]  
  
    # Check if all elements are even  
    if isEven(numbers):  
        print("The list is even")  
    else:  
        print("The list is not even")  
  
if __name__ == "__main__":  
    main()
```

Exercise 10

Write a function that returns the maximum of all the elements in a specified column in a matrix using the following header:

```
def maxColumn(m, columnIndex):
```

Write a test program that reads 3×4 matrix and displays the maximum of each column.

```
Enter a 3-by-4 matrix row 0: 12 63 59 82  
Enter a 3-by-4 matrix row 1: 75 65 92 52  
Enter a 3-by-4 matrix row 2: 41 73 63 88  
Maximum of the elements at column 0 is 75.0  
Maximum of the elements at column 1 is 73.0  
Maximum of the elements at column 2 is 92.0  
Maximum of the elements at column 3 is 88.0
```

Exercise 10 Solutions

```
def maxColumn(m, columnIndex):
    max_value = m[0][columnIndex]
    for i in range(1, len(m)):
        if m[i][columnIndex] > max_value:
            max_value = m[i][columnIndex]
    return max_value

def main():
    rows, cols = 3, 4
    matrix = []

    # Read 3x4 matrix from user
    for i in range(rows):
        row_input = input(f"Enter a 3-by-4 matrix row {i}: ")
        row_numbers = [float(num) for num in row_input.split()]
        matrix.append(row_numbers)

    # Find and display maximum of each column
    for j in range(cols):
        max_val = maxColumn(matrix, j)
        print(f"Maximum of the elements at column {j} is {max_val}")

if __name__ == "__main__": main()
```

Exercise 11

Write a function that finds average of the numbers of the major diagonal in a $n \times n$ matrix of integers using the following header:

```
def avgMajorDiagonal(m):
```

The main diagonal is the diagonal that runs from the top left corner to the bottom right corner in the square matrix. Write a test program that reads a 4×4 matrix and displays the average of elements at the major diagonal.

```
Enter a 4-by-4 matrix row 0: 7 9 5 2
```

```
Enter a 4-by-4 matrix row 1: 8 3 7 9
```

```
Enter a 4-by-4 matrix row 2: 2 6 1 7
```

```
Enter a 4-by-4 matrix row 3: 3 9 7 5
```

```
Average of the elements in the major diagonal is 4.0
```

Exercise 11 Solutions

```
def avgMajorDiagonal(m):
    total = 0
    n = len(m)
    for i in range(n):
        total += m[i][i]
    return total / n

def main():
    n = 4
    matrix = []

    # Read 4x4 matrix from user
    for i in range(n):
        row_input = input(f"Enter a 4-by-4 matrix row {i}: ")
        row_numbers = [int(num) for num in row_input.split()]
        matrix.append(row_numbers)

    # Calculate and display average of major diagonal
    average = avgMajorDiagonal(matrix)
    print(f"Average of the elements in the major diagonal is {average}")

if __name__ == "__main__":
    main()
```

Exercise 12

Write the following function that returns the location of the smallest element in a two-dimensional list:

```
def locateSmallest(lst):
```

The return value is a one-dimensional list that contains two elements. These two elements indicate the row and column indexes of the smallest elements in the two-dimensional list. Write a test program that prompts the user to enter a two-dimensional list and displays the location of the smallest element in the list.

```
Enter the number of rows in the list: 3
```

```
Enter a row: 3 5 8
```

```
Enter a row: 5 8 2
```

```
Enter a row: 9 6 3
```

```
The location of the smallest element is at (1, 2)
```

Exercise 12 Solutions

```
def locateSmallest(lst):
    min_value = lst[0][0]
    min_row, min_col = 0, 0

    for i in range(len(lst)):
        for j in range(len(lst[i])):
            if lst[i][j] < min_value:
                min_value = lst[i][j]
                min_row, min_col = i, j
    return [min_row, min_col]

def main():
    rows = int(input("Enter the number of rows in the list: "))
    matrix = []
    for i in range(rows):
        row_input = input("Enter a row: ")
        row_numbers = [int(num) for num in row_input.split()]
        matrix.append(row_numbers)

    location = locateSmallest(matrix)
    print(f"The location of the smallest element is at ({location[0]}, {location[1]})")

if __name__ == "__main__": main()
```