

Week 6

Write and run your programs with IDLE editor. Submit finished programs to CodeGrade. Note that some tasks have several steps (A, B, C, ...) in CodeGrade.

IMPORTANT: End each input-command string with a newline symbol `\n`. For example:

```
variable = input("Some text:\n")
```

Task 1: Implement a function that takes a list of integers and returns the same list with all duplicate elements removed, preserving the **original** order.

In the main program ask for integers using the function `input_integers()` given in the lecture.

Example run 1:

```
Give integers separated by comma:
3,2,3,1,6,2,3,4,5
Original List: [3, 2, 3, 1, 6, 2, 3, 4, 5]
List with duplicates removed: [3, 2, 1, 6, 4, 5]
```

Task 2: Create a function that takes a list of integers and returns the kth smallest element in the list. You are not allowed to use `sorted()` function or `.sort()` method.

In the main program ask for integers using the function `input_integers()` given in the lecture. After that ask the integer k. First verify that it is between 1 and then length of the array according to the example below.

Example run 1:

```
Give integers separated by comma:
6,3,5,2,11,4
Give an integer:
3
3th smallest element is 4
```

Example run 2:

```
Give integers separated by comma:
6,3,5,2,11,4
Give an integer:
9
Not suitable
```

Task 3: Write a function that reverses a list of integers, without using the `reverse()` method or slicing.

In the main program ask for integers using the function `input_integers()` given in the lecture.

Example run 1:

```
Give integers separated by comma:
1,2,3,4,5
Reversed list: [5, 4, 3, 2, 1]
```

Task 4: Write a function `create_matrix(rows, cols)` that takes the number of rows and columns as input and prompts the user to enter the elements row by row separated by whitespaces. Validates the number of elements in each row. If the number of elements is not correct, ask the elements again. The function **returns** the resulting matrix.

Note that when you use the `split()` method in Python without providing any parameter, it will split the string using whitespace characters (spaces, tabs, and newlines) as the default delimiter.

After that write a function `print_matrix(matrix)` that prints `matrix` row by row. Separate element by using tabulator. See example run below. Note that print does not number of rows and columns as parameters, but the method can print any matrix.

Ask the number of rows and columns in the main program.

Example run 1:

```
Enter the number of rows:
2
Enter the number of columns:
3
Give row 1:
4 5
Error: Invalid number of elements in the row. Please try again.
Give row 1:
1 2 3
Give row 2:
4 5 6
|1    2    3|
|4    5    6|
```

Task 5: For a given matrix M with dimensions $m \times n$ (m rows and n columns), its transpose, denoted as M^T will have dimensions $n \times m$ (n rows and m columns). The element at row i and column j in the original matrix becomes the element at row j and column i in the transposed matrix.

Write a function `transpose(matrix)` which takes a matrix as input and **returns** its transpose. Hint: Generate first a matrix of correct size by filling it by zeros. Then add the correct elements.

Extend your solution to Task 4 so that after the original matrix is printed, the `transpose()` function is called. After that the transpose is printed. See example below.

Example run 1:

```
Enter the number of rows:
3
Enter the number of columns:
2
Give row 1:
2 4
Give row 2:
3 6
Give row 3:
4 8
The original matrix:
|2  4|
|3  6|
|4  8|
Its transpose:
|2  3  4|
|4  6  8|
```