Mock Test2

IN CAPITAL LETTERS, enter your name, surname and student ID (6-digit index/album number):

…….

READ THIS CAREFULLY

1. Save your programs in files with names given in round brackets at the beginning of each task (p1.py, p2.py, p3.py, …).
2. Do not use print() or input() in your programs (inside defined functions).
3. To test the defined function, do it in the statement below. Do not remove the statement from the file.

   ```
   if __name__ == "__main__":
       print( f(…) )
       print( f(…) )
   ```

4. **To find errors in your program, use a debugger.**
5. Upload each program file separately to moodle.uek.krakow.pl.
6. Upload each program file immediately after completing the task. Don't wait until the end of the test!

(p1.py) The playing cards have the following values: Ace (A), King (K), Queen (Q), Jack (J) and 10 (T) have a value of 10 each. The other cards have the value indicated by the card number. Define a function f(player1,player2) that returns true if the first player has cards of the same or higher value, and false otherwise. Example:

```
f("AJ972","AQT72") → False
f("9532","K8") → True
```

(p2.py) An array contains at least 3 integers. All numbers in the array are equal except one. Define a function f(arr) that returns a number in the arr array that is different from the other numbers. Example:

```
f([7,7,7,7,7,5,7,7]) → 5
```

(p3.py) A two-dimensional array contains the same number of rows and columns. Define a function f(array2D) that, for the given two-dimensional array array2D, returns True when the sum of the values in each row of the array is equal to the sum of the values in the corresponding column (e.g., the sum of the values in row 3 is equal to the sum of the values in column 3) , and False otherwise. Example:

```
f([[3,7,2],[4,2,5],[5,2,1]]) → True
f([[3,7,2],[4,2,5],[9,2,1]]) → False
```

(p4.py) The dictionary contains the names of subjects and the grades obtained. Define a function f(subjects) that, for the given subjects and their grades, returns the name of the subject for which the average grade is the highest. Example:

```
f({"math":[3,4,4],"geo":[5,4,4,4],"comp":[5,4]}) → "comp"
```

(p5.py) Define a function f(first_letter,last_letter) that, for the data.txt file, returns the number of words that start with the first_letter and end with the last_letter. Example:

```
f("w","d") → compare your result with other students
```

(p6.py) Define a function f(years, course, average_grade) that, for the data.json file, returns the number of students who are at least the given number of years and have a grade average of at least average_grade in the given course name. Example:

```
f(21, "statistics", 4) → compare your result with other students
```

(p7.py) A valid username consists of 4 to 12 characters: lowercase letters, numbers and the underscore character. Define a function f(array) that, for a given array of usernames, returns the number of valid usernames in the array. Example:

```
f(["uek","water_7_x","anna.may","a_b_c_d_e_f"]) → 2
```

(p8.py) Reverse Polish Notation (RPN) is a mathematical notation in which operators follow their operands, e.g. 2 3 + 4 *. Define a function f(expression) that, for an RPN expression, returns the value of that expression. The expression contains only non-negative integers and the + and - operators. Example:

```
f("2 3 +") → 5
f("2 6 + 4 5 - +") → 7
f("11 7 + 15 - 14 +") → 17
```

(p9.py) The file data.csv contains a list of employees of the company ABC-Data. Define a function f(value) that returns the number of employees whose salary is greater than or equal to the given value. Example:

```
f(9200) → compare your result with other students
f(11640) → compare your result with other students
```

(p10.py) A two-dimensional array contains different integer numbers. Define a function f(array) that returns True if the row and column numbers for the smallest value in the array are equal, and False otherwise. Example:

```
f([[7,8],[5,3],[9,4]]) → True     # 3, row 1, col 1
f([[7,8,5,3],[9,4,2,6]]) → False  # 2, row 1, col 2
```