0. Select a group

In Moodle, please register to the group **with the same label** as is written on your card. Make sure that you and your partner are in the same group.

1. Name of the month

Show and explain your solution of month name homework exercise (*home1.py*) to your partner. Discuss and write below: what are the biggest differences in your programs?

Can you think of yet another way of how this exercise can be solved? What is it?

2. Food

Compare your food label program (*home2.py*) with that of your partner. Find one detail or idea in either of the solutions that is definitely worth using in your future programs.

Based on both programs, create a common program that solves the same problem, by combining the best ideas from both partner’s solutions. Try to arrive to a version that covers all essential aspects that need to be taken care of in the situation described in the exercise. Save this program to a separate file.

Food recipes often use measures like teaspoon, tablespoon, and cup. Reasonable equivalents are, for example, 1 teaspoon = 5 grams, 1 tablespoon = 15 grams, and 1 cup = 220 grams. Write a program that allows to convert between each pair of these units. For example, it lets you calculate how many grams are 3.5 teaspoons or how many tablespoons are 2 cups.

Write below: in what respects is this program more elaborate than the program you saved to file?
3. Grade

Next we try “proper” pair programming. Please sit behind one computer so that both partners can see the screen.

- One of you will be the driver and his/her task will be to actually write the code. The driver takes care of all details and technicalities.
- The other will be the observer/navigator who reviews each line. Additionally, he/she keeps in mind broader strategic direction of the work, comes up with ideas for improvement and tries to foresee likely future problems.

In courses there are two common types of grading: differentiated (numerical grades) or non-differentiated (pass/fail). Write a program that asks the user to enter the final score between 0 and 100 and the type of grading. For non-differentiated grading, if the score is larger than 50, then the program prints a message saying that the user has passed the course; otherwise the course is failed. For differentiated grading, the program prints the grade according to the following schema: >=91 is A, >=81 is B, >=71 is C, >=61 is D, >=51 is E, and <51 is F.

Develop your program to the point where, if the user enters something inappropriate, the program doesn't crash (use try-except or if-else).

Enter score: 95
Is the grading differentiated on non-differentiated grading (d/n)? n
PASSED

Enter score: -15
Is the grading differentiated on non-differentiated grading (d/n)? d
Score must be in the range 0-100

Enter score: 65
Is the grading differentiated on non-differentiated grading (d/n)? d
D

Enter score: 50
Is the grading differentiated on non-differentiated grading (d/n)? don’t know
Grading type not recognized

4. Number in words

Now please switch the roles: the person who was the driver will now be observer/navigator and vice versa. Otherwise the responsibilities of both roles are the same.

Pick a language that you know reasonably well (or are able to learn in 15 minutes): English, Estonian, etc.

Write a program that prompts the user for an integer between 0 and 999, and prints it on the screen in words.

(Example on the next page.)
Example
Language: Dutch

Enter the number: 16
zestien
Enter the number: 530
vijfhonderd dertig
Enter the number: 127
honderd zeven en twintig

Write below: what language you picked?

What do you think: how difficult would it be to adapt your program to another language? Of course it depends on the language but can you formulate some general statement(s)?

What are your preliminary conclusions of this type of programming? Did you solve exercises more quickly? Are resulting programs better in some respect than if you were to write them alone?

5. Poker hand ranking

In poker, a hand is a set of five ordinary playing cards. Each hand has a rank, which determines the strength of the hand. Ranks are used to compare different hands with each other.

Hand ranking in order of strength from the strongest to the weakest is as follows (https://www.cardplayer.com/rules-of-poker/hand-rankings).
1. Royal flush: A, K, Q, J, 10, all of the same suit
2. Straight flush: five cards in a sequence, all of the same suit
3. Four of a kind: four cards of the same rank
4. Full house: three cards of the same rank plus two cards of the same but different rank
5. Flush: five cards of the same suit but not in a sequence
6. Straight: five cards in a sequence but not of the same suit
7. Three of a kind: three cards of the same rank
8. Two pair: two cards of the same rank plus two cards of the same but different rank
9. Pair: two cards of the same rank
10. High card: if nothing above, then the highest card plays

Individual card ranks are A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2. Suits are clubs, diamonds, hearts, spades.
Write a program that gets a hand from the user and prints out the name of the highest ranking that this hand belongs to.

Enter 1. card: 7 C  
Enter 2. card: 8 H  
Enter 3. card: 8 D  
Enter 4. card: 8 S  
Enter 5. card: 9 D  
Three of a kind

Simpler version. You can assume that the hand is entered in order of card strengths.

Harder version. The hand can be entered in any order.

At the end of session

Please submit your programs in Moodle as a group. It is enough that only one partner submits the work.