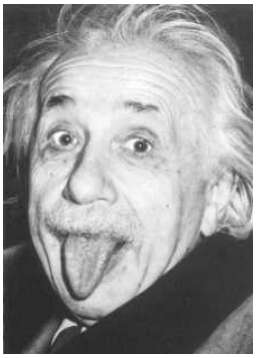
Intro Ex1 – Einstein



Submission deadline
Thursday, 24 October, 20:55

Objectives

The purpose of this exercise is to become familiar with basic programming in python including receiving input from the user and run simple arithmetic expressions. This exercise is not long, but do not wait with it until the last minute. Your code must obey the Coding Style Guidelines.

Background:
Welcome to the Hebrew University. As you may know, one of the founders of HUJI was Albert Einstein. It is said that Einstein used to take great delight in baffling friends with the puzzle below.

First, write the number 1089 on a piece of paper, fold it, and hand it to a friend for safekeeping. What you wrote down is not to be read until you have completed your amazing mental feat.
Next, ask your friend to write down any three-digit number, emphasizing that the first and last digits must differ by at least one. Make sure leading zeros are considered. Close your eyes or turn your back while this is being done. Better still, have someone blindfold you.

After your friend has written down the three-digit number, ask him to reverse it, and then subtract the smaller from the larger.
Example: 773 - 377 = 396.
Once this is done, tell your friend to reverse the new number.
Example: 396 becomes 693.

Next ask your friend to add the new number and its reverse together.
Example: 396 + 693= 1089.
If all goes as planned, your friend will be amazed. The number you wrote down at the start -- 1089 -- will always be the same as the end result of this mathematical trick.

Specifications:
Your program will play the Einstein game as follows:

1. Print a welcome message
2. Prompt the user for a 3 digit number as described
3. Print out the number the user choose and the reverse of that number
4. Print out the difference between the entered number and the reversed number. This should always be a positive number.
5. Print the reverse of the difference
6. Print the sum of the difference and the reversed difference. It should be 1089 for any 3 digit

number which follows the required conditions.

7. **Your program must behave exactly as the school solution. See more details below.**

Program notes

1. Open a new file in the IDLE - call it: ex1.py
   Save it under ~/safe/intro2cs/ex1/
   don't forget to create the directory ex1 before trying to save the file
2. **print** is a Python function that will print on the output window any combination of variables, values and strings. Each item to be printed must be separated from other items by a comma. All the items will be printed together, followed by a new line. For example:

```
input_number = 321
reversed_number  = 123
print("For the number:", input_number, "the reverse number is:",
reversed_number)
```

   Try it. Open the Python Shell window and print this code and see what you get.

3. **Prompting the user for input** is being done in python using another function named input().The input() function takes a string, a sequence of characters between quotes, as a prompt to print to the user. It then waits until the user types a response, terminated by the user typing the Enter key. A string, again as a sequence of characters, is returned.

```
num1_str = input("Give me a number:")
```

   You may assume that the input from the user is legal – i.e. a three digit number string

4. **String to Integer -** In order to do arithmetic use of the three digits number, the string input must be converted from a string to representation of a number in python.
   You may do it using the int function:

```
num1_int = int(num1_str)
```

   If you are interested in doing the opposite operation (int to str) run

```
num1_str= str(num1_int)
```

5. **Reverse the digits order**. After receiving a number and changing its type to an int, we want to reverse its order. How can we do it with math? Take a pen and a paper and try to think about it.

   IMPORTANT - Division in Python: For this arithmetic operation you may need to use division, we differentiate between three different operations in python
   / - "Regular" division - 10/3 = 3.3333333333333335

// - Division without the remainder- 10//3 = 3
% - The modulo operation - return only the remaining of the division 10%3 = 1

Play with it - Type different values and different operators in the Shell.

Note that we are working with a new version of python - 3.3, comparing with the old 2.7. Division operators are only few of the differences between the versions. For those of you who were familiar with python 2.7 read this article to learn what is new: http://docs.python.org/3/whatsnew/3.0.html

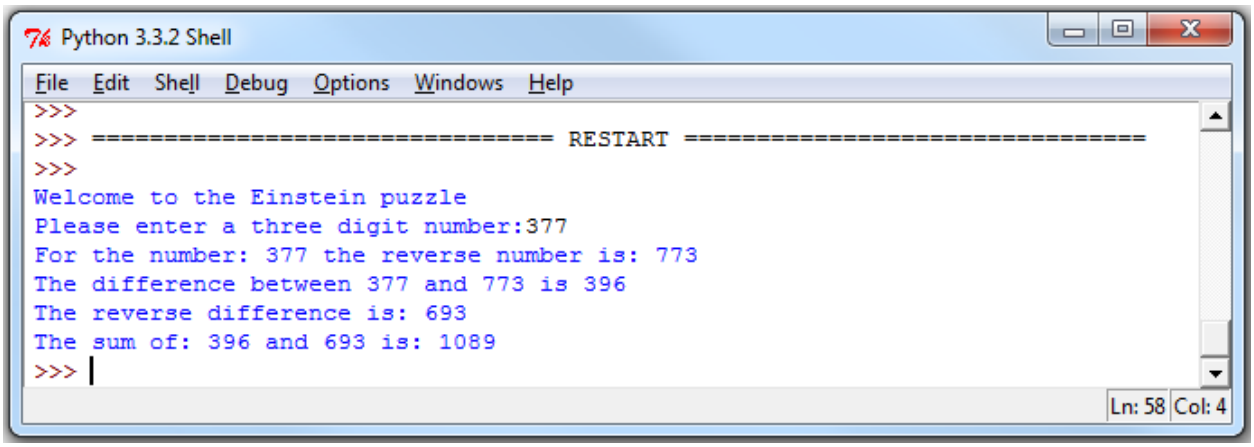
6. Other arithmetic calculation: Once the numbers are represented as the type int – it's easy to do mathematic with it. For example type in the shell:

```
a = 4
b = 5
c = a + b
d = a - b
```

The specification mentions that we should always subtract the smaller of the two numbers (the user number or its reverse) from the larger. Another way to do this is to take the absolute value of any subtraction. The absolute value function is called abs in Python.
Thus `abs(-27)` yields 27. `abs(45)` yields 45 and
`abs(a - b) = abs(b - a)`.

7. **IMPORTANT: Your program output should be exactly as the school solution output: Do not change the messages to the users, do not add extra white spaces or lines, do not ignore capital letters.**
To test how the school solution behaves on legal input run in your linux shell:
`python3 ~intro2cs/bin/ex1/ex1`

If the user choose to enter the number 377 it must look as in the following picture:



8. Test your program with the automatic testers, to ensure your output is correct. more details about automatic testers can be found online in the course website.

Submission and Further Guidelines:

Creating a tar file

- By now, you should have created the following files:
    1. ex1.py
    2. README (as explained in the course guidelines)
    3. Create a TAR file named ex1.tar containing only the above files by invoking the shell command:
    4. **tar cvf ex1.tar ex1.py README** (See tirgul 1 slides for details about tars)
- The TAR file should contain only these files!
    1. It is recommended to check your TAR file by copying it to a different directory, opening it by executing the command: **tar xvf ex1.tar** and verifying that ex1.py and README were indeed successfully created.

Submitting the tar file

- You should submit the file ex1.tar via the "Upload File" link on the course home page, under the ex1 link.
- Note that submission requires you to be registered as a student and you need to be logged in.
- Shortly after you upload the file you should receive an email to your university mailbox. That email contains the test results of your submission and the pdf file that was passed to the grader just like in ex0.

Running the tests yourself

- There are two ways you can run these testers and see the results without submitting the exercise and getting the pdf.
- The first - place your tar in an empty directory, go to that directory using the shell and type in: **`~intro2cs/bin/testers/ex1 ex1.tar`**
- The second - download a file (ex1testing.tar) from here. Extract the contents (**`tar -xjf ex1testing.tar.bz2)`** of the file and follow the instructions in the file called "TESTING".

School Solution:

- To test how the school solution behaves on legal input run in your linux shell: `python3 ~intro2cs/bin/ex1/ex1`
- You may assume in this exercise that the input from the user is legal – i.e. a three digit number string, where the first and the last digits differ by at least one.