The filename for your homework must be HW2_Python.

1. Define a function ‘fib’ that takes a number, ‘n’, as a parameter and prints all the Fibonacci numbers less than ‘n’ to the screen.

Exercises compiled by Torbjörn Lager.

2. Define a function mymax() that takes two numbers as arguments and returns the largest of them. Use the if-then-else construct available in Python. (It is true that Python has the max() function built in, but writing it yourself is nevertheless a good exercise.)

3. Define a function max_of_three() that takes three numbers as arguments and returns the largest of them.

4. Define a function mylen() that computes the length of a given list or string. (It is true that Python has the len() function built in, but writing it yourself is nevertheless a good exercise.)

5. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.

6. Write a function translate() that will translate a text into "rövarspråket" (Swedish for "robber's language"). That is, double every consonant and place an occurrence of "o" in between. For example, translate("this is fun") should return the string "tothohisos isos fofunon".

7. Define a function sum() and a function multiply() that sums and multiplies (respectively) all the numbers in a list of numbers. For example, sum([1, 2, 3, 4]) should return 10, and multiply([1, 2, 3, 4]) should return 24.

8. Define a function reverse() that computes the reversal of a string. For example, reverse("I am testing") should return the string "gnitset ma I".

9. Define a function is_palindrome() that recognizes palindromes (i.e. words that look the same written backwards). For example, is_palindrome("radar") should return True.

10. Write a function is_member() that takes a value (i.e. a number, string, etc) x and a list of values a, and returns True if x is a member of a, False otherwise. (Note that this is exactly what the in operator does, but for the sake of the exercise you should pretend Python did not have this operator.)

11. Define a function overlapping() that takes two lists and returns True if they have at least one member in common, False otherwise. You may use your

is_member() function, or the in operator, but for the sake of the exercise, you should (also) write it using two nested for-loops.

12. Define a function generate_n_chars() that takes an integer n and a character c and returns a string, n characters long, consisting only of c:s. For example, generate_n_chars(5,"x") should return the string "xxxxx". (Python is unusual in that you can actually write an expression 5 * "x" that will evaluate to "xxxxx". For the sake of the exercise you should ignore that the problem can be solved in this manner.)