PYTHON PROGRAMMING SHEET – 1

1. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4,..., 1/10.

2. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero. What should your program do if the user inputs a negative number? As a programmer, you should always consider “edge conditions” like these when you program! (Another way to put it-always assume the users of your program will be trying to find a way to break it! If you don’t include a condition that catches negative numbers, what will your program do?)

3. Write a program using a for loop that calculates exponentials. Your program should ask the user for a base base and an exponent exp, and calculate $base^{exp}$.

4. Write a program using a while loop that asks the user to enter a number that is divisible by 2. Give the user a witty message if they enter something that is not divisible by 2-and make them enter a new number.

5. Define a function max() 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.)

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

7. Define a function 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.)

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

9. 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".

10. 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.

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

12. 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.

13. 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.)

14. 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.