1) Newton’s Second Law of motion is expressed in the formula *F* = *m × a* where

*F* is force,*m* is mass, and *a* is acceleration. Assume that the user knows the

mass of an object and the force on that object but wants to obtain the object’s

acceleration *a*. Write a program to Calculate the acceleration and display the

result to the user.

2) Write a program which will find all such numbers which are divisible by 7 but

are not a multiple of 5, between 2000 and 3200 (both included). The numbers

obtained should be printed in a comma separated sequence on a single line.

3) Write a function called “calc\_weight\_ on\_ planet()” which calculates your

equivalent weight on another planet .It should take two arguments: your

weight on Earth and the surface gravity of the planet .

**Note:**

23.1 m/s2 which is the approximate surface gravity of Jupiter and Earth’s

surface gravity is approximately 9.8 m/s2. weight is equal to mass times

surface gravity.

4) Write a function called num\_atoms() that calculates how many atoms are in *n*

grams of an element given its atomic weight. This function should take two

parameters: the amount of the element in grams and atomic weight of the

element

**Note**:. atomic weight of gold (Au) 196.97 with units in grams/mole.

Atomic weight of carbon=12.001

Atomic weight of hydrogen=1.008

Avogadro’s number is a constant, 6*.*022 *×* 1023

5) Write a recursive function and an iterative function to compute the Fibonacci

sequence. Compare the performance of recursive and iterative function

6) Write a program that prompts the user to enter a list of words and stores in a

list only those words whose first letter occurs again within the word (for

example, 'Baboon'). The program should display the resulting list.

7) Write a version of a palindrome recognizer that also accepts phrase

palindromes such as "Go hang a salami I'm a lasagna hog.", "Was it a rat I

saw?", "Step on no pets", "Sit on a potato pan, Otis", "Lisa Bonet ate no

basil", "Satan, oscillate my metallic sonatas", "I roamed under it as a tired

nude Maori", "Rise to vote sir", or the exclamation "Dammit, I'm mad!". Note

that punctuation, capitalization, and spacing are usually ignored.

8) In English, the *present continuous* is formed by adding the suffix -*ing* to the

verb

*go* -> *going*. A simple set of heuristic rules can be given as follows:

1. If the verb ends in *e*, drop the *e* and add *ing* (if not exception: *be*, *see*,

*flee*, *knee*, etc.)

2. If the verb ends in *ie*, change *ie* to *y* and add *ing*

3. For words consisting of consonant-vowel-consonant, double the final

letter before adding *ing*

4. By default just add *ing*

Write a function make\_ing\_form() which given a verb converts to present

Continuous form. Test your function with words such as *lie*, *see*, *move* and

*hug*.

9) Define a *procedure* histogram() that takes a list of integers and prints a

histogram to the screen. For example, histogram([4, 9, 7]) should print the

following:

\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

10) A *pangram* is a sentence that contains all the letters of the English alphabet at

least once, for example: *The quick brown fox jumps over the lazy dog*. write a

function to check a sentence to see if it is a pangram or not.

11) Write a program to solve a classic ancient Chinese puzzle: We count 35 heads

and 94 legs among the chickens and rabbits in a farm. How many rabbits and

how many chickens do we have?

12) A website requires the users to input username and password to register.

Write a program to check the validity of password input by users.

Following are the criteria for checking the password:

1. At least 1 letter between [a-z]

2. At least 1 number between [0-9]

3. At least 1 character from [$#@]

4. Minimum length of transaction password: 6

5. Maximum length of transaction password: 12

6. At least 1 letter between [A-Z]

Your program should accept a sequence of comma separated passwords and

will check them according to the above criteria. Passwords that match the

criteria are to be printed, each separated by a comma.

13) Write a program that maps a list of words into a list of integers representing

the lengths of the correponding words. Write it in three different ways: 1)

using a for-loop, 2) using the higher order function map(), and 3) using list

comprehensions

14) Write a program that prompts the user to enter types of fruit, and weight of

fruit. The program should then display the information in the form *fruit*,

*weight* listed in alphabetical order, one fruit type per line as shown below

Apple, 6 lbs.

Banana, 11 lbs.

etc.

15) Write a program to sort the (name, age, height) tuples by ascending order

where name is string, age and height are numbers. The tuples are input by

console. The sort criteria is:

1: Sort based on name

2: Then sort based on age;

3: Then sort by score.

16) In the word game Mad Libs, people are asked to provide a part of speech, such 6

as a noun, verb, adverb, or adjective. The supplied words are used to fill in the

blanks of a preexisting template or replace the same parts of speech in a

preexisting sentence. Although we don’t yet have the tools to implement a full

Mad Libs game, we can implement code that demonstrates how the game

works for a single sentence. Consider this sentence from P. G. Wodehouse:

Jeeves lugged my purple socks out of the drawer as if he were a vegetarian

fishing a caterpillar out of his salad.

Write a program that will do the following:

*•* Print the following template:

Jeeves [verb] my [adjective] [noun] out of the [noun]

as if he were a vegetarian fishing a [noun] out of his

salad.

*•* Prompt the user for a verb, an adjective, and three nouns.

*•* Print the template with the terms in brackets replaced with the words the user

provided.

17) In cryptography, a *Caesar Cipher* is a very simple encryption techniques in

which each letter in the plain text is replaced by a letter some fixed number of

positions down the alphabet. For example, with a shift of 3, A would be

replaced by D, B would become E, and so on.. *ROT-13* ("rotate by 13 places")

is a widely used example of a Caesar cipher where the shift is 13.

Write a program to implement an encoder/decoder of ROT-13. Once you're

done, you will be able to read the following secret message:

Pnrfne pvcure? V zhpu cersre Pnrfne fnynq!