BDAT 1004 – Data Programming  
Page 3 of 6  
F2021  
BDAT 1004Computer Studies  
Question 5  
a. Write a function inside(x,y,x1,y1,x2,y2) that returns True or False  
depending on whether the point (x,y) lies in the rectangle with lower left  
corner (x1,y1) and upper right corner (x2,y2).  
>>> inside(1,1,0,0,2,3)  
True  
>>> inside(-1,-1,0,0,2,3)  
False  
b. Use function inside() from part a. to write an expression that tests whether  
the point (1,1) lies in both of the following rectangles: one with lower left  
corner (0.3, 0.5) and upper right corner (1.1, 0.7) and the other with lower  
left corner (0.5, 0.2) and upper right corner (1.1, 2).  
Question 6  
16. You can turn a word into pig-Latin using the following two rules (simplified):  
• If the word starts with a consonant, move that letter to the end and append  
'ay'. For example, 'happy' becomes 'appyhay' and 'pencil' becomes 'encilpay'.  
• If the word starts with a vowel, simply append 'way' to the end of the word.  
For example, 'enter' becomes 'enterway' and 'other' becomes 'otherway' . For  
our purposes, there are 5 vowels: a, e, i, o, u (so we count y as a consonant).  
Write a function pig() that takes a word (i.e., a string) as input and returns its pig-  
Latin form. Your function should still work if the input word contains upper case  
characters. Your output should always be lower case however.  
>>> pig('happy')  
'appyhay'  
>>> pig('Enter')  
'enterway'

BDAT 1004 – Data Programming  
Page 4 of 6  
F2021  
BDAT 1004Computer Studies  
Question 7  
File bloodtype1.txt records blood-types of patients (A, B, AB, O or OO) at a clinic.  
Write a function bldcount() that reads the file with name name and reports (i.e.,  
prints) how many patients there are in each bloodtype.  
>>> bldcount('bloodtype.txt')  
There are 10 patients of blood type A.  
There is one patient of blood type B.  
There are 10 patients of blood type AB.  
There are 12 patients of blood type O.  
There are no patients of blood type OO.  
Question 8  
Write a function curconv() that takes as input:  
1. a currency represented using a string (e.g., 'JPY' for the Japanese Yen or  
'EUR' for the Euro)  
2. an amount  
and then converts and returns the amount in US dollars.  
>>> curconv('EUR', 100)  
122.96544  
>>> curconv('JPY', 100)  
1.241401  
The currency rates you will need are stored in file currencies.txt:  
AUD 1.0345157 Australian Dollar  
CHF 1.0237414 Swiss Franc  
CNY 0.1550176 Chinese Yuan

BDAT 1004 – Data Programming  
Page 5 of 6  
F2021  
BDAT 1004Computer Studies  
DKK 0.1651442 Danish Krone  
EUR 1.2296544 Euro  
GBP 1.5550989 British Pound  
HKD 0.1270207 Hong Kong Dollar  
INR 0.0177643 Indian Rupee  
JPY 0.01241401 Japanese Yen  
MXN 0.0751848 Mexican Peso  
MYR 0.3145411 Malaysian Ringgit  
NOK 0.1677063 Norwegian Krone  
NZD 0.8003591 New Zealand Dollar  
PHP 0.0233234 Philippine Peso  
SEK 0.148269 Swedish Krona  
SGD 0.788871 Singapore Dollar  
THB 0.0313789 Thai Baht  
Question 9  
Each of the following will cause an exception (an error). Identify what type of  
exception each will cause.  
Trying to add incompatible variables, as in  
adding 6 + ‘a’  
Referring to the 12th item of a list that has only 10  
items  
Using a value that is out of range for a function’s  
input, such as calling math.sqrt(-1.0)  
Using an undeclared variable, such as print(x)  
when x has not been defined  
Trying to open a file that does not exist, such as  
mistyping the file name or looking in the wrong  
directory.

BDAT 1004 – Data Programming  
Page 6 of 6  
F2021  
BDAT 1004Computer Studies  
Question 10  
Encryption is the process of hiding the meaning of a text by substituting letters in the  
message with other letters, according to some system. If the process is successful, no  
one but the intended recipient can understand the encrypted message. Cryptanalysis  
refers to attempts to undo the encryption, even if some details of the encryption are  
unknown (for example, if an encrypted message has been intercepted). The first step  
of cryptanalysis is often to build up a table of letter frequencies in the encrypted text.  
Assume that the string letters is already defined as  
'abcdefghijklmnopqrstuvwxyz'. Write a function called frequencies()  
that takes a string as its only parameter, and returns a list of integers, showing the  
number of times each character appears in the text. Your function may ignore any  
characters that are not in letters.  
>>> frequencies('The quick red fox got bored and went home.')  
[1, 1, 1, 3, 5, 1, 1, 2, 1, 0, 1, 0, 1, 2, 4, 0, 1, 2, 0, 2,  
1, 0, 1, 1, 0, 0]  
>>> frequencies('apple'