```python
#Family name: Mohammad Jaafreh
# Student number:  123456
# Course: IT1 1120
# Assignment Number 1

import math

########################
# Question 1
########################
## Repeat a string n times, with each repetition seperated by delim.
#  @param string the string of characters to repeat
#  @param n the number of repetitions
#  @param delim the delimiter between each repetition
#  @return the repeated string with delimiters
#
def repeat(string, n, delim) :
'''(string, number, symbol) -> string'''
   retval = string
   for i in range(1, n) :
      retval = retval + delim + string

   return retval

########################
# Question 2
########################

def is_prime(n):
     '''(number)-> Boolean
     checks whether positive integer n is a prime'''
    if n == 2:
        return True
    if n < 2 or n % 2 == 0:
        return False
    upper = math.ceil(math.sqrt(n))
    i = 3
    while i <= upper:
        if n%i == 0:
            return False
        i += 2
    return True


########################
# Question 3
########################
def points(x1, y1, x2, y2):
    ''' (number, number, number, number) -> none
    print slope and length of line segment passing
     through points (x1,y1) and (x2,y2)'''
    if x1 != x2:
        slope = (y2-y1)/(x2-x1)
    else:
        slope = 'infinity'
    distance = math.sqrt((x2-x1)**2+(y2-y1)**2)
    print('The slope is ' + str(slope) + ' and the distance is ' + str(distance))

########################
# Question 4
########################
(four solutions shown)
def month_apart(m1, d1, m2, d2):
    if (m1 == m2):
        return False
    elif (m1 <= m2 - 2):
```

```
67              return True
68          elif (m1 >= m2 + 2):
69              return True
70          elif (m1 == m2 - 1):
71              if (d1 <= d2):
72                  return True
73              else:
74                  return False
75          elif (m1 == m2 + 1):
76              if (d1 >= d2):
77                  return True
78              else:
79                  return False
80          else:
81              return False
82
83
84
85      def month_apart2(m1, d1, m2, d2):
86          if (m1 < m2 - 1 or m1 > m2 + 1):
87              return True
88          elif (m1 == m2 - 1 and d1 <= d2):
89              return True
90          elif (m1 == m2 + 1 and d1 >= d2):
91              return True
92          else:
93              return False
94
95
96      def month_apart3(m1, d1, m2, d2):
97          return (m2 - m1 > 1) or (m1 - m2 > 1) or (m2 - m1 == 1 and d1 <= d2) or (m1 - m2 ==
            1 and d1 >= d2)
98
99
100     def month_apart4(m1, d1, m2, d2):
101         return abs((m1 * 31 + d1) - (m2 * 31 + d2)) >= 31
102     ########################
103     # Question 5
104     ########################
105     def reverse_int(n):
106         'return integer obtained by reversing digits of 3-digit number n'
107         last = n%10
108         middle = (n//10)%10
109         first = n//100
110         return last*100 + middle*10 + first
111
112     ########################
113     # Question 6
114     ########################
115     def vowelCount(s):
116         'counts and prints the number of occurrences of each vowel in s'
117         print('a, e, i, o, and u appear, respectively', end='')
118         vowels = 'aeiou'
119         for vowel in vowels:
120             print(', {}'.format(s.count(vowel)), end='')
121         print(' times.')
122
123     ########################
124     # Question 7
125     ########################
126
127     ## Determine if 3 values are all the same.
128     #  @param x the first value
129     #  @param y the second value
130     #  @param z the final value
131     #  @return True if all the values are the same, False otherwise
```

```
#
def allTheSame(x, y, z) :
   if x == y and x == z :
      return True
   return False

## Determine if 3 values are all different.
#  @param x the first value
#  @param y the second value
#  @param z the final value
#  @return True if all the values are different, False otherwise
#
def allDifferent(x, y, z) :
   if x != y and x != z and y != z :
      return True
   return False

## Determine if 3 values are in increasing order.
#  @param x the first value
#  @param y the second value
#  @param z the final value
#  @return True if the values are in increasing order, False otherwise
#
def sorted(x, y, z) :
   if x <= y and y <= z :
      return True
   return False

 ########################
# Question 8
########################
def leap(year):
    'checks whether year is a leap year'
    if year%4 != 0:
        return False
    elif year%100 != 0:
        return True
    elif year%400 != 0:
        return False
    else:
        return True

 ########################
# Question 9
########################
def letter2number(lgrade):
    'returns the number grade corresponding to the letter grade lgrade'
    # handle + and - signs first
    if len(lgrade) == 1:
        add = 0.0
    elif lgrade[1] == '-':
        add = -0.3
    elif lgrade[1] == '+':
        add = 0.3

    if lgrade[0] == 'A':
        return 4 + add
    elif lgrade[0] == 'B':
        return 3 + add
    elif lgrade[0] == 'C':
        return 2 + add
    elif lgrade[0] == 'D':
        return 1 + add
    else:               # lgrade[0] must be 'F'
        return 0
 ########################
```

```
# Question 10
#########################
# Check if a string is a palindrome
def isPalindrome(s):
    # The index of the first character in the string
    low = 0

    # The index of the last character in the string
    high = len(s) - 1

    while low < high:
        if s[low] != s[high]:
            return False # Not a palindrome

        low += 1
        high -= 1

    return True # The string is a palindrome

 ########################
# Question 11
#########################
def is_nneg_float(s):
    """Does s denote a non-negative float?"""
    i = s.find('.')
    if i == -1:
        return s.isdigit()
    else:
        first = s[:i]
        second = s[i+1:]
        return ((first.isdigit() or len(first) == 0) and
                (second.isdigit() or len(second) == 0) and
                (len(first) > 0 or len(second) > 0))
 ########################
# Question 12
#########################
def rps(play1, play2):
    '''takes choices ('R', 'P', or 'S') of player 1 and 2,
       and returns -1 if player 1 wins, 1 if player 2 wins,
       or 0 if there is a tie'''
    if play1 == play2:
        return 0
    if (play1 == 'P' and play2 == 'R') or (play1 == 'R' and play2 == 'S') or (play1 ==
    'S' and play2 == 'P'):
        return -1
    else:
        return 1

 ########################
# Question 13
#########################
def alogical(n):
 ''' (number)->int
Precondition: n > 1
Returns the number of times n can be divided by 2 before we get something <=1
Thus you need to solve n/(2*2*2 ...)=n/(2**x) <= 1 equation for x.
That is a definition of log base 2 of n, log_2 n '''

 return math.ceil( math.log2(n) )

 ########################
# Question 14
#########################
def count_even_digits(n, length):
    count = 0
    for i in range(0,length):
```

```
        digit = n % 10
        n = n // 10
        if (digit % 2 == 0):
            count += 1

    return count

```