PHYS2160 Introductory Computational Physics 2021/22 Solutions to Exercise 1

1. (a)
```
Read length from the keyboard
Read width from the keyboard
Set the perimeter to two times the sum of the length and width
Print the value of the perimeter of a rectangle
```
   (b)
```
Read num from the keyboard
If num is equal to 5 or 6
    Print "Your number is 5 or 6."
Else
    Print "Your number is not 5 or 6."
End if
```
   (c)
```
Print 5 asterisks
Move to a new line
For every counter from 1 to 3
    Print *
    Print 3 spaces
    Print *
    Move to a new line
End For
Print 5 asterisks
Move to the new line
```

2. (a) `256`
   (b) `29.88888888888889`
   (c) `2.0`
   (d) `False`
   (e) `8`

3. (a)
```
from cmath import exp, polar
c = complex(input("Enter a complex number: "))
print(exp(c))
print(polar(c))
```
   (b)
```
from datetime import datetime
print("Today is", datetime.now().strftime("%Y-%m-%d"))
print("Current time is", datetime.now().strftime("%H:%M:%S"))
```

4. (a)
```
l = float(input("Enter the length of a square: "))
print("The area of the square is", l*l)
```
   (b)
```
p = int(input("Enter the principal amount: "))
n = int(input("Enter the number of years: "))
r = float(input("Enter the annual interest rate (%): "))
print("The simple interest is", p*n*r/100)
```

5. (a)
```
# sumofcubes.py
# This program prints the sum of the cubes of all the positive integers
# smaller than an input number using a while statement.
# Last update on 13 Jan 2021 by F K Chow

while True:
    n = int(input("Enter a positive integer: "))
    if n > 0:
        break
    print("invalid input")

i = 1
sum = 0
while i <= n:
    sum += i**3
    i += 1

print("The sum of the cubes of all the positive integers", end=" ")
print("up to", n, "is", sum)
```

(b)
```
# multtable.py
# This program prints the multiplication table showing all multiples
# from 1xn to nxn where 0 < n <= 10 using the for statement.
# Last update on 13 Jan 2021 by F K Chow

while True:
    n = int(input("Enter a positive integer no greater than 10: "))
    if n > 0 and n <= 10:
        break
    print("invalid input")

for i in range(1, n+1):
    print(i, "x", n, "=", i*n)
```

(c)
```
# divisors.py
# This program prints a list of all the divisors of an integer.
# Last update on 13 Jan 2021 by F K Chow

n = int(input("Enter an integer: "))
print("This number has divisors:")
for i in range(1, n+1):
    if (n % i) == 0:
        print(i)
```

(d)
```
# palindrome.py
# This program determines whether a five-digit integer is a palindrome.
# Last update on 13 Jan 2021 by F K Chow

while True:
    n = int(input("Enter a five-digit number: "))
    if (n >= 10000) and (n < 100000):
        break
    print("Invalid input!")

d1 = n % 10
d2 = (n % 100) // 10
d4 = (n % 10000) // 1000
d5 = (n % 100000) // 10000
if ((d1 == d5) and (d2 == d4)):
    print(n, "is a palindrome.")
else:
    print(n, "is not a palindrome.")
```

(e)

```
# fizzbuzz.py
# This program prints the numbers from 1 to 100. But it prints "Fizz"
# for the multiples of three, print "Buzz" for the multiples of five,
# and print "FizzBuzz" for the multiples of both three and five.
# Last update on 13 Jan 2021 by F K Chow

for i in range(1, 101):
    if ((i % 3) == 0) and ((i % 5) == 0):
        print("FizzBuzz")
    elif (i % 3) == 0:
        print("Fizz")
    elif (i % 5) == 0:
        print("Buzz")
    else:
        print(i)
```

6. (a)

```
ab_string = "abababababababab"
a_string = ""
for ch in ab_string:
    if ch != "b":
        a_string += ch
```

(b)

```
s = "abcdefghij"
s[::-1]
s[::3]
s[-2::-2]
```

(c)

```
# longerstr.py
# This program prints the longer string among two strings. If the two
# strings have the same length, then the program prints all these
# strings line by line.
# Last update on 13 Jan 2021 by F K Chow

s1 = input("Enter the first string: ")
s2 = input("Enter the second string: ")
if len(s1) > len(s2):
    print("The longer string is", s1)
```

```
elif len(s2) > len(s1):
    print("The longer string is", s2)
else:
    print("The following strings have the same length:")
    print(s1)
    print(s2)
```

(d)

```
# strlower.py
# This program prints a string in lower case without using the
# string.lower method.
# Written on 22 Jan 2020 by F K Chow

s = input("Enter a string: ")
sl = ""
for ch in s:
    if (ord(ch) >= 65) and (ord(ch) <= 90):
        sl += chr(ord(ch)+32)
    else:
        sl += ch

print(sl)
```

(e)

```
# revorder.py
# This program prints a long string with the words in reverse order.
# Last update on 13 Jan 2021 by F K Chow

lstr = input("Enter a long string: ")

lstrsp = " ".join(lstr.split(" ")[::-1])
print("The string with word order reversed:", lstrsp)
```

7. (a)

```
# printeven.py
# This program prints all even numbers from a given numbers list in
# the same order and stops printing any numbers that come after 237
# in the list.
# Last update on 13 Jan 2021 by F K Chow

list = [386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328,
        615, 953, 345, 399, 162, 758, 219, 918, 237, 412, 566, 826, 248,
        866, 950, 626, 949, 687, 217, 815, 67, 104, 58, 512, 24, 892,
        894, 767, 553, 81, 742, 717, 379, 843, 831, 445, 958, 743, 527]

for i in list:
    if not (i % 2):
        print(i, end=" ")
    if i == 237:
        break
```

(b)

```
# listoflists.py
# This program generates a list of lists of the form [[n], [n - 1, n],
# [n - 2, n - 1, n], ..., [1, 2, ..., n]] for a positive integer n.
# Last update on 2 Feb 2021 by F K Chow

while True:
    n = int(input("Enter a positive integer: "))
    if n > 0:
        break
    print("Invalid input")

lsto = []
for i in range(n, 0, -1):
    lsti = []
    for j in range(i, n+1):
        lsti.append(j)
    lsto.append(lsti)

print(lsto)
```

(c)

```
# oddsquareslst.py
# This program accepts a sequence of comma-separated integers and uses
# list comprehension to generate a list of the squares of each odd
# integer in the sequence.
# Last update on 13 Jan 2021 by F K Chow

instr = input("Enter a sequence of comma-separated integers: ")
ele = instr.split(",")
lst = []
for ch in ele:
    lst.append(int(ch))

oddsq = [i*i for i in lst if i % 2 == 1]
print("Squares of the odd numbers in the sequence:", end="")
for i in range(len(oddsq)-1):
    print(oddsq[i], end=",")

print(oddsq[-1])
```

(d)

```
# eventuple.py
# This program generates and prints another tuple whose values are
# even numbers in the tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10).
# Last update on 13 Jan 2021 by F K Chow

tup1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
lst = []
for i in tup1:
    if i % 2 == 0:
        lst.append(i)

tup2 = tuple(lst)
print(tup2)
```

(e)
```
# delevenele.py
# This program print the list after removing the 0th, 2nd, 4th, 6th
# elements from the list [12, 24, 35, 70, 88, 120, 155] by using list
# comprehension.
# Last update on 2 Feb 2021 by F K Chow

lst1 = [12, 24, 35, 70, 88, 120, 155]
lst2 = [x for (i, x) in enumerate(lst1) if i % 2 == 1]
print(lst2)
```

8. (a)
```
five times six equals 30.
```

(b)
```
Sum:   88, Average: 18.67
```

(c)
```
**** Surprise!###### ****
```

(d)
```
   +128
x    -8
-------
= -1024
```

(e)
```
1.235E-08
 0.000000012
```

9. (a)
```
def printProduct(s1, s2):
    ### Accepts two integers as strings and print their product ###
    return int(s1)*int(s2)
```

(b)
```
def ctUpLow(s):
    ### Count the uppercase and lowercase letters in a string ###
    up = low = 0
    for ch in s:
        if (ord(ch) >= 65) and (ord(ch) <= 90):
            up += 1
        elif (ord(ch) >= 97) and (ord(ch) <= 122):
            low += 1

    print("Number of uppercase letters = ", up)
    print("Number of lowercase letters = ", low)
```

(c)
```
def min_of_four(a, b, c, d):
    ### Find the minimum of four numbers ###
    lst = [a, b, c, d]
    return min(lst)
```

(d)
```
def isPrime(n):
    ### Determine whether a number is prime ###
    if (n == 1):
        return False
    elif (n == 2):
        return True
    else:
        for i in range(3, n):
            if (n % i == 0):
                return False
        else:
            return True
```

(e)

```
def sumOfSquares(n):
    ### Compute the sum of squares of the first n integers recursively ###
    if n == 1:
        return 1
    else:
        return n*n + sumOfSquares(n - 1)
```

10. (a)

```
# numoflines.py
# This program counts the number of lines in a text file.
# Last update on 2 Feb 2021 by F K Chow

with open("myfile.txt", "r") as infile:
    num = 0
    for line in infile:
        num += 1

print("Number of lines in myfile.txt =", num)
```

(b)

```
# combine.py
# This program prints the combinations of each line from the first
# file with the corresponding line from the second file as individual
# lines.
# Written on 22 Jan 2020 by F K Chow

with open('myfile1.txt') as infile1, open('myfile2.txt') as infile2:
    for line1, line2 in zip(infile1, infile2):
        # line1 from myfile1.txt, line2 from myfile2.txt
        print(line1.rstrip("\n") + " " + line2.rstrip("\n"))
```

(c)

```
# writeletterlines.py
# This program creates a file in which all letters in the English
# alphabet are listed in order by specified number of letters on
# each line.
# Last update on 2 Feb 2021 by F K Chow

while True:
    n = int(input("Enter the number of letters per line:  "))
    if (n > 0) and (n <= 26):
        break

alphabet = ""
for i in range(26):
    alphabet += chr(i+65)
```

```
with open("letterlines.txt", "w") as outfile:
    letters = [alphabet[i:i+n] + "\n" for i in range(0, len(alphabet),
                                                      n)]
    for line in letters:
        outfile.write(line)
```

11. (a)

```
# genarray.py
# This program takes two digits m and n as input and generates a mxn
# array where the element in the i-th row and j-th column of the
# array is i*j with i = 1, 2, ..., m and j = 1, 2, ..., n.
# Last update on 13 Jan 2021 by F K Chow

import numpy as np

while True:
    m = int(input("Enter the number of rows (< 10):  "))
    if (m > 0) and (m < 10):
        break
    print("Invalid input")

while True:
    n = int(input("Enter the number of columns (< 10):  "))
    if (n > 0) and (n < 10):
        break
    print("Invalid input")

a = np.zeros((m, n), dtype=int)
for i in range(1, m+1):
    for j in range(1, n+1):
        a[i-1, j-1] = i*j

print(a)
```

(b)

```
import numpy as np
a = np.arange(1, 17).reshape(4, 4)
a[:, 2]
a[1::2, :3]
a[::3, ::2]
```

(c)

```
# sortarray.py
# This program sorts a given 2x2 numpy array along the first axis,
# the last axis, and on the corresponding flattened array.
# Last update on 2 Feb 2021 by F K Chow

import numpy as np

a = np.random.randint(10, size=(2, 2))
b = a.copy()
b.sort(axis = 0)
c = a.copy()
```

```
c.sort(axis = 1)
d = a.flatten()
d.sort()
print("Original array:\n", a)
print("Sorting it along the first axis:\n", b)
print("Sorting it along the last axis:\n", c)
print("Sorting the corresponding flattened array:\n", d)
```

(d)

```
array([[1, 2],
       [3, 4],
       [2, 4],
       [6, 8]])
array([[1, 2, 2, 4],
       [3, 4, 6, 8]])
array([[[1, 2],
        [2, 4]],

       [[3, 6],
        [4, 8]]])
[array([[[0, 1, 2, 3],
        [4, 5, 6, 7]]]), array([[[ 8, 9, 10, 11],
        [12, 13, 14, 15]]]), array([[[16, 17, 18, 19],
        [20, 21, 22, 23]]])]
[array([[[ 0, 1],
        [ 4, 5]],

       [[ 8,  9],
        [12, 13]],

       [[16, 17],
        [20, 21]]]), array([[[ 2,  3],
        [ 6,  7]],

       [[10, 11],
        [14, 15]],

       [[18, 19],
        [22, 23]]])]
```

(e)

```
# tempconvertarr.py
# This program converts Celsius temperatures stored in a numpy array
# to Fahrenheit.
# Written on 22 Jan 2020 by F K Chow

import numpy as np

celsius = np.random.randint(100, size=10)
fahrenheit = 9*celsius/5 + 32
print("Temperatures in Celsius degrees:\n", celsius)
print("Correponding temperatures in Fahrenheit degrees:\n",
    fahrenheit)
```

(f)

```
# genarrayvec.py
# This program uses vectorization to create a numpy array that
# stores the values of the function z = x*y*exp(-(x*x+y*y)/2) for
# the dataset x and y which are both 100 uniformly spaced points
# over the interval [-3, 3].
# Written on 22 Jan 2020 by F K Chow

import numpy as np

x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
z = x*y*np.exp(-(x**2+y**2)/2)
print("z = ", z)
```

12. (a)

```
# bdaydict.py
# This program creates a dictionary of names and birthdays, then asks
# the user to enter a person’s name, and prints the birthday of that
# person.
# Last update on 13 Jan 2021 by F K Chow

bdays = dict([("Niels Bohr", "7 Oct 1885"),
              ("Albert Einstein", "14 Mar 1879"),
              ("Werner Heisenberg", "5 Dec 1901"),
              ("Max Planck", "23 Apr 1858"),
              ("Erwin Schrodinger", "12 Aug 1887")])

print("Welcome to the birthday dictionary. We know the birthdays of:")
for n in bdays:
    print(n)

name = input("Who’s birthday do you want to look up?  ")
if name in bdays:
    print("{:s}’s birthday is {:s}.".format(name, bdays[name]))
else:
    print("Sorry, we don’t have {:s}’s birthday.".format(name))
```

(b)

```
# charcount.py
# This program counts and prints the numbers of each character in a
# string input by console.
# Written on 22 Jan 2020 by F K Chow

instr = input("Enter a string: ")

charcount = {}
for ch in instr:
    if charcount.get(ch) == None:
        charcount[ch] = 1
    else:
        charcount[ch] += 1

print("Numbers of each character in this string:")
print("{:>9s} {:>8s}".format("Character", "Numbers"))
print("{:18s}".format("-"*18))
for ch, chcount in sorted(charcount.items()):
    print("{:^9s}  {:^7d}".format(ch, chcount))
```