Question 1 (5 points)

The following code computes a value of the sine function using the following formula:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$$

Remember, $n! = 1 \times 2 \times 3 \times 4 \times \ldots\ldots\ldots \times n$

```
def fact(var):
    x = 1
    y = sign(    )
    while var > 0:
        x = x*var
        var = var-1
    return [x,y]

def sign(var):
    return (-1)**var

def sin(x):
    var = 0
    for n in range(5):
        var = var + x**(2*n+1)*fact(2*n+1)[1]/fact(2*n+1)[0]
    return var

print(sin(3.1416))
```

What should be the correct argument of the sign() function?

- ○ var/2
- ○ (2*n+1)
- ○ 2*var
- ○ var
- ○ (var-1)/2
- ○ n

Question 2 (5 points)

```
def  (a)  (fName,lName):
    name = fName + ' ' + lName
    return (name)

pName = name("Mark", "Twain")
rName = name( (b) ("Samuel", "Langhorne"), "Clemens")

name = {}
name['pen'] = rName
name['real'] = pName

print('The pen name of', name[' (c) '], 'is', name[' (d) '])
```

Output:

```
The pen name of Samuel Langhorne Clemens is Mark Twain
```

Answer for the blank (a)

Answer for blank (b)

Answer for blank (c)

Answer for blank (d)

Question 3 (5 points)

Once the following program runs it asks for an input from the keyboard. If the input to the following program is 'Wednesday' then it outputs -100.

```
def fileRead(a):
    income = open('ledger.txt','r')
    text = income.read()
    income.close()
    lines = text.split('\n')
    return compute1(lines[1:],a)

def compute1(X,b):
    d = {}
    for x in X:
        x = x.split()
        d[x[0]] = (x[1][1:],x[2][1:])
    return compute2(d,b)

def compute2(s,c):
    return [redacted]

day = input()
print(fileRead(day))
```

Which of the following expression should be returned by the compute2 method?

- int(s[c][1]-s[c][0])
- f) int(s[c][1:])-int(s[c][:1])
- int(s[1][0]-c[1][0])
- int(s[c][1])-int(s[c][0])
- float(s[c][1]-s[c][0])
- float(s[c][1])-float(s[c][0])

Question 4 (5 points)

Consider the following code.

```
class AvgGrade:
    def __init__(self,name):
        self.name = name
    def avgGrade(self):
        f1 = open('Assignment1.txt','r')
        f2 = open('Assignment2.txt','r')
        scores1 = scores2 = avg = dict()
        for line in f1:
            if line.startswith('Name'):
                continue
            line = line.rstrip()
            x = line.split()[0]
            y = line.split()[2]
            scores1[x] = y
        for line in f2:
            if line.startswith('Name'):
                continue
            line = line.rstrip()
            x = line.split()[0]
            y = line.split()[2]
            scores2[x] = y
            s2 = int(y); s1 = [          ]
            avg[x] = (s2+int(s1))/2
        f1.close(); f2.close()
        print('Name:',self.name,' Average: ',avg[self.name])

name = input("""Input student's name: """)
student = AvgGrade(name)
student.avgGrade()
```

Input files:

Assignment1.txt - Notepad

File Edit Format View Help

```
Name ID Score
Mary 12387 85
Robert 12388 84
John 12432 81
Michael 12472 83
William 12516 89
David 12534 81
Susan 12550 77
Jessica 12594 73
Sarah 12641 87
Karen 12690 81
Nancy 12740 80
Lisa 12746 71
Matthew 12768 78
Margaret 12782 67
Mark 12832 74
```

Assignment2.txt - Notepad

File Edit Format View Help

```
Name ID Score
Mary 12387 95
Robert 12388 94
John 12432 91
Michael 12472 93
William 12516 79
David 12534 91
Susan 12550 76
Jessica 12594 83
Sarah 12641 86
Karen 12690 89
Nancy 12740 90
Lisa 12746 81
Matthew 12768 88
Margaret 12782 88
Mark 12832 94
```

Output:

```
Input student's name: Mark
Name: Mark  Average:  94.0
```

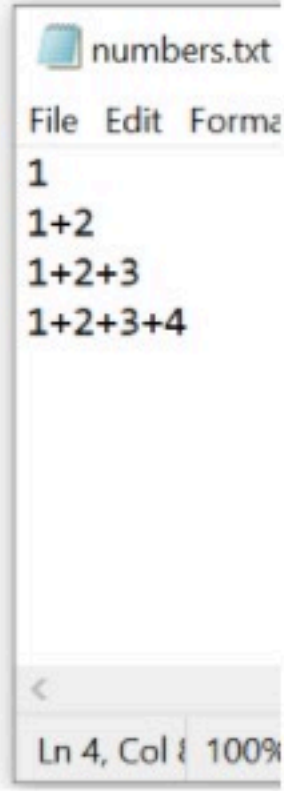
What should be the correct expression of avg[x]?

- ◯ scores1[int(x)]
- ◯ int(scores1[int(y)])
- ◯ int(scores1[x])
- ◯ int(scores1[y])
- ◯ int(scores1[int(x)])
- ◯ scores1[int(y)])

Question 5 (5 points)

```
def numfile():
    file = open('numbers.txt','r')
    numbers = []
    for line in file:
        num = ''
        line = line.rstrip()
        numList = line.split('+')
        for i in range(len(numList)):
            num = numList[i] + num
        numbers.append( (a) (num))
    file.close()
    return numbers
def compute(x):
    return sum(x)
print(compute(numfile()))
```

Text file:



Output: (b)

Answer of the blank (a)

Answer of the blank (b)

Page 1:

1

Page 2:

2

Page 3:

3

Page 4:

4

Page 5:

5

Page 6:

6

Page 7:

7

Page 8:

8

Page 9:

9

Page 10:

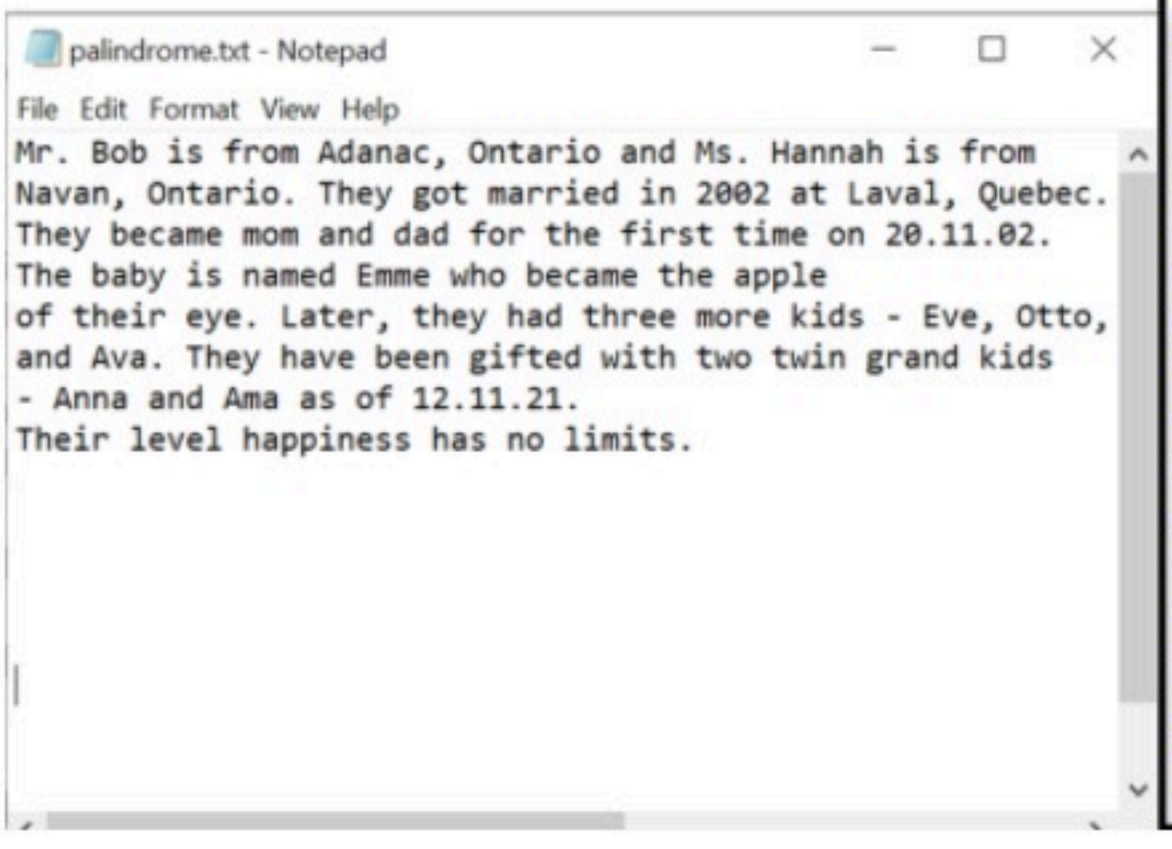
10

Question 6 (5 points)

A palindrome is a word or phrase which reads the same backward as forward, such as rotor, madam, level. The following code finds the palindrome words form a text and outputs the list of palindrome words with frequency (i.e., number of occurrence).

```
import string
file = open('palindrome.txt','r')
p = dict()
for x in file:
    x = x.rstrip()
    x = x.translate(x.maketrans('','',string.punctuation))
    x = x.lower()
    ws = x.split()
    for w in ws:
        flag = i = 0
        while [          ] :
            if w[i]!=w[-(i+1)]:
                flag = 1
                break
            i = i+1
        if bool(flag) == False:
            if w not in p:
                p[w] = 1
            else:
                p[w] = p[w] + 1
file.close()
for i in p:
    print(i,':',p[i])
```

Input text file:



Output:

```
bob : 1
hannah : 1
navan : 1
2002 : 1
laval : 1
mom : 1
dad : 1
201102 : 1
emme : 1
eye : 1
eve : 1
otto : 1
ava : 1
anna : 1
ama : 1
121121 : 1
level : 1
```

What should be the correct condition in the while statement?

What should be the correct condition in the while statement?

- flag<=int(len(w)/2)
- i< =len(w)/2
- i<int(len(w)/2)
- i<int(len(w))
- i<int(len(w))
- i<int(len(w)/3)

Next Page

Question 7 (5 points)

```
class exam:
    num = 25.0
    def __init__(self,num):
        self.num = num
        print((self.num)**(1/2),end='')

    def update(self,num):
        print(self.num%5,end='')
        return 55

    def __del__(self):
        print(self.num+10,end='')
        return 100

final = exam(16)
final = final.update(36)
```

Output:

Output of the code is

Next Page

Page 7 of 20

Question 8 (5 points)

```
class file:
    num = 1
    def __init__(self):
        try:
            f1 = open('final.txt','x')
        except:
            f1 = open('final.txt','a')
        f1.write(str(self.num))
        f1.close()
    def info(self):
        f1 = open('final.txt','a')
        f1.write(str(self.num))
        f1.close()
    def __del__(self):
        f2 = open('final.txt','a')
        f2.write(str(self.num))
        f2.close()
for x in range(3):
    a = file()
    a.info()
```

The output printed in the final.txt file:

Consider that the final.txt file did not exist before the code run.

Next Page

Question 9 (5 points)

Consider the following code snippet.

```
import random
items = [1000]
for i in range(999):
    items.append(random.randint(1,9))
items.sort(reverse=True);
del items[:1]
items.sort(reverse=True);
def average(x,y):
    return(x+y)
```

What would be the output of the following statement?

```
print(average(items[len(items)-1],items.pop()))
```

- ◯ 2
- ◯ 0
- ◯ 1
- ◯ 999
- ◯ 1000
- ◯ 1001

Question 10 (5 points)

The following code outputs 10.0.

```
def tic(a):
    a = a*a
    return tac(a)
def tac(b):
    b = b*b
    return toe(b)
def toe(c):
    c = c**(1/2)
    if c>10:
        return tic(c)
    else:
        return c
d = {'x':1,'y':2}
print(    )
```

What would be the correct print statement?

- None of the given answers
- print(tic(d['x'])-tac(d['x']-d['y'])+toe((d['x']-d['y'])**4))
- print(tic(d['x']+d['y'])+toe((d['x']-d['y'])**4))
- All of the given answers
- print(tic(d['x'])+tac(d['x']-d['y'])+toe((d['x']+d['y'])**4))
- print(tac(d['x'])+toe((d['x']+d['y'])**4))

Question 11 (5 points)

```
course = 'ITM200'
ln = len(course)
d = dict()
for i in (a) (ln):
    d[course[i]] = i
    print(i**2,end=' ')
print()
print(d['2']+d['0']+d['0'])
```

Outout:

0 1 4 9 16 25

(b)

Answer for the blank (a)

Answer for the blank (b)

Question 12 (5 points)

Consider the following code and output.

```
grades = {'Alex':4.0,'John':3.4,'Marie':4.33,
          'Pat':2.0,'Lee':3.9,'Bob':2.8,'Patel':3.6}
sortL = list()
def sortG(gList):
    ln = len(gList)
    for i in range(ln):
        temp = list(gList[0])
        for x,y in gList:
            if y < temp[1]:
                temp[1] = y
                temp[0] = x
        sortL.append((temp[0],temp[1]))
        [                              ]
    return printG(sortL)

def printG(var):
    for n,g in sortL:
        print(n,g)
sortG(list(grades.items()))
```

Output:

```
Pat 2.0
Bob 2.8
John 3.4
Patel 3.6
Lee 3.9
Alex 4.0
Marie 4.33
```

What would be the correct statement in the blank?

- ◯ sortL.remove((temp[0],temp[1]))
- ◯ sortL.extend((temp[1],temp[0])
- ◯ gList.append((y,x)
- ◯ del gList[i]
- ◯ gList.pop((temp[0],temp[1]))
- ◯ gList.remove((temp[0],temp[1]))

Question 13 (5 points)

The following code computes median. The x represents the median value. What should be the value of x?

```
import random
items = [0]
for i in range(10):
    items.append(random.randint(3,10))
items.sort()
print(items)
x = [ ]
print(items[x])
```

- ◯ len(items)/2
- ◯ int(len(items)%2)+1
- ◯ int(len(items)/2)+1
- ◯ int(len(items+1)/2)
- ◯ len(items)/2-1
- ◯ len(int(items)/2)+1

Question 14 (5 points)

Consider the following code.

```
class Student:
    def __init__(self,fName,lName,iD):
        self.fName= fName
        self.lName = lName
        self.iD = iD
    def info(self):
        print('Name:',self.fName,self.lName,', ID:',self.iD)

class University(Student):
    def __init__(self,fName,lName,iD,institute,year):
        super().__init__(fName,lName,iD)
        self.institute = institute
        self.year = year
    def infoUni(self):
        print("\tUniversity:",self.institute,", Year:",self.year)

class Department(University):
    def __init__(self,fName,lName,iD,institute,year,dept):
        super().__init__(fName,lName,iD,institute,year)
        self.dept = dept
    def infoDept(self):
        print('\tDepartment:',self.dept)

s = Department('Issac','Newton',123,'Ryerson','Sophomore','ITM')
s.info()
s.infoUni()
s.infoDept()
```

Which of the following information is incorrect?

- None of the given information is incorrect.
- The s object has 6 attributes. It calls constructor of Department once the object is created. It also calls the constructor of University from the Department's constructor.
- The constructor of Department is called once, the constructor of University is called twice, and constructor of Student is called thrice.
- All of the given information is correct.
- Student is a parent class, University is a both parent and child class, Department is not a parent class but a child class.
- There are 3 attributes in Student, 5 attributes in University, and 6 attributes in Department.

Next Page

Question 15 (5 points)

```
x = 1
if bool(x) is True:
        (a) (b) :
            x = 0;
            def (c) (self,x):
                self.x = x
b = a(2)
c = a(3)
b.x = b.x + c.x
c.x = b.x - c.x
b.x = b.x - c.x
d = str(c.x) + str(b.x) + str(a.x) + str(x)
print(d)
```

Output: (d)

Answer for the blank (a)

Answer for the blank (b)

Answer for the blank (c)

Answer for the blank (d)

Question 16 (5 points)

```
file = open('grades.txt','r')
gList = list()
grades = {}
i = 1
for line in file:
    line = line.rstrip()
    numList = line.split()
    grades[numList[0]] = (a)(numList[1])
file.close()
for x in grades:
    gList.append((grades[x],x))
gList.(b)(reverse=True)
for x in gList:
    print((c),(d))
```

Output:

```
Xing 4.0
Lee 3.9
Paul 3.8
Patel 3.6
Alex 3.4
Kim 2.8
Khan 2.0
```

Text file:

Answer for the blank (a)

Answer for the blank (b)

Answer for the blank (c)

Answer for the blank (d)

Next Page

Question 17 (5 points)

Consider the following code.

```
def fact(n):
    if n<0:
        print(n,end='!=')
        return 0
    elif n==0:
        print(n,end='!=')
        return 1
    elif n==1:
        print(n,end='=')
        return 1
    else:
        print(n,end='')
        print('x',end='')
        return [ ]
print(fact(-1))
print(fact(0))
print(fact(1))
print(fact(5))
print(fact(10))
```

Output:

```
-1!=0
0!=1
1=1
5x4x3x2x1=120
10x9x8x7x6x5x4x3x2x1=3628800
```

What should be in the blank?

- ◯ fact(n)
- ◯ (n-1)*fact(n-1)
- ◯ fact(n-1)
- ◯ (n-1)*fact(n)
- ◯ n*fact(n-1)
- ◯ n*fact(n)

Next Page

Question 18 (5 points)

```
(a)  string
file = open('limerick.txt','r')
(b)  = {}
for x in file:
    x = x.rstrip()
    x = x.translate(x.maketrans('','',string.punctuation))
    x = x.upper()

    if 'RELATIVITY' (c) x:
        words = (d) .split()
        for w in words:
            if w not in count:
                count[w] = 1
            else:
                count[w] = count[w]+1
print(count['RELATIVITY'])
```

*limerick.txt - Notepad

File Edit Format View Help

```
The Relativity is a popular comical limerick about Einstein's theory
of relativity, first appeared in an issue of the London humor magazine.
--------------------------------------------------
                Relativity.
There was a young lady named Bright
Whose speed was far faster than light;
She set out one day
In a relative way
And returned on the previous night.
--------------------------------------------------
Source:https://quoteinvestigator.com/2013/12/19/lady-bright/
```

output:

3

Main Content

Answer for the blank (a)

Answer for the blank (b):

Answer for the blank (c)

Answer for the blank (d):

Question 19 (5 points)

```
class Salary:
    def __init__(salary,rate,hour):
        (a).rate = rate
        (b).hour = hour

    def income(self):
        return self.rate*self.hour

def compute(x,y):
    total = x+y
    avg = total/2
     (c) (total,avg)

mr_X  = Salary(50,50)
mrs_X = Salary(60,40)


print('Mr. X\'s weekly income: ',mr_X.income())
print('Mrs. X\'s weekly income: ',mrs_X.income())
print('Weekly household income:',  (d) (mr_X.income(),mrs_X.income())[0])
```

Answer for blank (a) Answer for blank

(b) Answer for blank (c)

Answer for blank (d)

Next Page

Submit Quiz

15 of 20 questions saved