Santa Clara
University
School of Engineering

Honor Code of the School of Engineering

"All students taking courses in the School of Engineering agree, individually and collectively, that they will not give or receive unpermitted aid in examinations or other course work that is to be used by the instructor as the basis of grading."
-From the Graduate/Undergraduate Bulletin

I have read, understood, and agree to abide by the Honor Code of the School of Engineering.

Name: ____________________________

ID: ____________________________

Signature: ____________________________

Date: ________________________________

1. [30 points]

2. [30 points]

3. [30 points]

4. [30 points]

5. [30 points]

6. [30 points]

7. [30 points]

8. [30 points]

9. [30 points]

10. [30 points]

Total Score:

Final Examination

COEN 256 Principles of Programming Languages
Department of Computer Science and Engineering
Santa Clara University

Dr. Ming-Hwa Wang	Fall Quarter 2021
Phone: (408) 805-4175	Email address: mwang2@cse.scu.edu
Course website:	http://www.cse.scu.edu/~m1wang/language/
Office Hours:	Friday 9:00pm-9:30pm

1. [30 points] Given the DataFrame *df* indexed by *dates* below:

```
$ print(df)
                   A         B         C   D     F
2013-01-01  0.000000  0.000000 -1.509059   5   0.0
2013-01-02  1.212112 -0.173215 -1.389850  10   1.0
2013-01-03  0.350263 -2.277784 -1.884779  15   3.0
2013-01-04  1.071818 -2.984555 -2.924354  20   6.0
2013-01-05  0.646846 -2.417535 -2.648122  25  10.0
2013-01-06 -0.026844 -2.303886 -4.126549  30  15.0
```

a) How to get 0.350263 at row ‘2013-01-03’ and column ‘A’ by using df.at, df.iat, df.loc, df.iloc?

b) What is the output of df.apply(lambda x: x.max() – x.min())

2. [30 points] Give the coin change problem, we use greedy method to find the solution. Please provide two “coin systems” which one always get optimal greedy solution and the other may not get optimal greedy solution. Please also show why it is not optimal (i.e., counter example) for the second coin system.

3. [30 points] What is the output (order is important) of the following code:

```
class Bird:
  def __init__(self, name):
    print(name, "is a bird")
  def flight(self):
    print("Most of the birds can fly but some cannot.")
class Sparrow(Bird):
  def __init__(self, name):
    super().__init__(name)
  def flight(self):
    print("Sparrows can fly.")
class Ostrich(Bird):
  def __init__(self, name):
    super().__init__(name)
  def flight(self):
    print("Ostriches cannot fly.")
```

```
def func(obj):
    obj.flight()
for b in (Ostrich("ostrich"), Sparrow("sparrow"),
Bird("bird")):
    func(b)
```

4. [30 points]

a) What is the output of the code below?

```
from functools import reduce
def foo(m):
  return reduce(
    lambda x, y: x + y, [c * c for r in m for c in r])
m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(foo(m))
```

b) Please re-write the function foo using nested loops without reduce and get the same output.

5. [30 points] Given the incomplete class definition and main routine as below:

```
from datetime import date
class Person:
    pass
person1 = Person('Mary', 21)
person2 = Person.fromBirthYear('John', 2006)
print(person1.age)
print(person2.age)
# print the result
print(Person.isAdult(person1.get_age()))
print(Person.isAdult(person2.get_age()))
```

Please implement the methods __init__(), get_age(), fromBirthYear(), and isAdult() with optional decorator (@classmethod or @staticmethod) for the Person class such that it can generate the output as below.

```
21
15
True
False
```

6. [30 points] If f is a continuous function, $f(l) < 0$ and $f(r) > 0$ for integer $l < r$, there must be a root between l and r. If we know the root between l and r is an integer root. Please design a most efficient way to find the root by writing the pseudocode and give the big-Oh of your algorithm. Hint: divide-and-conquer.

7. [30 points] Substitute different digits (0, 1, 2, .., 9) for different letters below, so that the corresponding addition is correct, and the resulting value of MONEY is as large as possible. What are the values for all characters here?

```
  S E N D
+ M O R E
M O N E Y
```

8. [30 points] Given a matrix multiplication chain with n matrices (A_1, A_2, …, A_n with dimensions p_0xp_1, p_1xp_2, …, p_{n-1}xp_n) multiple together, the following code is not complete with missing statements indicated by “pass”. Please replace them to make the program working. To run an example, make n = 5, and p = [5, 10, 8, 15, 20, 4], then the output should be:

```
The number of scalar multiplications needed: 2200
Optimal parenthesization: (A[1](A[2](A[3](A[4]A[5]))))
```

The incomplete code is below:

```
def matrix_product(p):
    “””Return m and s.
    m[i][j] is the minimum number of scalar multiplications
    needed to compute the product of matrices A(i), A(I + 1),
    ..., A(j).
    s[i][j] is the index of the matrix after which the product is
    split in an optimal parenthesization of the matrix product.
    P[0... n] is a list such that matrix A(i) has dimensions
    p[I – 1] x p[i].
    “””
    length = len(p) # len(p) = number of matrices + 1
    # m[i][j] is the minimum number of multiplications needed to
    # compute the product of matrices A(i), A(i+1), ..., A(j)
    # s[i][j] is the matrix after which the product is split in
    # the minimum number of multiplications needed
    m = pass
    s = pass
    matrix_product_helper(p, 1, length – 1, m, s)
    return m, s
def matrix_product_helper(p, start, end, m, s):
    “””Return minimum number of scalar multiplications needed to
    compute the product of matrices A(start), A(start + 1), ...,
    A(end).
    The minimum number of scalar multiplications needed to
    compute the product of matrices A(i), A(I + 1), ..., A(j) is
    stored in m[i][j].
    The index of the matrix after which the above product is
    split in an optimal parenthesization is stored in s[i][j].
    p[0... n] is a list such that matrix A(i) has dimensions
    p[I – 1] x p[i].
    “””
    if m[start][end] >= 0:
        return m[start][end]
    if start == end:
        q = 0
    else:
        q = float(‘inf’)
        for k in range(start, end):
            temp = pass
            if q > temp:
                q = temp
                s[start][end] = k
    m[start][end] = q
    return q
def print_parenthesization(s, start, end):
    “””Print the optimal parenthesization of the matrix product
    A(start) x A(start + 1) x ... x A(end).
    S[i][j] is the index of the matrix after which the product is
    split in an optimal parenthesization of the matrix product.
    “””
    if start == end:
        pass
        return
    k = s[start][end]
    print(‘(‘, end=’’)
    print_parenthesization(s, start, k)
    print_parenthesization(s, k + 1, end)
    print(‘)’, end=’’)
n = 5
p = [5, 10, 8, 15, 20, 4]
m, s = matrix_product(p)
print(‘The number of scalar multiplications needed:’, m[1][n])
print(‘Optimal parenthesization: ‘, end=’’)
print_parenthesization(s, 1, n)
```

9. [30 points] Given a binary tree, the following incomplete code can report if the tree is balanced (i.e., the height difference between the left sub-tree and the right sub-tree of any node will be at most 1.) Please implement check_height() to make it working:

```
def check_heightI:
  “””
  Check height of balanced tree recursively.
  “””
  pass
def is_balancedI:
  “””
  Find if a tree is balanced tree.
  “””
  if check_heightI == -1:
    return False
  else:
    return True
```

10. [30 points] Given the input data as [3, 5, 2, 1, 4, 2, 5], please do the following operations and show the list/array (which implements the minheap with an extra first element 0 for easy computing parent/child):

a) build a minheap (i.e., by heapify())

b) insert 6 in the minheap

c) insert 1 in the minheap

d) delete the minimum element from the heap one by one