ES112 - Computing: Mid-Semester Exam

Guidelines

This exam lasts for 2 hours and papers will be collected at 12:30PM.

IMPORTANT: Please enter your name AND roll number CLEARLY on BOTH pages at the places indicated. Also write your roll number on top of all the middle pages. Otherwise, your exam will not be graded!

Mark your answers by filling out the appropriate circles. Every question has exactly one correct answer. Any multiple choice question that is marked more than once or has any overwriting will not be graded. An incorrect answer to any multiple choice question will incur a penalty of -1 point.

All the best!

NAME

_____

ROLL NUMBER

1 Part A

1. (1 point) Which of the following is not a comparison operator?
   ◯ >= ◯ != ◯ == √ =
2. (1 point) If x is the list $[1, 2, 3, 0]$, what is $x[-3 : -1]$?
   √ **[2, 3]** ◯ [2] ◯ [1,2,3] ◯ []
3. (1 point) If z=4//2/2, what is the value and type of z?
   ◯ 1, int √ **1, float** ◯ 4, int ◯ 4, float
4. (1 point) Let y=[1,2,3,4], what is y[y[2]]?
   ◯ 1 ◯ 2 ◯ 3 √ **4** ◯ Error

5. (1 point) What will be displayed by the following code?

```
myList = [1, 5, 5, 5, 5, 1]
max = myList[0]
indexOfMax = 0
for i in range(1, len(myList)):
    if myList[i] > max:
        max = myList[i]
        indexOfMax = i
print(indexOfMax)
```

◯ 0 √ **1** ◯ 2 ◯ 3 ◯ 4

6. (1 point) What is the output of the following code block?

```
d = {}
d[1] = 1
d['1'] = 2
d[1.0]=4
count = 0
for i in d:
    count += d[i]
print(count)
```

◯ An exception is thrown ◯ 3 √ **6** ◯ 2

7. (1 point) Which of the following is an invalid statement in Python?

◯ pqr=1,00,000 √ **p q r= 100 20 30** ◯ p,q,r=1,2,3 ◯ p_q_r=2,00,000

8. (1 point) Suppose T= [“abc", “def", “ghij"], Which of the following outputs ’FALSE’?

◯ T[1][-1]==T[1][2] ◯ T[2][2]==T[-1][2] √ **T[0][1]==T[-3][-3]** ◯ T[1][2]==T[-2][-1]

9. (1 point) What is the output of the following code?

```
s = "hello world"
print(s[::-1])
```

◯ ‘hello worl’ ◯ ‘world hello’ √ **‘dlrow olleh’** ◯ Error

10. (1 point) What is the output of the following program?

```
def replace(v):
  v[0]=9
x=[2,5,7]
replace(x)
print(x)
```

◯ [2,5,7] ◯ [9,2,5,7] √ **[9,5,7]** ◯ Error

11. (1 point) What does the following program do?

```
P = [1,2,3]
Q = P
for x in P:
    Q.append(x)
```

◯ Appends 1,2,3 to Q √ **Infinite Loop** ◯ Error ◯ Appends 1,2,3 to both P and Q.

12. (1 point) What is the output of the following program?

```
s=0
for i in range(0,2):
  for j in range(3,1,-1):
    if j>i:
      s=s+i**j
print(s)
```

√ **2** ◯ 5 ◯ 4 ◯ 3

13. (2 points) What is the output of the following program?

```
a=0
b=1
for i in range(10):
  temp=a+b
  a=b
  b=temp
print(temp)
```

◯ 34 √ **89** ◯ 55 ◯ 144

14. (2 points) What is the output of the following program?

```
def fun(n):
  s=1
  if n%2==1:
    return s
  else:
    return 1+fun(n/2)
print(fun(1024),fun(10240))
```

◯ (10,11) √ **(11,12)** ◯ (12,13) ◯ (13,14)

15. (1 point) What is the output of the following?

```
a = [0, 1, 2, 3]
i = -2
for i not in a:
    print(i)
    i += 1
```

◯ -2 -1 ◯ 0 √ **Error** ◯ None of the Above

2 Part B

1. (5 points) The code below is trying to print the GCD and LCM of n pairs of integers given as input. Identify four syntax errors and one semantic in the code. Write down the line number and the type of the error. You don't need to replicate the exact Python error message, but make sure to convey the essence of the problem.

```
def gcd(a, b):
    if b = 0:
        return a
    else
        return gcd(b, a % b)
lcm(a, b):
    return int((a * b) / gcd(a, b))
n = int(input())
for i in range n:
    l = list(map(int, input().strip().split()))
    a = l[0]
    b = l[1]
print(gcd(a, b), lcm(a, b))
```

Syntax Error: Assignment used instead of comparison: should be b == 0 on Line2

Syntax Error: Missing semicolon after else

Syntax Error: "def" missing before lcm

Syntax Error: range n should be range(n)

Semantic Error: Indentation of the print statement on the last line is incorrect.

2. (2 points) You want to write a program that prints the following self-explanatory pattern:

```
          *
         * *
        * * *
       * * * *
      * * * * *
     * * * * * *
    * * * * * * *
   * * * * * * * *
  * * * * * * * * *
 * * * * * * * * * *
```

The following code attempts to achieve this, taking as input n for the number of rows for which the pattern is to be repeated.

```
def triangle(n):
    k = 2*n - 2
    for i in range(0, n):
        for j in range(0, k):
            print(end=" ")
        k = k - 1
        for j in range(0, i+1):
            print("* ", end="")
        print("\r")
```

This code, unfortunately, is not correct. However, one small change can make it accurate. Identify the line to change and what to change it to. First give the line number and then in the box below it, state the correct replacement code.

2. _____

Correct Code: k = n - 1

3 Part C

1. Given a list **a**, you would like to swap the positions of the min and max elements. If there are multiple occurrences of the maximum and/or the minimum element in the list, then the swap should take place among the occurrences that have the smallest index. The following code attempts to perform this swap:

```
def swap_minmax(a):
    a[a.index(max(a))]=min(a)
    a[a.index(min(a))]=max(a)
```

(a) (1 point) Give an example of a list for which the code fails to swap the min and max elements.

Solution: Any list with an unique max element, or multiple max elements with the first max appearing before the first min.

(b) (1 point) Give an example of a list for which the code does swap the min and max elements.

Solution: [2,1,6,1,2,3,6] (Multiple max elements with the first min appearing before the first max.)

(c) (1 point) Is it possible to have a list with no repeated elements for which the code above works?
◯ Yes √ **No**

2. Let X be a list of n positive integers. A subset Y of X is called nice if it does not have any consecutive elements from X. For example, if $X = [1, 3, 2, 5, 9, 7, 9, 9]$, then $[1, 2, 9]$ is a nice subset, $[9, 9]$ is a nice subset (given by indices 4 and 6) but $[9, 7]$ and $[9, 9, 9]$ are not nice. The value of a subset Y of X is simply the sum of the elements in Y. You are asked to find a nice subset that has the maximum possible sum.

First, consider the following code to solve this problem:

```
def find_nicest(X):
    evenSet, oddSet = [],[]
    for j in range(len(X)):
        if j%2 == 0:
            evenSet.append(X[j])
        else:
            oddSet.append(X[j])
    if(sum(evenSet) > sum(oddSet)):
        return(evenSet)
    else:
        return(oddSet)
```

(a) (1 point) Give an example of a list for which the code fails to find the nicest subset.

Solution: [100,1,1,100,1] - any list with very large elements at odd *and* even positions.

(b) (1 point) Give an example of a list for which the code finds the nicest subset.

Solution: [1,1,1,1,1,1,1]

Now consider the following re-attempt:

```
def find_nicest_greedily(X):
    sum = 0
    while(X != []):
        j = X.index(max(X))
        sum += max(X)
        if j == 0:
            X = X[2:]
        elif j == len(X)-1:
            X = X[:-2]
        else:
            X = X[:j-1] + X[j+2:]
        print(X)
    return(sum)
```

(a) (1 point) Give an example of a list for which **find_nicest_greedily** fails to find the value of the nicest subset of X.

Solution: [50,98,50]

(b) (1 point) Give an example of a list for which **find_nicest_greedily** finds the value of the nicest subset of X.

Solution: [1,1,1,1,1,1,1]

3. In a barn, there are **C** cats and **D** dogs. Also, one day you go the field and find that there were **L** legs of the animals touching the ground. You know that cats love to ride on the dogs. So, they might ride on the dogs,

and their legs won't touch the ground and you would miss counting their legs. Any dog can have at most two cats on his back.

Since it is a foggy morning, you are wondering whether you counted the legs properly or not. Specifically, you are wondering if it is even possible that your counting is correct: that is, you want to know if there exist numbers X and Y such that $X + Y = C$, $X \leqslant 2D$, and $L = 4(D + Y)$. Although you can check this by brute force over all possible X and Y, but your friend has come up with the following shorter solution:

```
def sanity_check(C,D,L):
    if L%4 != 0:
        print "impossible"
    x = L/4
    if x < max(D, D + C - 2*D):
        print "impossible"
    else:
        print "possible"
```

(a) (1 point) Give an example of C, D and L for which the code above does not output the right answer. Enter three numbers below.

Solution: Any C and D such that $L/4 > D + C$.

(b) (2 points) How will you change the if condition in Line 4 of the function **sanity_check** to make the code correct?

Solution: if x < max(D, D + C - 2*D) or x > (D+C)

4. A palindrome is a string that reads the same from left to right as it does from right to left. Given a string x, return the shortest palindrome that can be produced by adding zero or more characters to the right end of x. There is always one unique answer. Consider the following attempt to solve the problem:

```
def palindromize(x):
    for i in range(1,len(x)):
        y = x + x[:i][::-1]
        if y == y[::-1]:
            ans = y
            break
    return ans
```

(a) (1 point) Give an example of a string for which **palindromize** does not return the optimal answer.

Solution: deabaed

(b) (1 point) Describe the set of strings for which the function above fails.

Solution: Any string that is already a palindrome.

(c) (1 point) Will the code have the same behavior if we replace **x[:i][::-1]** with **x[:i:-1]**?
◯ Yes √ **No**

5. In a modern city for Greek gods, the streets are geometrically arranged as a grid with integer coordinates with streets parallel to the x and y axes. For each integer value Z, there is a horizontal street at y=Z and a vertical

street at x=Z. This way, integer coordinate pairs represent the street junctions. During the hot days, the gods rest in cafeterias at street junctions. Messenger Hermes is to send photon messages to gods resting in the cafeterias by only moving along the city streets. Each message is for a single god, and it does not matter if the other gods see the message.

The messages are to be sent in a given order, and Hermes is provided the coordinates of the cafeterias in that order. Hermes starts from (0,0). To send a message to a cafeteria at (X_i, Y_i), Hermes only needs to visit some point on the same horizontal street (with y-coordinate Y_i) or on the same vertical street (with x-coordinate X_i). Having sent all of the messages, Hermes stops.

Below is a program that, given a sequence of cafeterias, attempts to find the minimum total distance Hermes needs to travel to send the messages.

```
def hermes(cafes):
    distance = 0
    curr = (0,0)
    for location in cafes:
        go_x = abs(location[0] - curr[0])
        go_y = abs(location[1] - curr[1])
        distance += min(go_x,go_y)
        if(go_x <= go_y):
            curr = (location[0],curr[1])
        else:
            curr = (curr[0],location[1])
    return(distance)
```

Give an example of input for which the code above fails. In the following three boxes, give a sequence of at most five locations, followed by the correct answer, followed by the output of the algorithm.

(a) (2 points) Example:

Solution: (8,3),(7,-7),(8,1),(-2,1),(6,-5)

(b) (2 points) Optimal Answer:

Solution: 11

(c) (2 points) Output of Algorithm:

Solution: 15

6. The latest idea floating in ES112 is that some students might work better in pairs for their lab project. So, we are going to experiment by pairing up some of the students to see if the performance of the class increases. However, only some pairs of students are compatible. Two students that are not compatible cannot be paired together.

   For each pair of compatible students, we have a number estimating how well the overall performance of the class might increase. This is called the performance potential of the pair. Of course, each student can only be paired with at most one other student. Furthermore, it is ok to not pair some students. So, your goal is to help us decide how to pair the students to maximize the overall performance increase in the class.

   You are given a list of compatible student pairs. The ith pair on the list has performance potential i. For example, if the list of compatible students is $[(6,3),(2,5),(6,4),(5,7)]$, then the performance increase we get by pairing 6 with 3 is just one, while if we pair 6 with 4, then we get a performance increase of three. Your are

required to return a set of pairs whose total performance potential is as large as possible. If there are several solutions, you can return any one of them.

The following code attempts to solve the problem.

```
def optimize_ES112(potentialpairs):
    answer = 0
    c,d = [],[]
    m = len(potentialpairs)
    for i in range (m-1,-1,-1):
        if ((potentialpairs[i][0] not in d) and (potentialpairs[i][1] not in d)):
            c.append(i)
            d.append(potentialpairs[i][0])
            d.append(potentialpairs[i][1])
    return(c)
```

Give an example of input for which the code above fails. In particular, below, give a sequence of at most five pairs for which the program above will not work, the optimal answer and the output of the function **optimize_ES112** on your input.

(a) (1 point) Example:

Solution: [(1,2),(3,4),(5,6),(3,6)]

(b) (1 point) Optimal Answer:

Solution: [2,1,0] with total potential 6

(c) (1 point) Output of Algorithm:

Solution: [3,0] with total potential 5

(d) (1 point) If the performance potential of the ith pair was 2^i instead of i, then would the function **optimize_ES112** always return the optimal answer?

√ **Yes** ○ No

Justify your answer below, either by providing an explanation (if you answered Yes) or a counterexample (if you answered No):

Solution: Compare the sum of the first j powers of two with the $(j+1)^{th}$ power of two.

7. Wall-E is learning to read today. He knows some subset of the letter of alphabet. In order to help Wall-E to study, you gave him a book with the text consisting of N words. Wall-E can read a word if it consists only of the letters he knows. Now you is curious about which words the robot can read, and which he cannot.

   You are given a lowercase letter string S consisting of the letters Wall-E can read. Every letter will appear in S no more than once. You are also given a word. The goal is to return Yes or No depending on whether Wall-E can read the word or not, repsectively.

   The code on the next page attempts to solve this problem.

```
def canread(S,word):
    flag = 1
    for k in word:
        if k in S:
            flag = 1
        else:
            flag = 0
    if flag:
        return('Yes')
    else:
        return('No')
```

(a) (1 point) Give an example of input for which the code above fails.

Solution: S = abc, word = xyza

(b) (1 point) Give an example of input for which the code above gives the correct answer.

Solution: S = abc, word = aabbcc; S = abc, word = abcxyz; etc.

(c) (1 point) How can you fix the code above so that it works? You can do this by adding one line of code. Specify where to add the line and what the code should be.

(c) _____

Solution: Add a break statement after line 7.

NAME

_____

ROLL NUMBER
