German University in Cairo
Media Engineering and Technology Faculty
Prof. Dr. Slim Abdennadher
Dr. Aysha El Safty

November 24, 2016

CSEN102: Introduction to Computer Science
Winter Semester 2016-2017
Midterm Exam

Bar Code

Instructions: Read carefully before proceeding.

1) Please tick your major

	Major
	Engineering
	BI

2) Duration of the exam: 2 hours (120 minutes).

3) No books or other aids are permitted for this test.

4) This exam booklet contains 11 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete**.

5) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the four extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets**.

6) When you are told that time is up, stop working on the test.

Good Luck!

Don't write anything below ;-)

Exercise	1	2	3	4	5	6	$\sum$
Possible Marks	10	14	10	12	16	10	72
Final Marks							

Exercise 1 Tracing (2+4+2+2 = 10 Marks)
Mysterious Task

a) Assign a value to x so that the character `'A'` is printed out:

```
x = _____________________________
if x%2==0 and x%3==1:
␣ print('A')
```

Solution:

$x = 4$

b) Assign values to x and y so that the character `'D'` is printed out. **Hint: The values could be floating-point numbers.**

```
x = _____________________________
y = _____________________________
if not ((0<=x<=3) and (0<=y<=3)):
␣ print('A')
elif y<=1 or y>=2:
␣ print('B')
elif x<=1 or x>=2:
␣ print('C')
else:
␣ print('D')
```

Solution:

$x = 1.2$
$y = 1.3$

c) What would be the output if the following code is executed?

```
x = float(10/4)
print(x)
```

Solution:

$x = 2.5$

d) Consider the two assignment statements

```
a = 1/2
b = 1.0/2
```

What are the values in the variables `a` and `b`? Explain your answers.

Solution:

$a = 0.5$
$b = 0.5$

Exercise 2 Conditional Algorithms (14 Marks)

Consider an airport boarding counter. Each passenger carries two suitcases with no extra charge if the weight is 23 Kg, or less, per each. The passenger will have to pay an extra charge of 50 LE per extra Kg for extra weight, up to a maximum weight of 32Kg per suitcase. If a suitcase weight exceeds 32Kg, the suitcase is rejected. Write a Python program that takes as input weights of two suitcases and prints out the weight charge.

For Example:

- If the first suitcase has a weight of 27Kg and the second Suitcase has a weight of 25Kg, then the program should display:

```
Suitcase 1 accepted
Suitcase 2 accepted
Total charge = 300 LE
```

- If the first suitcase has a weight of 45Kg and the second Suitcase has a weight of 20Kg, then the program should display:

```
Suitcase 1 rejected
Suitcase 2 accepted
Total charge = 0 LE
```

- If the first suitcase has a weight of 25Kg and the second Suitcase has a weight of 40Kg, then the program should display:

```
Suitcase 1 accepted
Suitcase 2 rejected
Total charge = 100 LE
```

- If the first suitcase has a weight of 25Kg and the second Suitcase has a weight of 18Kg, then the program should display:

```
Suitcase 1 accepted
Suitcase 2 accepted
Total charge = 100 LE
```

Solution:

```
A = eval(input ())
B = eval(input ())
total = 0
if(A <= 32):
␣ print("Suitcase 1 accepted")
␣ if(A> 23):
␣ ␣ total += (A-23) * 50
␣ if(B <= 32):
␣ ␣ print("Suitcase 2 accepted")
␣ ␣ if(B> 23):
␣ ␣ ␣ total += (B-23) * 50
␣ else:
␣ ␣ print("Suitcase 2 rejected")
else:
␣ print("Suitcase 1 rejected")
␣ if(B <= 32):
␣ ␣ print("Suitcase 2 accepted")
```

```
    if(B> 23):
        total += (B-23) * 50

    else:
        print("Suitcase 2 rejected")
print("Total charge =", total ,"LE")
```

Exercise 3 Iterative Algorithms (10 Marks)

A perfect square is a number that can be expressed as the product of two equal integers.

Write a function `first_N` that takes a positive integer, `n`, as its only argument. The function should print the first `n` perfect squares that are not even numbers.

For example

- if `n` was 3 it should print the perfect squares 1, 9, and 25.
- if `n` was 5 it should print the perfect squares 1, 9, 25, 49 and 81.

Solution:

```
def first_N(n):
    r = 1
    i = 1
    p = 1
    while(i <= n):
        r = p*p
        if(r%2 != 0):
            print(r)
            i += 1
        p += 1

first_N(5)
```

Exercise 4 Iterative Algorithms (12 Marks)

Write a Python program that takes a sorted (possibly empty) list of numbers `thelist` and an integer `x`. The program creates a new list containing the same elements of the input list and `x`. `x` should be put into the new list at the correct position so it is still ordered. If `x` is already in `thelist`, then the program should not insert `x` into the new list. For example,

- if `a = [0,2,4,5]` and `x=3`, the program should create and display `[0,2,3,4,5]`.
- if `a = [1,2,3,7]` and `x=-1`, the program should create and display `[-1,1,2,3,7]`.
- if `a = [1,2,7]` and `x=2`, the program should create and display `[1,2,7]`
- if `a = []` and `x=4`, the program should create and display `[4]`
- if `a = [0,2,4,5]` and `x=7`, the program should create and display `[0,2,4,5,7]`

Solution:

```
a = eval(input())
x = eval(input())
i = 0
c = []
flag = False
if(len(a) == 0):
    c = c + x
else:
    while(i < len(a)):
        if(a[i] > x and flag == False):
            c = c + x
            c = c + a[i]
            flag = True
        elif(a[i] == x and flag == False ):
            c = c + a[i]
            flag = True
        elif(a[i] > x or a[i] < x):
            c = c + a[i]
        i += 1
    if(flag == False):
        c = c + x
print(c)
```

Exercise 5 Iterative Algorithms (16 Marks)
Shuffling Cards

In the spirit of card games, this question is about shuffling cards or more generally, shuffling lists. When you shuffle cards you make two stacks, then interleave them, and if you were to do it perfectly, the even numbered cards would come from one stack and the odd numbered cards from the other stack. We call this even-odd interleaving a perfect shuffle. Of course, you can do the same thing with any two lists, regardless of what type they contain. Your job is to write a function that shuffles together two lists in this way. Here is the specification: the function `shuffle(a, b)` perfectly shuffles two lists and returns a new list that contains the items in `a` and `b`, interleaved in the order `a[0], b[0], a[1], b[1], a[2], ....` If one list is longer than the other, the extra items go at the end. For example,

- `shuffle([1,4,6],[2,8,0,3,6])` returns `[1,2,4,8,6,0,3,6]`
- `shuffle([1,4,6],[2,8,0])` returns `[1,2,4,8,6,0]`

Solution:

```
def shuffle(a,b):
␣  c = []
␣  i = 0
␣  if(len(a) <= len(b)):
␣␣     n = len(a)
␣  else:
␣␣     n = len(b)
␣  while(i < n):
␣␣     c = c + a[i] + b[i]
␣␣     i += 1
␣  if(i < len(a)):
␣␣     while(i < len(a)):
␣␣␣        c = c + a[i]
␣␣␣        i += 1
␣  elif(i < len(b)):
␣␣     while(i < len(b)):
␣␣␣        c = c + b[i]
␣␣␣        i += 1
␣  print(c)
a = eval(input())
b = eval(input())
shuffle(a,b)
```

Exercise 6 Tracing (8+2 = 10 Marks)
Mysterious Task

```
A = eval(input ())      # list
n = eval(input ())      # integer
i = 0
j = len(A) - 1
while ( i <= j ):
␣ if( A[i] >= n and A[j] < n):
␣ ␣ temp = A[i]
␣ ␣ A[i] = A[j]
␣ ␣ A[j] = temp
␣ ␣ i += 1
␣ ␣ j -= 1
␣ elif(A[i] >= n):
␣ ␣ j -= 1
␣ elif(A[j] < n):
␣ ␣ i += 1
␣ else:
␣ ␣ i += 1
␣ ␣ j -= 1
print(A)
```

a) What does the program display for the following input? Trace the program for:

A = [11,4,2,8,9,7,3,5]
n = 6

Solution:

i	j	temp	A[i]	A[j]
0	7	11	5	11
1	6	11	4	3
2	6	11	2	3
3	6	8	3	8
4	5	8	9	7
4	4	8	9	9
4	3			

$$[5, 4, 2, 3, 9, 7, 8, 11]$$

b) What does the program do for any input list and an integer? State it in one statement in English. Don't explain the algorithm steps.

Solution:

The algorithms re-orders the elements in the list based on their values and the integer `n`. Elements less than `n` are put at the beginning of the list and elements larger than `n` are put at the end.

Exercise 7 Iterative Algorithms (Bonus = 8 Marks)
Bonus

Write a Python program that takes as input the result of multiplying two integers A and B and the difference between them. Your program then finds the values of A and B.

Example:

a) Input: multiplication result= 12, difference = 1
Output: A=3, B=4

b) Input: Multiplication result= 540, difference = 7
Output: A = 20, B = 27

c) Input: Multiplication result= 121, difference = 0
Output: A = 11, B = 11

Solution:

```
m = eval(input())
d = eval(input())
a = 1
b = a + d
while(m != a*b):
    a += 1
    b = a+d
print(a,b)
```

Scratch paper

Scratch paper

Scratch paper