German University in Cairo
Media Engineering and Technology Faculty
Prof. Dr. Slim Abdennadher
Dr. Amr Desouky
Dr. Mohamed Elmahdy

December 4, 2014

CSEN102: Introduction to Computer Science
Winter Semester 2014 - 2015
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 13 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	6	6	10	16	16	16	70
Final Marks							

Exercise 1 Sequential Algorithms (6 Marks)

Consumer Price Index

The formula for calculating the Inflation Rate using the Consumer Price Index (CPI) is relatively simple.

Assume for the sake of simplicity that the index consists of one item and that in 1984 that item cost 100 dollars. In December of 2014 that same item would probably cost 198 dollars.

Let us calculate the price difference between 1984 and 2014.

- Step 1: Calculate How Much has the Consumer Price Index Increased?

  By looking at the above example, common sense would tell us that the index increased (it went from 100 to 198).

  The question is how much has it increased?

  To calculate the change we would take the second number (198) and subtract the first number (100). The result would be 98. So we know that from 1984 until 2014 prices increased (Inflated) by 98 points.

- Step 2: Comparing the CPI Change to the Original CPI

  Since we know the increase in the Consumer Price Index we still need to compare it to something, so we compare it to the price it started at (100). We do that by dividing the increase by the first price or 98/100. the result is (.98).

- Step 3: Convert it to a Percentage

  This number is still not very useful so we convert it into a percentage. To do that we multiply by 100.

  So the result is a 98% increase in prices since 1984.

Write an algorithm that given two prices of the same item in two different years computes the CPI. The algorithm should display the following for the example above

```
The price of the item in year 1984 is 100
The price of the item in year 2014 is 198
The CPI is 98%
```

Solution:

```
get firstYear, secondYear, oldPrice, newPrice
set CPI to newPrice - oldPrice
set CPI to CPI / oldPrice
set CPI to CPI * 100
print "The price of the item in year ", firstYear, "is", oldPrice
print "The price of the item in year ", secondYear, "is", newPrice
print "The CPI is", CPI, "%"
```

Exercise 2 Conditional Algorithms - Refactoring (6 Marks)
Grade and Percentage

Given the following algorithm

```
get grade
if (grade = 'A')
then print  85
endif
if (grade = 'B')
then print  67
endif
if (grade = 'C')
then print  50
endif
if (grade = 'F')
then print  30
endif
if (NOT (grade = 'A' OR  grade = 'B' OR grade = 'C' OR grade = 'F'))
then print  -1
endif
```

a) What are the main drawbacks (disadvantages) of the this algorithm?

Solution:

All conditions will be checked even if we do not need to check them. Moreover, the last condition includes many checks that can be removed if nested-if statements are used.

b) Rewrite this algorithm to avoid the drawbacks you listed in part a).

Solution:

```
get grade
if (grade = 'A')
then print  85
else
  if (grade = 'B')
  then print  67
  else
    if (grade = 'C')
    then print  50
    else
      if (grade = 'F')
      then print  30
      else
        print  -1
      endif
    endif
  endif
endif
```

Exercise 3 Conditional Algorithms (10 Marks)

The Sum of Rounded Numbers

Write an algorithm that given three integers displays the sum of their rounded values.

We will round an integer value to the next multiple of 10 if its rightmost digit is 5 or more, so 15 rounds up to 20. Alternately, round down to the previous multiple of 10 if its rightmost digit is less than 5, so 122 rounds down to 120.

Here are some examples of evaluating the algorithm on representative input values:

```
 16 17 12      -> 50
 12 13 14      -> 30
 6  4  4       -> 10
```

Solution:

```
get a, b, c
set aa to a % 10
set bb to b % 10
set cc to c % 10

if(aa >= 5)
then
  set a to (10 + a) - aa
else
  set a to a - aa
endif

if(bb >= 5)
then
  set b to (10 + b) - bb
else
  set b to b - bb
endif

if(cc >= 5)
then
  set c to (10 + c) - cc
else
  set c to c - cc
endif

set sum to a + b + c
print sum
```

Exercise 4 Iterative Algorithms (8+8=16 Marks)
362 Pattern

a) Write an algorithm that given a list of integers displays `true` if the list contains a `3, 6, 2` pattern; that is, a value, followed directly by the value plus 3, followed directly by the value minus 1.

Here are some examples of evaluating the algorithm on representative input values:

```
{1, 3, 6, 2}       -> true
{1, 2, 7, 1}       -> false
{3, 6, 2}          -> true
{4, 7, 3}          -> true
{5, 8}             -> false
{1, 3, 6, 2, 5, 1} -> true
```

Your algorithm should stop whenever the pattern is found.

Solution:

```
get n
get A1....An
set i to 1
set flag to false

if(n >= 3)
then
   while(i < n-1 and flag = false)
   {
      if(A[i+1] - A[i] = 3 and A[i+2] - A[i] = -1)
      then
         set flag to true
      endif
      set i to i+1
   }
endif

print flag
```

b) Write an algorithm that given a list of integers displays the number of `3, 6, 2` pattern in the list. Here are some examples of evaluating the algorithm on representative input values:

```
{1, 3, 6, 2, 5, 1} -> 2
{1, 2, 7, 1}       -> 0
{3, 6, 2, 5}       -> 1
{4, 7, 3}          -> 1
```

Solution:

```
get n
get A1....An
set i to 1
set count to 0

if(n >= 3)
then
   while(i < n-1)
   {
      if(A[i+1] - A[i] = 3 and A[i+2] - A[i] = -1)
      then
         set count to count + 1
      endif
      set i to i+1
   }
endif
print count
```

Exercise 5 Iterative Algorithms (8+8=16 Marks)
Deletion of Bad Pairs

a) Write an algorithm that takes a list of numbers and removes any adjacent pair of integers in the list if the left element of the pair is larger than the right element of the pair. Please note that every pair's left element is stored in an odd-numbered index in the list, and every pair's right element is stored in an even-numbered index in the list. For example, suppose a list `A` stores the following element values:

```
[3, 7, 9, 2, 5, 5, 8, 5, 6, 3, 4, 7, 3, 1]
```

the algorithm should display the following sequence:

```
3 7 5 5 4 7
```

becauase the pairs `(9,2)`, `(8,5)`, `(6,3)`, and `(3,1)` are considered to be *bad* because the left element is larger than the right one. Thus, these pairs should be removed.

Solution:

```
get n
get A1....An
set i to 1
while(i < n)
{
   if(A[i] <= A[i+1])
   then
      print A[i]
      print A[i+1]
   endif
   set i to i + 2
}
```

b) Assume that the elements of the list are represented now as an integer number. For example, the list

```
[3, 7, 9, 2, 5, 5, 8, 5, 6, 3, 4, 7, 3, 1]
```

is now represented as the integer number

```
37925585634731
```

You are asked to perform the same task described in Part a) however this time you are not allowed to use lists. You have to work on the integer number and your algorithm should display for the number above the following result:

```
375547
```

Note that the order is important. You are not allowed to display `745573` and **you are not allowed to use neither lists nor strings at all**.

For simplicity, assume that the number consists of an even number of digits.

Solution:

```
get n
set i to 1
set j to 10
set result to 0

while(n > 0)
{
   set n1 to n % 10
   set n to INT(n/10)
   set n2 to n % 10
   set n to INT(n/10)

   if( n1 >= n2)
   then
      set result to result + (n2*j) + (n1*i)
      set i to i * 100
      set j to j * 100
   endif
}

print result
```

Exercise 6 Tracing Iteration (8+4+4=16 Marks)
Mysterious Task

Given the following algorithm

```
get m
get A1, ..., Am
get n
get B1, ..., Bn
set i to 1
set j to 1
set flag to true
while(i <= n AND j <= m AND flag = true){
    if (A[j] < B[i])
       then set j to j + 1
    else
       if (A[j] = B[i])
       then
           set i to i + 1
           set j to j + 1
       else
          if (A[j] > B[i])
          then
             set flag to false
          endif
       endif
    endif
}

if (i  <= n OR flag = false)
then
    print("Mysterious No")
else
    print("Mysterious Yes")
endif
```

a) What is the output of the algorithm above for the following two lists:

```
List A: 1 2 5 8 44
List B: 2 6 8
```

Draw a tracing table.

Solution:

i	j	flag	A[j]	B[i]
1	1	true	1	2
1	2	true	2	2
2	3	true	5	6
2	4	true	8	6
2	4	false		

Mysterious No

b) What does the algorithm do for any two sorted (in increasing order) lists `A` and `B`, i.e. what is the meaning of `"Mysterious Yes"` and `"Mysterious No"`?

Solution:

The algorithm checks whether the list `B` is a subset of list `A`. Thus, it will print `"Mysterious Yes"` if list `B` is a subset of list `A` and prints `"Mysterious No"` otherwise.

c) Try to execute the algorithm on the following lists

```
List A: 1 2 5 8 44
List B: 1 2 5 8 44 100 202
```

According to the functionality of your algorithm, this case can be handled in a more efficient way. Add the corresponding statements in the code below for the case where the length of list `B` is greater than the length of list `A`.

```
get m
get A1, ..., Am
get n
get B1, ..., Bn
set i to 1
set j to 1
set flag to true
while(i <= n AND j <= m AND flag = true){
    if (A[j] < B[i])
       then set j to j + 1
    else
       if (A[j] = B[i])
       then
           set i to i + 1
           set j to j + 1
       else
          if (A[j] > B[i])
          then
             set flag to false
          endif
       endif
    endif
}

if (i  <= n OR flag = false)
then
    print("Mysterious No")
else
    print("Mysterious Yes")
endif
```

Solution:

```
get m
get A1, ..., Am
get n
get B1, ..., Bn
set i to 1
set j to 1
set flag to true
```

```
if(m > n)
then
   while(i <= n AND j <= m AND flag = true){
       if (A[j] < B[i])
          then set j to j + 1
       else
          if (A[j] = B[i])
          then
              set i to i + 1
              set j to j + 1
          else
             if (A[j] > B[i])
             then
                set flag to false
             endif
          endif
       endif
   }
endif
if (i  <= n OR flag = false)
then
    print("Mysterious No")
else
    print("Mysterious Yes")
endif
```

Scratch paper

Scratch paper

Scratch paper