**German University in Cairo**
**Computer Science and Engineering Department**
**Prof. Dr. Slim Abdennadher**

December 1, 2007

# Introduction to Computer Science
# Winter Term 2007–2008
## Midterm Exam

**Bar Code**

**Instructions: Read carefully before proceeding.**

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

2) (Non-programmable) Calculators are allowed.

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

4) This exam booklet contains 12 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 three 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 | 7 | ∑ |
|---|---|---|---|---|---|---|---|---|
| Marks | 6 | 7 | 5 | 8 | 8 | 8 | 8 | 50 |
| Final Marks | | | | | | | | |

**Exercise 1**          (6 Marks)
       **Sequential Algorithm**

A theater owner would like to calculate his daily profit. The theater has three classes of tickets; the price of the first class tickets is 150 LE, second class is 100 LE, and third class is 75 LE. The profit is 20% of the ticket price. The owner loses 70 LE. for each unreserved seat in the theater.

Write an algorithm that given the total seats in the theater and the reserved seats out of each class calculates the total profit, total loss, and the difference between them.

Running example: For the following input

```
Total Number of seats: 300
First class tickets sold: 70
Second class tickets sold: 75
Third class tickets sold: 90
```

the algorithm should output

```
Total profit: 4950
Total loss: 4550
Difference: 400
```

**Solution:**

```
get totalseats
get FClass, SClass, TClass
get totalprofit to (FClass*150+ SClass*100+ TClass*75) * 0.2
set totalloss to (totalseats - FClass - SClass - TClass)*70
set difference to totalprofit-totalloss
print totalprofit
print totalloss
print difference
```

**Exercise 2**                                                                                              (7 Marks)
### Conditional Algorithm

The tuition fees of a student is dependent on his/her registered credit hours according to the given following rules.

| Credit Hours | Tuition Fees |
|--------------|--------------|
| 0 − 40       | 3000         |
| 41 − 75      | 5000         |
| 76 − 100     | 6000         |

Write an algorithm that given the number of credit hours for a student determines the corresponding tuition fees for the student and print out the result. Note that a student is not allowed to take more than 100 credit hours. Thus the algorithm should print the corresponding message.

**Solution:**

```
get creditHours
if (creditHours >= 0 AND creditHours =< 40)
then print 3000
else if (creditHours =< 75)
     then print 5000
     else if (creditHours =< 100)
          then print 6000
          else print "You are not allowed to take more than 100 credit hours"
          endif
     endif
endif
```

**Exercise 3** (5 Marks)
### Conditional Algorithm

The following algorithm prints out whether a current student is in elementary (1st - 5th), middle (6th - 8th), or high school (9th - 12th).

```
get grade
if (grade <= 5) then
    print "this student is in elementary school"
else if (grade <= 8) then
        print "this student is in middle school"
        else if (grade <= 12) then
                print "this student is in high school"
            endif
    endif
endif
```

The algorithm above uses nested if-statements.

a) Write an equivalent algorithm that will print the same messages as the algorithm above **without using any nested if-statements**.

**Solution:**

```
get grade
if (grade <= 5) then
    print "this student is in elementary school"
endif
if (grade <= 8 AND grade > 5) then
    print "this student is in middle school"
endif
if (grade <= 12 AND grade > 8) then
    print "this student is in high school"
endif
```

b) Discuss the drawback of your algorithm? **Hint:** Compare the efficiency of both algorithms.

**Solution:**

The first algorithm is more efficient than the second one since in the second algorithms all conditions should be checked. In the first algorithm the number of conditions that should be checked depend on the input.

The worse-case scenario in the first algorithm occurs if the grade is greater than 8. In this case, three conditions (three operations) should be checked. In the second algorithm, three conditions should be checked. However each condition consists of 2 operations.

**Exercise 4**                                                                                   (8 Marks)
          **Iterative Algorithm**

The *Rabbitiski sequence* of numbers is formed by the following rule: Each number is equal to the product of the 2 previous numbers.

Write an algorithm that given the first two numbers and the length of the sequence **n** prints the Rabbitiski sequence consisting of **n** numbers

Example:

```
Enter first number: 4
Enter second number: 5
Enter the length of the series: 5
The series is:
4 5 20 100 2000
```

**Solution:**

```
get firstNumber
get SecondNumber
get length
print firstNumber
print SecondNumber
set i to 2
while (i < length) {
   set result to firstNumber * SecondNumber
   print result
   set firstNumber to SecondNumber
   set SecondNumber to result
   set i to i + 1
}
```

**Exercise 5** (8 Marks)
**Iterative Algorithm – Tracing**

Given the following algorithm, where `A1, A2, ..., An` is a list of `n` elements

```
get n
get A1,A2,...,An
set i to n
set j to 1
set temp to 0
while (i>=j)
{
   set temp to Ai
   set Ai to Aj * 2
   set Aj to temp * 2
   set i to i-1
   set j to j+1
}
set i to 1
while (i<=n)
{
   print Ai
   set i to i+1
}
```

a) What is the output of the algorithm for the following list of length 7?

   8 15 3 12 10 11 14

Use a tracing table to trace the first `while` loop.

**Solution:**

| temp | i | j | Ai | Aj |
|------|---|---|----|----|
| 0    | 7 | 1 | 14 | 8  |
| 14   | 6 | 2 | 16 | 28 |
| 11   | 5 | 3 | 30 | 22 |
| 10   | 4 | 4 | 6  | 20 |
| 12   | 3 | 5 | 24 | 24 |

What will be displayed using the second `while` loop?

**Solution:**

28 22 20 24 6 30 16

b) What is the output of the algorithm for any list of length `n`?

**Solution:**

Given a list `A1, A2, ..., An` the algorithm reverses the list and double the values of the elements. Thus the resulting list is of the form `2*An, ..., 2*A2, 2*A1`.

**Exercise 6**       (8 Marks)
### Iterative Algorithm – Lists

Write an algorithm that given a list of integers `A1, ..., An` and a number `x` prints the number of occurrences of `x` in the list. In addition, the algorithm should print the positions where `x` occurs

For example, if the list is `1, 2, 4, 1, 3` and `x` is 1 then the algorithm should print

```
1 occurs in the following positions
1, 4
The number of occurences of 1 is 2
```

If the list is `1, 2, 4, 1, 3` and `x` is `0` then the algorithm should print

```
The number of occurences of 0 is 0
```

**Solution:**

```
get n, A1, ..., An
get x
set i to 1
set occurence to 0
while (i =< n) {
    if (x = Ai)
    then set occurence to occurence + 1
    endif
    set i to i + 1
}

if (occurence = 0)
then print "The number of occurences of" + x + "is" + 0
else print x + "occurs in the following positions"
    set i to 1
    while (i =< n) {
        if (x = Ai)
        then print i
        endif
        set i to i + 1
    }
    print "The number of occurences of" + x + "is" + occurence
endif
```

**Exercise 7**                                                                    (7 Marks)
                    **Java**

Mark the correct answer in the following (**only one choice**):

1) If a program does not compile correctly, then the program suffers from

    ☑ Syntax error
    ☐ Run time error
    ☐ Logic error
    ☐ None of the above

2) What does the following Java code fragment write to the monitor?

```java
int sum = 25;
if ( sum < 20 )
    System.out.print("CSEN102 is ");
    System.out.println("great.");
```

    ☐ CSEN 102 is
    ☑ great.
    ☐ CSEN 102 is great.
    ☐ It will not print anything
    ☐ None of the above

3) What is the value of the following expression: 2 - 6 / 2 + 9

    ☐ 7
    ☑ 8
    ☐ 9
    ☐ 0
    ☐ None of the above

4) Assume that x and y are integer variables and the following nested if statement.

```java
if (x > 3) {
  if (x <= 5) y = 1;
  else if (x != 6) y = 2;
       else y = 3;
} else y = 4;
```

If the variable y has a value of 3 after executing the above program fragment, then what do you know about the variable x?

    ☐ x!=6
    ☑ x==6
    ☐ x==4
    ☐ x!=4
    ☐ x>3 && x<=4

5) What is the value of `x` after:

```
 int x = 5/2;
```

**Solution:**
Because x is an int, it will have the value 2.

6) What is the value of `x` after:

```
double x = 5/2;
```

**Solution:**
x will have the value `2.0`

7) What is the value of `x` after:

```
double x = 5/2.0;
```

**Solution:**
x will have the value `2.5`

8) What is the value of `x` after:

```
 int x = 5%2;
```

**Solution:**
`x` would have the value 1, because 5/2 is 2 remainder 1.

**Extra Sheet**

**Extra Sheet**

**Extra Sheet**