German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher

January 17, 2010

Introduction to Computer Science, Winter Term 2009-2010

Final Exam

Bar Code

Instructions: Read carefully before proceeding.

1) Duration of the exam: 3 hours (180 minutes).
2) (Non-programmable) Calculators are allowed.
3) No books or other aids are permitted for this test.
4) This exam booklet contains 14 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	8	$\sum$
Possible Marks	12	10	14	6	12	6	12	8	80
Final Marks									

Exercise 1 (12 Marks)

There is an island which is shared by foxes and geese. Each year the populations of these animals varies according to these equations:

$$\begin{aligned}\text{New fox population} &= (1-d+bG)F \\ \text{New goose population} &= (1+r-rG/k-aF)G\end{aligned}$$

The variables in these equations are:

- F = last year's fox population
- G = last year's goose population
- d = death rate of foxes
- b = conversion rate
- r = growth rate of geese
- a = ability rate
- k = carrying capacity of island

These fancy phrases probably don't mean much to you, but the equations should be enough to program the simulation.

Write an algorithm that takes as input the variables above and the total number of years and produces the following output, for example, for 100 years. Note that the new populations are calculated every year however the algorithm should print them once every 10 years.

```
---------- Year ---------- Foxes ---------- Geese
              0              100            10000
             10              215            23238
             20              542            14541
             30              607             8693
             40              518             9005
             50              502            10676
             60              535            10476
             70              539             9913
             80              529             9975
             90              527            10168
            100              531            10148

The minimum population for foxes was 100 in year 0.
The maximum population for foxes was 622 in year 26.
The minimum population for geese was 8338 in year 34.
The maximum population for geese was 23404 in year 9.
```

The algorithm should work for any number of years and should print the minimum and maximum population for foxes and geese.

Solution:

```
get F, G, d, b, r, a, k
get years
set minGeese to G
set maxGeese to G
set minFox to F
set maxFox to F
set minGYear to 0
set maxGYear to 0
set minFYear to 0
set maxFYear to 0

print "---------- Year ----------- Foxes ------------ Geese"
print "           0              " +F+ "             " +G

set i to 1
while(i <= years)
{
   set newF to (1 - d + (b * G)) * F
   set newG to (1 + r - (r * G)/k - (a* F)) * G

   if (newG < minGeese)
      set minGeese to newG
      set minGYear to i
   endif
   if (newG > maxGeese)
      set maxGeese to newG
      set maxGYear to i
   endif
   if (newF < minFox)
      set minFox to newF
      set minFYear to i
   endif
   if (newF > maxFox)
      set maxFox to newF
      set maxFYear to i
   endif
   if (i % 10 == 0)
      print "           " +i+ "             " +newF+ "             " +newG
   endif

   set F to newF
   set G to newG
   set i to i + 1
}
print "The minimum population for foxes was" +minFox+ " in year " +minFYear
print "The maximum population for foxes was" +maxFox+ " in year " +maxFYear
print "The minimum population for geese was" +minGeese+ " in year " +minGYear
print "The maximum population for geese was" +maxGeese+ " in year " +maxGYear
```

Exercise 2 (10 Marks)

Write an algorithm for finding the second-largest number in a list of n integers of the form `A0, A1, ..., An`. For example, for the input list

```
2 15 4 22 35 6
```

the algorithm should print 22.

Solution:

```
get n
get A1,...,An

if(A1 > A2)
{
   set largest1 to A1
   set largest2 to A2
}
else
{
   set largest1 to A2
   set largest2 to A1
}
set i to 3
while(i <= n)
{
   if(Ai > largest1)
   {
      set largest2 to largest1
      set largest1 to Ai
   }
   else
      if(Ai > largest2)
      {
         set largest2 to Ai
      }
   set i to i+1
}
print "The second largest number is " +largest2
```

Exercise 3 (5+3+5+1=14 Marks)

Given the following algorithm

```
get n
get A1, ..., An
get k
set i to 1
while (i<=n) {
   set j to [(i-1+k)%n]+1
   set B[j] to A[i]
   set i to i+1
}
set x to 1
while(x <= n) {
   print Bx
   set x to x + 1
}
```

a) What is the output of the algorithm for the following list and for `k` equal to 3?

`9 12 6 20 18`

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

Solution:

i	j	B[j]
1	-	-
2	4	9
3	5	12
4	1	6
5	2	20
6	3	18

Output of the algorithm: `6 20 18 9 12`

b) What is the output of the algorithm for any list of the form `A1,...,An` and for any `k`?

Solution:

It returns the list rotated of `k` positions to the right.

c) Find the total number of executed operations in worse case. Show your workout.

```
get n
get A1, ..., An
get k
set i to 1 ------------------> 1 operation --> executed once
while (i<=n) { --------------> 1 operation --> n+1 times
   set j to [(i-1+k)%n]+1 ---> 1 operation --> n times
   set B[j] to A[i] ---------> 1 operation --> n times
   set i to i+1 -------------> 1 operation --> n times
}
set x to 1 ------------------> 1 operation --> executed once
while(x <= n) { -------------> 1 operation --> n+1 times
   print Bx -----------------> 1 operation --> n times
   set x to x + 1 -----------> 1 operation --> n times
}
```

Total number of operations: $7n + 4$

d) Determine the order of magnitude of the algorithm: $\mathbf{O(n)}$

Exercise 4 $(2 + 2 + 2 = 6$ Marks)

a) Convert the binary number 110011_2 which is in two's complement to a decimal number (base 10). Show your workout.

Solution:

The number is negative. First, find the corresponding positive value:

$001101_2 = 2^3 + 2^2 + 2^0 = 13_{10}$

Thus $110011_2 = -13_{10}$

b) Convert the decimal number 121_{10} to a number in base 7. Show your workout.

Solution:

Division	Quotient	Remainder
121/7	17	2
17/7	2	3
2/7	0	2

$121_{10} = 232_7$

c) Convert the octal number 753764_8 to a hexadecimal number. Show your workout.

Solution:

7	5	3	7	6	4
111	101	011	111	110	100

The number in groups of four bits:

0011	1101	0111	1111	0100
3	D	7	F	4

Thus $753764_8 = 3D7F4_{16}$

Exercise 5 $(3 + 2 + 3 + 2 + 2 = 12$ Marks)

We would like to store the floating-point number -33.66 in a computer that uses 16 bits to represent real numbers.

a) The aim now is to find out the number of bits that will be used for the exponent and the mantissa. Assuming that the number of bits to represent the exponent will be the least number of bits needed to represent the exponent for the number -33.66. Find

- the total number of bits needed to represent the exponent and

  Solution:
  $(33)_{10} = (100001)_2$ thus the exponent should have the value 6 which needs 3 bits to be represented in binary (110). Therefore 4 bits will be needed to represent the exponent including the sign bit.

- the total number of bits to represent the mantissa (assuming that we have in total 16 bits to represent real numbers)

  Solution:
  $16 - 4 = 12$ remaining bits to represent the mantissa including the sign bit.

b) Give the largest number in binary that can be represented using the number of bits of the mantissa and the exponent from part a).

1111111.1111

c) Show the binary representation of the decimal number -33.66.

Solution:

$-33.66_{10} = -100001.10101_2$

$$\begin{aligned} 0.66 * 2 &= \underline{1}.32 \\ 0.32 * 2 &= \underline{0}.64 \\ 0.64 * 2 &= \underline{1}.28 \\ 0.28 * 2 &= \underline{0}.56 \\ 0.56 * 2 &= \underline{1}.12 \end{aligned}$$

d) Show the binary number in normalized scientific notation.

Solution:

$-33.66_{10} = -0.10000110101 \times 2^6$

e) Show how the binary number will be stored in the 16 bits below.

Solution:

1	10000110101	0	110
Sign of mantissa 1 bit	Mantissa x bits	Sign of exponent 1 bit	Exponent y bits

Exercise 6 (6 Marks)

Assume that our computer stores decimal numbers using 6 bits. Perform the subtraction

$$(-27)_{10} - (25)_{10}$$

using 2's complement notation. Give the result of the subtraction in decimal. Show your workout, i.e. all steps performed.

Solution:

- Convert 27 and 25 to binary:

  $27_{10} = 011011_2$
  $25_{10} = 011001_2$

- Two's complement representation of -27

  $-27_{10} = 100101$

- Two's complement representation of -25

  $-25_{10} = 100111$

- Perform the addition $(-27)_{10} + (-25)_{10}$ in binary:

  $100101 + 100111 = 1001100$

- Remove the overflow:

  01100

- The binary number 01100 represents the positive decimal value 12.

Exercise 7 (3+3+4+2=12 Marks)

Given the following the following truth table, where **A**, **B** are the input variables and **X**, **Y** and **Z** are the output variables.

A	B	X	Y	Z
0	0	0	1	0
0	1	0	1	1
1	0	1	0	0
1	1	1	0	1

a) Use the sum-of-products algorithm to find the Boolean expressions that describe the output of the truth table.

Solution:

$$X = AB' + AB$$

$$Y = A'B' + A'B$$

$$Z = A'B + AB$$

b) What is the functionality of the circuit?

Solution:

The circuit computes the operation $x + 2$, where x consists of two bits.

c) Draw the Boolean circuit. **Note** that each gate can have only two inputs.

d) How many half-adders will be needed to perform the task defined by the truth table above. Justify your answer by drawing the circuit using only half-adders.

Solution:

1 half adder will be needed.

Exercise 8 (8 Marks)

Given the following Boolean expression

$$((A+B)(B'+C'+D'))+B'C'(A+B'+C)+A'C+D$$

Simplify the Boolean expressions using the Boolean algebra. Please mention the applied rules.

$x+0=x$	$x*1=x$	
$x+1=1$	$x*0=0$	
$x+x=x$	$x*x=x$	
$x+x'=1$	$x*x'=0$	
$(x')'=x$		
$x+y=y+x$	$xy=yx$	Commutativity
$x+(y+z)=(x+y)+z$	$x(yz)=(xy)z$	Associativity
$x(y+z)=xy+xz$	$x+yz=(x+y)(x+z)$	Distributivity
$(x+y)'=x'y'$	$(xy)'=x'+y'$	DeMorgan's Law

Hint: The circuit of the simplified expression consists of zero gates.

Solution:

$$\begin{array}{rcl}
= & AB'+AC'+AD'+BB'+BC'+BD'+AB'C'+B'B'C'+B'C'C+A'C+D & (Distributivity)\\
= & AB'+AC'+BB'+BC'+BD'+AB'C'+B'C'+B'C'C+A'C+D+AD' & (Associativity)\\
= & AB'+AC'+BB'+BC'+BD'+AB'C'+B'C'+B'C'C+A'C+(D+A)(D+D') & (Distributivity)\\
= & AB'+AC'+BB'+BC'+BD'+AB'C'+B'C'+B'C'C+A'C+(D+A)(1) & (x+x'=1)\\
= & AB'+AC'+BB'+BC'+BD'+AB'C'+B'C'+B'C'C+A'C+D+A & (x*1=x)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+A'C+D+A+BC'+B'C' & (Associativity)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+A'C+D+A+C'(B+B') & (Distributivity)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+A'C+D+A+C'(1) & (x+x'=1)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+A'C+D+A+C' & (x*1=x)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+A+C'+A'C & (Associativity)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+A+(C'+A')(C'+C) & (Distributivity)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+A+(C'+A')(1) & (x+x'=1)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+A+C'+A' & (x*1=x)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+C'+(A+A') & (Associativity)\\
= & AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+C'+1 & (x+x'=1)\\
= & (AB'+AC'+BB'+BD'+AB'C'+B'C'C+D+C')+1 & (x+1=1)\\
= & 1 &
\end{array}$$

Extra Page

Extra Page

Extra Page