NTNU
Det skapende universitet

Department of Computer and Information Science

Examination paper - TDT4110 Information Technology, Introduction

Academic contact during the exam: **Björn Gambäck**

Phone: **+46 70 568 1535**

Examination date: **Aug 11, 2014**

Examination time (from-to): **09:00 – 13:00**

Permitted examination support material: **Approved calculator**

Other information:

The exam contains four exercises. A percentage is given to show how much every exercise and sub-problem counts when the exam is graded. Read through all the exercises before you start solving them. Be smart and make good use of your time! If you feel that the problems are not fully specified, please write your assumptions explicitly.

Answer briefly and clearly, and write the text in an easy to read manner. If your text is ambiguous or longer than necessary, points might be subtracted.

Language: **English**

Number of pages: **15 (incl. cover page)**

Contents:

Checked by:

Aug 1, 2014 Guttorm Signe

Date Sign.

Exercise 1: Multiple-Choice (25 %)

Use the enclosed forms to solve this exercise (you may keep one). You can get a new form if you need. **Only one answer is completely correct.** For each question, a correct answer counts as 1 point. Wrong answer or more than one answer gives -½ point. No answer gives 0 points.
You will not get less than 0 points in total for this exercise.

1) Which Internet service/application normally does NOT require low latency?

a) On-demand Video.
b) Interactive audio and video.
c) Online gaming.
d) IP telephony.

2) Which of the following is NOT part of the waterfall model?

a) Requirements analysis and definition
b) System and software testing
c) User interface evaluation
d) Integration and system testing

3) For a problem of size n, there are four algorithms with different time complexity. Which one will consume most time (in the worst case) on large problems?

a) O(1) (constant time)
b) O(n) (linear time)
c) $O(n^2)$ (quadratic time)
d) $O(2^n)$ (exponential time)

4) According to the Nyquist theorem, the sampling rate for audio is

a) half of the frequencies a human can hear.
b) the same as the frequencies a human can hear.
c) twice the frequencies a human can hear.
d) 3 times the frequencies a human can hear.

5) What is a protocol?

a) A definition of what computer communication is.
b) A set of rules that makes it possible for two computers to communicate.
c) Rules that determine the syntax of a programming language.
d) A definition of how TCP/IP works.

6) The correct order from smallest to largest is

a) giga, kilo, mega, tera
b) kilo, mega, giga, tera
c) tera, kilo, mega giga
d) kilo, mega, tera, giga

7) Given the following function:

```
function(n):
if n < 0:
 return n + function(n+1)
else if n> 0:
 return n + function(n-1)
else:
 return 0
```

What do we call the code sequence that is executed if `n==0`?

a) recursive case
b) iterative case
c) base case
d) return case

8) ALU is used in

a) Instruction Fetch
b) Instruction Execution
c) Result Return
d) Instruction Decode

9) Which of the following abbreviations is a well-known system development process model?

a) SCRUP
b) UTML
c) RUP
d) RAM

10) A digital-to-analogue converter (DAC)

a) Changes digital information to analogue
b) Converts continuous sound to digital sound
c) Converts sound to an electric signal
d) Provides approximate values

11) Which of the following is NOT a high-level language?

a) Java
b) C
c) Assembly
d) Visual Basic

12) When is sequential search efficient?

a) When the data is sorted
b) When the data is numeric
c) Sequential search is never efficient
d) When the sought item is early in the data

13) The song «A little bit» is 3 minutes and 47 seconds. How many bits are needed to store it in stereo on a standard music CD?

a) 1 411 200
b) 40 042 800
c) 84 672 000
d) 320 342 400

14) What is VPN?

a) A method for translating logical names to an IP-number
b) Part of the TCP/IP specification
c) A way to establish a safe / encrypted connection between two computers
d) A method an Internet Service Provider uses to distribute sensitive content

15) A router is a

a) a computer connecting several networks
b) a program that sends information packets between two computers
c) a program that merges message information packets before delivering them to the recipient
d) a computer connected to the Internet

16) Given the list of names «Alice, Byron, Carol, Duane, Elaine, Floyd, Gene, Henry, Iris». Which search algorithm will find Carol first (perform fewest comparisons)?

a) Binary search
b) Sequential search
c) Both will find Carol just as quickly
d) The answer depends on how the binary search algorithm is implemented

17) A byte of memory can store

a) one of 1024 different numbers
b) one word
c) one ASCII character
d) one block

18) How many steps are there in the Fetch/Execute cycle?

a) 3
b) 4
c) 5
d) 6

19) The Program Counter is changed directly by instructions called

a) Add and Multiply
b) Branch and Jump
c) Input and Output
d) Now and Next

20) How is the hexadecimal number A8 represented in binary?

a) 10101000
b) 10010100
c) 11001000
d) 10001100

21) In the course books, the abbreviation DDOS is short for…
   a) Digital Disk Operating System
   b) Double Density Optical Storage
   c) Distributed Denial Of Service
   d) Data Directory On Site

22) The main difference between Boehm's spiral model for software development and other, earlier process models was …
   a) explicit focus on risk analysis and management
   b) explicit focus on software company working environments
   c) explicit focus on gradual development of project team competence
   d) explicit focus on gradual development of business management competence

23) Person A wants to send a confidential message to person B. What kind of encryption keys must be used in this case?
   a) A encrypts with A's private key, B decrypts with A's public key
   b) A encrypts with B's private key, B decrypts with A's public key
   c) A encrypts with B's public key, B decrypts with B's private key
   d) A encrypts with B's public key, B decrypts with A's public key

24) A firewall is a type of security technology. What is the most accurate and relevant claim when it comes to firewalls and threats from so-called Trojan Horses?
   a) Firewalls can protect against Trojan Horses by preventing unexpected Internet traffic from the outside into a system.
   b) Firewalls can protect against Trojan Horses by preventing unexpected traffic from the inside out to the Internet.
   c) Firewalls can protect against Trojan Horses by warning users not to open fake e-mail messages.
   d) Firewalls provide NO protection against Trojan Horses. Only antivirus software is effective against Trojan Horses.

25) What is an important difference between system testing and acceptance testing?
   a) System testing focuses on finding errors in a program, while acceptance testing focuses on the parts that work.
   b) System testing tends to use fabricated test data while acceptance testing uses data from the customer that will use the system.
   c) System testing only tests the system module by module, while acceptance testing tests the entire system as a whole.
   d) System testing can be carried out by staff who do not know programming, while acceptance testing must be carried out by programmers in order to correct unacceptable errors.

Exercise 2: Basic Programming (20 %)

The purpose of the game Sudoku is to fill out 9 rows, columns and squares with all numbers from 1 to 9. See the figure on the right that shows a Sudoku board with three examples of a correctly filled column, row and square.

Example of a correctly filled column: →

8
9
1
2
3
4
5
6
7

Example of a correctly filled row:

4	5	6	1	2	3	9	8	7

Example of a correctly filled square:

1	3	5
2	4	6
9	8	7

8								
9								
1								
2						1	3	5
3						2	4	6
4	5	6	1	2	3	9	8	7
5								
6								
7								

The game is finished when all 9 columns, rows and squares have been correctly filled.
We may use the number 0 to represent a place that has not yet been filled by a number.
For example, in

2	0	0	0	0	0	1	3	5

the numbers 4, 6, 7, 8, 9 are still missing (in columns 2-6).

Exercise 2a) (4 %)

Write a simple function `readOneNumber()` that reads row, column and a number between 1 and 9. The function shall print out a nicely formatted confirmation to the user. You do not need to return the numbers from this function nor to check whether the user input is correct.
An example of how function execution should look like (grey print shows programme output, as the user enters the three numbers, with an <Enter> after each number):

```
>>> readOneNumber()
Row (1-9): 2
Column (1-9): 3
Number (1-9): 4
Position (2,3) now contains 4
```

Exercise 2b (4 %)

Create a completely new input function called `readPositionDigit`.
This function shall have the input parameters «rowNr», «colNr» and «board».
«board» contains all the numbers on the board (see the figure below), and the function shall ask the user to enter a new value for the address (rowNr, colNr) on the keyboard. The new value shall be saved in «board», in the row «rowNr» and the column «colNr». Afterwards, «board» shall be returned from the function. You can assume that the user enters correct input values in this exercise.
Example usage:

```
>>> readPositionDigit(2,3,[[1,0,0],[2,0,0],[3,0,0]])
Value for position (2,3): 8
[[1, 0, 0], [2, 0, 8], [3, 0, 0]]
```

1	0	0…
2	0	8…
3	0	0…
…	…	…

Exercise 2c (6 %)

It's important to assure that the board only contains numbers from 0-9, and not for example «abc». Write a new, improved complete input function «readValidPositionDigit». The function shall take the same input and output as above, but this time it should also include error handling: If the user enters anything other than a number from 0-9, the function shall print «Error! Please give a number between 0 and 9…» and continue to ask for a new input value until the user enters a number between 0-9. Example usage:

```
>>> readValidPositionDigit(2,3,[[1,0,0],[2,0,0],[3,0,0]])
Value for position (2,3): abc
Error! Please give a number between 0 and 9…
Value for position (2,3): [1,2,3]
Error! Please give a number between 0 and 9...
Value for position (2,3): 'a'
Error! Please give a number between 0 and 9...
Value for position (2,3): 10
Error! Please give a number between 0 and 9...
Value for position (2,3): 9
[[1, 0, 0], [2, 0, 9], [3, 0, 0]]
```

1	0	0…
2	0	9…
3	0	0…
…	…	…

Exercise 2d (6 %)

In order to solve Sudoku (for example in a newspaper), we may use the function «readValidPositionDigit» both to read a partially filled board and to fill out new single digits. We must also check so that each number doesn't occur more than once in each row, column or square, but you may ignore that in this exercise.

Write a function «**readSudokuBoard**» that uses the function «**readValidPositionDigit**» to fill out «**board**» with correct numbers from 0 to 9. It shall read all the 9 numbers in column 1 from top to bottom, before continuing with the remaining columns (2-9) from left to right in the same fashion, one column at a time. The function shall output «**board**» with 81 numbers (0-9). Example usage:

```
>>> readSudokuBoard()
Value for position (1,1): 1
Value for position (1,2): 2
…
Value for position (9,8): 6
Value for position (9,9): 7
[[1, 2, 3, 4, 5, 6, 7, 8, 9], [2, 3, 4, 5, 6, 7, 8, 9, 1],  [3, 4, 5, 6, 7, 8, 9, 1, 2],  ...
 [4, 5, 6, 7, 8, 9, 0, 1, 2],  [3, 4, 5, 6, 7, 8, 9, 0, 1],  [2, 3, 4, 5, 6, 7, 8, 9, 0],  ...
 [1, 2, 3, 4, 5, 6, 7, 8, 9],  [0, 1, 2, 3, 4, 5, 6, 7, 8],  [9, 0, 1, 2, 3, 4, 5, 6, 7]]
```

Exercise 3 – Program Comprehension (15 %)

Exercise 3 a) (5 %)

1. What will be the value of `res` after executing the command **`>>> res= o3a(6)`** with the function shown below,
2. Explain in one short sentence what the function does.

```
def o3a(n):
    if n == 0:
        a = 0
    elif n == 1:
        a = 1
    else:
        m2 = 0
        m1 = 1
        for i in range(1,n):
            a = m1 + m2
            m2 = m1
            m1 = a
    return a
```

Exercise 3 b) (5 %)

3. What will be the value of `res` after executing **`>>> res = o3b(6)`** with the function below?
4. Explain in one short sentence what the function does.

```
def o3b( n ):
    if n == 0:
        a = 1
    else:
        a = n * o3b( n-1 )
    return a
```

Exercise 3 c) (5 %)

We want a function "**checkRowOK**" that shall test whether a sudoku row with 9 numbers uses all of the number 1-9 exactly once, in line with the Sudoku rules. Example of expected program output:

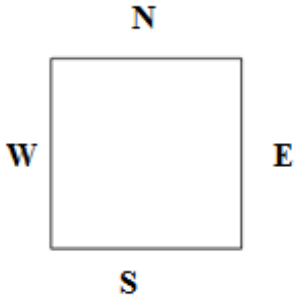
```
>>> checkRowOK ( [ 1,2,3,4,5,6,9,8,7 ] )
OK.
>>> checkRowOK ( [ 1,2,3,4,5,2,9,8,7 ] )
The number 2 occurs 2 times. The number 6 is not used.
```

"Per" has written the following function:

```
def checkRowOK (row):
    timesUsed = [0 for x in range(9)]
    for n in row:
        timesUsed[n-1] += 1
    response = ""
    for i in range(9):
        if (timesUsed[i] != 1):
            response += "The number " + str(i+1)
            if (timesUsed[i] == 0):
                response += " is not used. "
            else:
                response += " occurs " + str(timesUsed[i]) + " times. "
        else:
            response = "OK."
    print(response)
```

This gives correct output for the first example above (OK), but also gives OK for the second example, due to an error in the function. What is the error, and how can it be corrected?
(You don't need to write any code in this exercise, only explain in general how to fix the error.)

Exercise 4: More programming (40 %)



Bridge is a card game played by two two-person teams. The teams are called North/South ('N/S') and East/West ('E/W'). A regular deck of cards is used, with 52 cards, so that each player is dealt 13 cards. The players bid how many tricks they count on winning and on which suit shall be trump. The bids go from **1 clubs** to **7 notrump**. The bidder is supposed to win 6 tricks more (7-13) than the number he/she bids and the text behind the number indicates the trump suit (clubs, diamonds, hearts, spades, or possibly notrump). If the last bid is in "notrump", there will be no trump suit.
The bid "3 clubs" thus means that the bidder should win 9 tricks with clubs as the trump suit.
The final (highest) bid in the bidding auction is called the **contract.** If one wins more tricks than claimed in the contract (e.g., 9 tricks after the bid "3 diamonds" which only requires 8 tricks), the extra tricks are called **overtricks.** If one fails to win the number of tricks claimed in the contract, the contract has been **set** (defeated).

NB: You can/may reuse functions created in the first sub-exercises in the latter ones, even if you haven't been able to fully solve the first ones. You may assume that the user always enters legal input values.

Exercise 4a (5 %)

Write a function with two input parameters, «bid» and «number of tricks won», and a return value which is «True» in case the bid contract has been made, or «False» otherwise. Example: bidOk('3 diamonds', 10) => True, bidOK('3 diamonds', 8) => False.

Exercise 4b (10 %)

Some contracts give bonus points, e.g., «**game**». A **game** means that one has bid and made at least 3 notrump (at least 9 tricks, with no trump suit), 4 hearts/spades or more (at least 10 tricks with hearts or spades as trump suit), or 5 clubs/diamonds or more (at least 11 tricks with clubs or diamonds as trump suit). Write a function which takes the result of a deal as input (bid and number of tricks won) and determines whether this was a game contract, and whether the team has made the game. Return True if a game has succesfully been made, otherwise return False.

Exercise 4c (15 %)

The scoring in Bridge can be pretty complicated. Following is a simplified description:
No points are awarded for the first six tricks in Bridge. All tricks after the sixth are called «odd tricks», so that seven tricks constitute one «odd trick». The highest possible bid is for 7 odd tricks, which is the same as winning all 13 tricks.

- Playing with clubs or diamonds as trump suit gives 20 points per won odd trick (1-7).
- Playing with hearts or spades as trump suit gives 30 points per won odd trick (1-7).
- Playing with no trump suit gives 30 points per won odd trick (1-7) plus 10 extra points.

A 50 point bonus is awarded if the bid contract is made. However, if the contract was a game (at least 5 clubs or diamonds, 4 spades or hearts, or 3 notrump), a 300 point bonus is awarded instead. If the contract is set (defeated), 0 points are awarded, but the opponents get 50 points for each trick that is missing.

Write (two) functions that together counts the sum of points for a deal (play round). Note that points are awarded for odd tricks and bonuses, or points given to the opponents in case the contract is set.

If the team that plays the contract makes it, the function shall return a positive number of points. However, if they fail to make the contract, a negative number of points shall be returned. Examples:

```
>>> bridgePoints( '3 diamonds', 10)  returns 130        (4 * 20 + 50)
>>> bridgePoints( '3 diamonds', 8)   returns  -50       (-50 * 1)
>>> bridgePoints( '3 spades', 12)    returns  230       (6 * 30 + 50)
>>> bridgePoints( '4 spades', 12)    returns  480       (6 * 30 + 300)
>>> bridgePoints( '4 notrump', 12)   returns  490       (10 + 6 * 30 + 300)
```

Exercise 4d (10 %)

Write a program that registers more than one deal (round of play). Use the functions that you have written in exercises 4 a, b and c (and possibly write other/more functions in case you need to). Each deal shall be stored as a list containing which team played the contract, the bid, number of tricks won, number of points, and number of points given to the opponents (if the contract was set).
E.g.: ['N/S', '3 diamonds', 9, 110, 0]
The total summary of all deals will thus be a list of lists, as follows:

```
[['N/S', '3 diamonds', 9, 110, 0],
 ['E/W', '3 hearts', 9, 140, 0],
 ['N/S', '4 spades', 8, 0, 100],
…]
```

When all deals have been registred, the program shall count the total number of points for each team (N/S and E/W).
Example of program output:

```
Team (N/S or E/W, or anything else to finish): N/S
Bid: 4 notrump
Tricks: 9
Team (N/S or E/W, or anything else to finish): N/S
Bid: 4 notrump
Tricks: 10
Team (N/S or E/W, or anything else to finish): E/W
Bid: 5 diamonds
Tricks: 11
Team (N/S or E/W, or anything else to finish):
['N/S', '4 notrump', 9, 0, 50]
['N/S', '4 notrump', 10, 430, 0]
['E/W', '5 diamonds', 11, 400, 0]
Total score:
N/S 430
E/W 450
```

Appendix: Useful Python functions/methods

Built-in:

format(numeric_value, format_specifier)

Formats a numeric value into a string according to the format specifier, which is a string that contains special characters specifying how the numeric value should be formatted. Examples of various formatting characters are "f=floating-point, e=scientific notation, %=percentage, d=integer". A number before the formatting character will specify the field width. A number after the character "." will format the number of decimals.

%

Reminder: Divides one number by another and gives the remainder

len(s)

Return the length (the number of items) of a string, tuple, list, dictionary or other data structure.

int(x)

Convert a string or number to a plain integer.

float(x)

Convert a string or a number to floating point.

str([object])

Return a string containing a nicely printable representation of an object.

pow(x, y)

Return x to the power y (x**y or x^y)

Library: math

math.pow(x,y)

Return x to the power y (x**y or x^y)

math.sqrt(x)

Return the square root of x.

math.pi

The mathematical constant π = 3.141592...

math.e

The mathematical constant e = 2.718281...

Library: random

random.randint(a, b)

Return a random integer N such that a <= N <= b.

random.random()

Return the next random floating point number in the range 0 <= N < 1.

String methods:

s.isalnum()

Returns true if the string contains only alphabetic letters or digits and is ate least one character of length. Returns false otherwise.

s.isalpha()

Returns true if the string contains only alphabetic letters, and is at least one character in length; otherwise false.

s.isdigit()

Returns true if the string contains only numeric digits and is at least one character in length; otherwise false.

s.isspace()

Returns true if the string contains only whitespace characters, and is at least one character in length; otherwise false. (Whitespace characters are spaces, newlines (\n), and tabs (\t).)

s.ljust(width)

Return the string left justified in a string of length width.

s.rjust(width)

Return the string right justified in a string of length width.

s.lower()

Returns a copy of the string with all alphabetic letters converted to lowercase.

s.upper()

Returns a copy of the string with all alphabetic letters converted to uppercase.

s.find(substring)

Returns the lowest index where the substring is found. If the substring is not found it returns -1

s.split(char)

Splits a text into single elements and uses char as dividing character. Returns a list with the elements.

List operations:

s[i:j]

Return slice starting at position i extending to position j. Can also be used for strings.

item in s

Determine whether a specified item is contained in a list.

min(list)

Returns the item that has the lowest value in the sequence.

max(list)

Returns the item that has the highest value in the sequence.

s.append(x)

Append new element x to end of s

s.insert(index,item)

Insert an item into a list at a specified position given by an index)

s.index(*item*)

Return the index of the first element in the list containing the specified item.

s.pop()

Return last element and remove it from the list

s.pop(i)

Return element i and remove it from the list

s.remove(item)

Removes the first element containing the item.

s.reverse()

Reverses the order of the items in a list.

s.sort()

Rearranges the elements of a list so they appear in ascending order.

Dictionary operations:

clear()

Clears the contents of a dictionary

get(key, default)

Gets the value associated with a specific key. If the key is not found, the method does not raise an exception. Instead, it returns a default value.

items()

Returns all the keys in a dictionary and their associated values as a sequence of tuples.

keys()

Returns all the keys in a dictionary as a sequence of tuples.

pop(key, default)

Returns the value associated with a specific key and removes that key-value pair from the dictionary. If the key is not found, the method returns a default value.1

popitem()

Returns a randomly selected key-value pair as a tuple from the dictionary and removes that key-value pair from the dictionary.

values()

Returns all the values in dictionary as a sequence of tuples.

Form for Multiple-Choice questions

Candidate no: _____ Program: _____

Course code: _____ Date: _____

No. of pages: _____ Page: _____

Exercise nr	***A***	***B***	***C***	***D***
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				
1.21				
1.22				
1.23				
1.24				
1.25				

Blank back-page (for double-sided printing)

Form for Multiple-Choice questions

Candidate no: _____ Program: _____

Course code: _____ Date: _____

No. of pages: _____ Page: _____

Exercise nr.	***A***	***B***	***C***	***D***
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				
1.21				
1.22				
1.23				
1.24				
1.25				