BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Bachelor Degree Written Exam, September 4, 2019
Computer Science – English

VARIANT 1

REMARKS

- All subjects are compulsory and full solutions are requested.
- The minimum passing grade is 5,00.
- The working time is 3 hours.

SUBJECT Operating Systems

1 Answer the following questions, considering that the program below includes all required headers, and that all the instructions are executed successfully.

```
1  int n=0, a[2], b[2];
2  void w(int p[2], char c) {
3    n++;
4    if(fork() == 0) {
5      close(p[0]); write(p[1], &c, 1); close(p[1]);
6      exit(0);
7    }
8  }
9  void r(int p[2]) {
10   char c;
11   n++;
12   if(fork() == 0) {
13     close(p[1]);
14     if(read(p[0], &c, 1) > 0) {printf("%c\n", c);}
15     close(p[0]);
16     exit(0);
17   }
18 }
19 int main(int argc, char** argv) {
20   pipe(a); pipe(b);
21   w(a, 'x'); w(a, 'y'); r(a); r(a);
22   close(a[0]);close(a[1]);close(b[0]);close(b[1]);
23   for(int i=0; i<n; i++) {wait(0);}
24   printf("%d\n", n);
25   return 0;
26 }
```

a) Draw the process hierarchy diagram, including the parent process as well.

b) What will the program execution print?

c) What will the program execution print, if line 21 is replaced with the code below?

```
w(a, 'x'); w(b, 'y'); r(a); r(b);
```

d) What will the program execution print, if line 21 is replaced with the code below?

```
w(a, 'x'); w(a, 'y'); r(a); r(b);
```

e) Explain and justify the program functioning, if line 21 is replaced as in question d) above, and additionally, line 22 is removed.

2 Answer the following questions about the UNIX Shell script below.

```
1  #!/bin/bash
2  for F in *.sh; do
3    A=`grep "^[ \t]*[^ \t#]" $F | wc -l`
4    B=`wc -l < $F`
5    if [ $A -lt `expr $B - $A` ]; then
6      echo $F
7    fi
8  done
```

a) What will the script execution print?

b) What will the script execution print, if it is executed in a directory containing only the script itself?

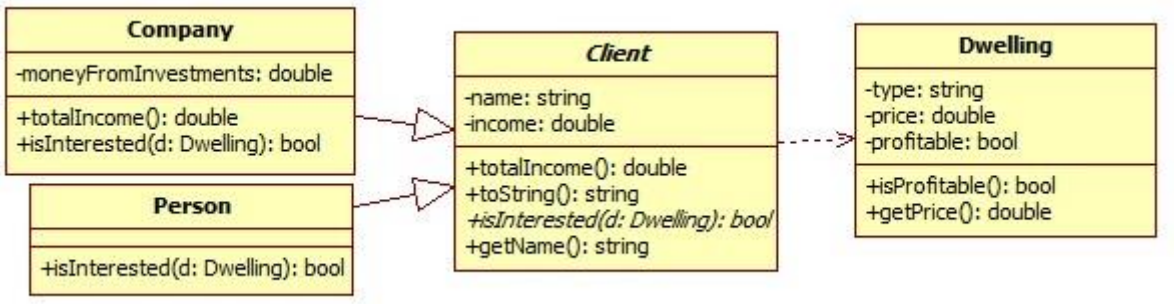
c) Explain in detail the regular expression on line 3.

VARIANT 1

SUBJECT Algorithms and Programming

Write a program in one of the programming languages Python, C++, Java, C#, with the following requirements:

a) Define the classes **Client**, **Person**, **Company**, **Dwelling** according to the following UML diagram (constructors are not shown on the diagram). Only the methods indicated at b) must be implemented, the others should only be declared.



- The attribute *name* in class **Client** (the client's name) must contain at least 3 characters and the *income* must be strictly positive. The constructors will enforce the constraints.
- The abstract class **Client** has an abstract method **isInterested**.
- The total income of a company is the income (*income*), to which the money from investments (*moneyFromInvestments*) is added. Any other client's total income is equal to her/his income.
- A person is interested in a dwelling if the dwelling's price divided by 360 is less than half the person's total income. A company is interested in a dwelling if the dwelling's price divided by 12 is less than the company's total income and if the dwelling is profitable.
- The method **toString** returns the name concatenated to the total income, for both persons and companies.

b) Implement the following methods from the diagram at a): constructors for classes **Client**, **Person**, **Company**, **Dwelling**; methods **totalIncome** from classes **Client** and **Company** and methods **isInterested** from classes **Person** and **Company**.

c) Define a function that has as parameters a list of clients and an order relation between clients (given as a boolean function that compares two clients) and sorts the list of clients according to the given relation. Sorting must be implemented using an algorithm of time complexity $\theta(n \cdot log_2 n)$.

d) Define a function that receives as parameters a dictionary of clients (key - client name, value - client) and a dwelling and returns, using the function at c), a list with all clients that are interested in the given dwelling, sorted alphabetically, by name.

e) The main function of the program creates a dictionary of clients and adds the following clients (choose values for the unspecified attributes): two companies, one named "TwoStar" and another named "Fort" and two persons, one named "Ana" and the other named "Mihai". Build two dwellings: one's type is "apartment", it costs 150000 and is not profitable and the other's type is "house", it costs 500000 and it is profitable. For the dictionary of clients and for each dwelling, call the function at d). Print the clients in the returned lists, using the method **toString**.

f) For the **Dictionary** data type used in the program write the specifications of the operations **add** and **search**.

- **Please indicate the used programming language.**
- **Do not define other methods than those shown in the diagram (except for the constructors).**
- **Do not use sorted containers and predefined sorting operations.**

*You may use existing libraries for **data structures** (Python, C++, Java, C#).*

VARIANT 1

SUBJECT Databases

Consider a database that stores the data managed by an electronic voting application for a referendum. The database has the following structure:

- table ***Persons*** with fields **PId**, **Name**, **CNP**, **CityId**, **CountyId**, **RegionId, Voted**;
- table ***Votes*** with fields **VId**, **Date, Time**;
- table ***Answers*** with fields **VId**, **QId**, **YesAnswer, NoAnswer, Cancelled**;
- table ***Questions*** with fields **QId**, **Text, CategoryName**, **CategoryDescription**.

An answer for a question is cancelled if neither *Yes* nor *No* has been selected or both options have been selected. Fields **YesAnswer**, **NoAnswer, Cancelled** and **Voted** can only take on values 0 or 1.

1. Determine the candidate keys and the foreign keys for each of the above tables.
2. Determine at least 4 functional dependencies that refer to fields that don't represent candidate keys.
3. Consider the following structure changes and specify which of them are essential for the database to be in **3NF** (the 3rd normal form). In the case of an affirmative answer, justify your choice.
   a. Creating a separate table to store categories.
   b. Eliminating the **Cancelled** field from the ***Answers*** table.
   c. Using a single field in the ***Votes*** table that stores both the date and the time of the vote.
   d. Adding a field **PId** in the ***Votes*** table that refers to records in the ***Persons*** table.
4. On the given structure, write an SQL query that is equivalent to the following query:

$$\Pi_{Text,\ Time}\ (\ \sigma_{Time\ >18:00}\ (\ \textbf{\textit{Votes}}\ \otimes_{Votes.VId=Answers.VId}\ \textbf{\textit{Answers}}\ \otimes_{Questions.QId=Answers.QId}\ \sigma_{CategoryName='EU'}\ (\textbf{\textit{Questions}})))$$

5. On the given structure, write an SQL query that returns, for each question, the total number of answers, the total number of valid *Yes* answers, the total number of valid *No* answers and the total number of cancelled answers (***Text, NumberAnswers, NumberYes, NumberNo, NumberCancelled***).

BAREM INFORMATICĂ
VARIANTA 1

Subiect (Algoritmică şi Programare)

Oficiu – 1p
Definirea clasei abstracte Client – 0.6p din care
- atribute – 0.1
- constructor si metode – 0.5

Definirea clasei Company– 1.5p din care
- relația de moștenire – 0.2
- atribut – 0.1
- constructor – 0.4
- metode – 0.8

Definirea clasei Person – 0.7p din care
- relația de moștenire – 0.2
- constructor – 0.2
- metoda – 0.3

Definirea clasei Dwelling – 0.2p din care
- atribute– 0.1
- constructor - 0.1

Funcția de la punctul c) – 3p din care
- signatura corectă - 0.4p
- sortare in $\theta(n \cdot log_2 n)$ - 2.5p
- returnare listă rezultat – 0.1

Funcția de la punctul d) – 1.5p din care
- signatura corectă - 0.1p
- construire lista clienti in ordine alfabetica dupa nume – 1.3p
- returnare listă rezultat – 0.1

Funcția principală e) – 0.5p
f) Specificaţiile operaţiilor **adăugare** și **căutare** pentru tipul de dată **Dicționar** – 1p

Subiect Baze de date

1. 0.5p (chei candidat) + 0.5p (chei externe) = 1p
2. 0.25p x 4 = 1p
3. a, b
   - 2 x (0.5p răspuns + 0.5p justificare) = 2p
4. rezolvarea completă a interogării = 2p
5. rezolvarea completă a interogării = 3p

1p of

Subiect Sisteme de operare

Oficiu – 1p

1.a Diagramă cu procesul părinte având patru procese fiu – 1p
1.b Va afișa “x y 4” sau “y x 4” – 1p
1.c Identic cu punctul 3.1.b – 1p
1.d Va afișa “x 4” sau “y 4” – 1p
1.e Va afișa “x” sau “y” și se va bloca la linia 14 pentru că pipe-ul este deschis pentru scriere – 1.5p

2.a Numele scripturilor Shell din directorul curent cu mai puține linii de cod decât restul liniilor din script – 1p
2.b Nu va afișa nimic – 1p
2.c Inceput de linie, 0 sau mai multe spații sau taburi, orice caracter care nu e spațiu, tab sau diez – 1.5p