

**Schriftliche Abschlussprüfung, 3. JULI 2023**  
**Fachgebiet Informatik in deutscher Sprache**

**VARIANTE 1**

**BEMERKUNG.**

- Alle Prüfungsthemen sind verpflichtend. Bei allen Prüfungsthemen müssen vollständige Lösungen angegeben werden.
- Die Mindestnote für das Bestehen der Abschlussprüfung ist 5,00.
- Die Arbeitszeit beträgt 3 Stunden.

**TEIL Algorithmen und Programmierung**

Wichtiger Hinweis:

- Das Fehlen eines richtigen Programmierstils (sinvolle Namen für Variablen, Einrückung des Codes, ggf. Kommentare, Lesbarkeit des Codes) führt zum Verlust von 10 % der entsprechenden Punktzahl.
- Für Pseudocode sind nur die folgenden Anweisungen erlaubt: *Zuweisung*, *For-Schleife*, *Wenn*, *Do-While-Schleife*, *While-Schleife*, *Ausgeben* und *Zurückgeben*.
- Man darf keine weiteren Attribute, Methoden definieren, außer Konstruktoren, ggf. Destruktoren und was in der Aufgabenbeschreibung angegeben ist. Die Sichtbarkeit der in der Aufgabenbeschreibung angegebenen Attributen darf nicht verändert werden.

**Aufgabe 1. (1.5 Punkte)**

Gegeben seien die unten in Pseudocode beschriebenen Algorithmen. Der Algorithmus **transformare** hat den Vektor  $x$  bestehend aus  $n$  Ganzzahlen ( $x[1], \dots, x[n]$ ) und den Vektor  $y$  bestehend aus  $m$  Ganzzahlen ( $y[1], \dots, y[m]$ ) als Eingabeparameter, und den Vektor  $z$  bestehend aus  $k$  Ganzzahlen ( $z[1], \dots, z[k]$ ) als Ausgabeparameter. Der Algorithmus **g** hat den Vektor  $x$  bestehend aus  $n$  Ganzzahlen ( $x[1], \dots, x[n]$ ) und eine natürliche Zahl  $i$  ( $1 \leq i \leq n$ ) als Eingabeparameter, und den Vektor  $y$  bestehend aus  $n$  Ganzzahlen ( $y[1], \dots, y[n]$ ) als Ausgabeparameter. Was wird der Vektor  $z$  nach dem Aufruf **transformare**( $x, n, y, 0, z, k$ ) beinhalten? Begründen Sie Ihre Antwort.

```
Algorithmus g(x, n, y, i)
  Wenn i ≤ n dann
    y[i] ← x[i]
    g(x, n, y, i + 1)
  Wenn_Ende
Algorithmus_Ende

Algorithmus transformare(x, n, y, m, z, k)
  Wenn n = 0 dann
    g(y, m, z, 1)
    k ← m
  sonst
    y[m + 1] ← x[n]
    transformare(x, n - 1, y, m + 1, z, k)
  Wenn_Ende
Algorithmus_Ende
```

**Aufgabe 2. (3 Punkte)**

Gegeben sei ein Vektor  $a[1], \dots, a[n]$  ( $4 \leq n \leq 10000$ ), bestehend aus Ganzzahlen aus dem  $[-10^{15}, 10^{15}]$  Intervall. Schreiben Sie in C++ ein Programm mit der im schlechtesten Fall Zeitkomplexität  $O(n^3)$ , das auf dem Bildschirm vier natürliche Zahlen  $x, y, z, t$  ( $x, y, z, t \in [1, n]$ ,  $x < y < z < t$ ) ausgibt, damit  $a[x] + a[y] + a[z] + a[t] = 0$ . Falls in dem Vektor keine vier an unterschiedlichen Positionen Zahlen mit der Summe 0 vorhanden sind, wird der Wert -1 auf dem Bildschirm ausgegeben. **Anmerkung.** Jede Lösung, die nicht in die angegebene Komplexitätsklasse fällt, erhält eine Teil-Punktzahl. Algorithmen und Container aus der STL dürfen verwendet werden.

### Aufgabe 3. (2.25 Punkte)

Implementieren Sie in C++ die Klassen **MyObject**, **MyInteger**, **MyString**, **MyObjectList** und **MyListIterator**, damit die Ausgabe der unten implementierten Funktion dem Kommentar entspricht und der Speicher korrekt verwaltet ist.

```
void function()
{
    MyObjectList list{};
    list.add(new MyInteger{ 2 }).add(new MyString{ "Hi" });
    MyString* s = new MyString{ "Bye" };
    list.add(s).add(new MyString{ "5" });

    MyListIterator i{ list };
    while (i.isValid()) {
        MyObject* o = i.element();
        o->print();
        i.next();
    } // prints: 2 Hi Bye 5
}
```

### Aufgabe 4. (2.25 Punkte)

Implementieren Sie in C++ eine **Person**-Klasse mit den folgenden privaten Attributen: *name* (Zeichenkette), *vorname* (Zeichenkette), *alter* (Ganzzahl). Die Klasse besitzt einen Konstruktor, der alle Attribute initialisiert, und für alle Attribute Zugriffsmethoden. Implementieren Sie eine Funktion, die eine Liste von Personen sortiert. Die Funktion bekommt als Parameter eine Liste von Personen und eine Funktion, die das Sortierkriterium darstellt. Im schlechtesten Fall sollte die Zeitkomplexität der Sortierfunktion  $\Theta(n \log_2 n)$  sein, wobei  $n$  die Anzahl von Elementen der Liste ist.

Rufen Sie die Funktion auf, um eine Liste von Objekten vom Typ-**Person** aufsteigend zu sortieren:

- nach dem Attribut *name*
- nach dem Attribut *alter*
- nach den Attributen *alter* und *name* (falls *alter* gleich ist, vergleichen Sie *name*)

*Anmerkung:* Vordefinierte Sortiermethoden sind nicht erlaubt.

## TEIL Datenbanken

### Aufgabe 1. (4 Punkte)

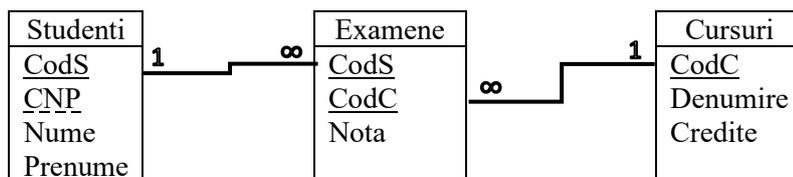
Bei einer Eiskunstlaufmeisterschaft können die Teilnehmer an verschiedenen Wettbewerben teilnehmen. In den Wettbewerben werden die Teilnehmer von Schiedsrichtern bewertet. Die Daten werden in einer relationalen Datenbank gespeichert.

- Ein Teilnehmer hat einen Code, einen Namen, einen Nachnamen und ein Geburtsdatum.
- Ein Wettbewerb hat einen Code, einen Namen, eine Beschreibung und ein Kalenderdatum.
- Ein Teilnehmer kann an mehr als einem Wettbewerb teilnehmen, und natürlich können auch mehrere Teilnehmer an einem solchen Wettbewerb teilnehmen.
- Ein Schiedsrichter hat einen Code, einen Namen und einen Nachnamen.
- Eine Bewertung wird von einem Schiedsrichter für einen Teilnehmer, der an einem bestimmten Wettbewerb teilnimmt, vorgenommen und beinhaltet die Vergabe einer vollen Punktzahl zwischen 1 und 10 für diesen Teilnehmer für diesen Wettbewerb. Ein Schiedsrichter kann mehr als einen Teilnehmer, der an demselben Wettbewerb teilnimmt, oder mehr als einen Wettbewerb für einen bestimmten Teilnehmer bewerten. Ein Schiedsrichter darf die Teilnahme eines bestimmten Teilnehmers an einem bestimmten Wettbewerb höchstens einmal bewerten. Ein Teilnehmer kann nur dann in einem Wettbewerb bewertet werden, wenn er an diesem Wettbewerb teilgenommen hat.

Erstellen Sie ein relationales Datenbankschema in BCNF, wobei Sie die Primärschlüssel, Kandidatenschlüssel und Fremdschlüssel sorgfältig hervorheben. Erstellen Sie das Schema auf eine der im folgenden Beispiel gezeigten Arten:

\* Beispiel für die Tabellen Studenti, Cursuri und Examene:

V1. Diagramm mit Tabellen, Primärschlüssel durch eine durchgezogene Linie unterstrichen, Kandidatenschlüssel durch eine gestrichelte Linie unterstrichen, Beziehungen direkt zwischen den Fremdschlüsseln und den entsprechenden Primär-/Kandidatenschlüsseln (z. B. Beziehung zwischen der CodS-Spalte in Examene und der CodS-Spalte in Studenti).



V2.

Studenti[CodS, CNP, Nume, Prenume]

Cursuri[CodC, Denumire, Credite]

Examene[CodS, CodC, Nota]

Primärschlüssel sind durch eine durchgezogene Linie und Kandidatenschlüssel durch eine gestrichelte Linie unterstrichen. {CodS} ist Fremdschlüssel in Examene, der auf {CodS} in Studenti verweist. {CodC} ist Fremdschlüssel in Examene, der auf {CodC} in Cursuri verweist.

### Aufgabe 2. (5 Punkte)

Gegeben seien folgende Relationenschemata:

Ziare[CodZiar, NumeZiar]

Jurnalisti[CodJurnalist, Nume, Prenume, *CodZiar*]

Categorii[CodCategorie, Denumire]

Articole[CodArticol, Titlu, DataPublicare, Text, *CodJurnalist*, *CodCategorie*]

Primärschlüssel sind unterstrichen. Fremdschlüssel sind kursiv geschrieben und haben denselben Namen wie die Spalten, auf die sie sich beziehen.

a. Schreiben Sie eine SQL-Abfrage, die die Gesamtzahl der Artikel in der Kategorie mit dem Namen *Economie* zurückgibt, die von Journalisten der Zeitung mit den Namen *Universul* veröffentlicht wurden.

b. Gegeben seien folgende Instanzen der Relationen *Categorii* und *Articole*:

### **Categorii**

<b>CodCategorie</b>	<b>Denumire</b>
1	c1
2	c2
3	c3

### **Articole**

<b>CodArticol</b>	<b>Titlu</b>	<b>DataPublicare</b>	<b>Text</b>	<b>CodJurnalist</b>	<b>CodCategorie</b>
1	a1	1-1-2022	t1	1	2
2	a2	1-2-2022	t2	2	3
3	a3	1-1-2022	t3	3	1
4	a4	1-5-2022	t4	1	2
5	a5	1-3-2022	t5	2	1
7	a7	1-7-2022	t7	2	2
8	a8	1-5-2022	t8	4	2

b1. Geben Sie das Ergebnis der Auswertung der folgenden Abfrage für die angegebenen Instanzen an. Geben Sie ausschließlich die Werte der Tupel und Spaltennamen im Ergebnis an, ohne alle Schritte der Abfrageauswertung zu nennen.

```
SELECT A.CodJurnalist, COUNT(*) Nr
FROM Article A INNER JOIN Categorii C ON A.CodCategorie = C.CodCategorie
WHERE C.Denumire = 'c2'
GROUP BY A.CodJurnalist
HAVING 2 < (SELECT COUNT(*)
            FROM Article A2
            WHERE A2.CodJurnalist = A.CodJurnalist)
```

b2. Begründen Sie für jede der folgenden funktionalen Abhängigkeiten, ob diese für die Daten in der Instanz *Articole* gilt oder nicht:

- $\{CodJurnalist\} \rightarrow \{Titlu\}$
- $\{CodArticol\} \rightarrow \{Text\}$

## THEMA Betriebssysteme

### Aufgabe 1 (5p)

Beantworten Sie die folgenden Fragen zur Ausführung des unten stehenden Codefragments, unter der Annahme, dass die Pipes und der Prozess erfolgreich erstellt wurden.

1	int main(int argc, char** argv) {	19	close(b[1]); close(c[0]);
2	int a[2], b[2], c[2], i;	20	for(i=0; i<3; i++) {
3	char val=0, x=0;	21	read(b[0], &x, 1);
4		22	read(a[0], &val, 1);
5	pipe(a); pipe(b); pipe(c);	23	val++;
6	if(fork() == 0) {	24	write(a[1], &val, 1);
7	close(b[0]); close(c[1]);	25	write(c[1], &x, 1);
8	for(i=0; i<3; i++) {	26	}
9	write(a[1], &val, 1);	27	close(a[0]);close(a[1]);
10	write(b[1], &x, 1);	28	close(b[0]);close(c[1]);
11	read(c[0], &x, 1);	29	printf("%d\n", x);
12	read(a[0], &val, 1);	30	
13	}	31	wait(NULL);
14	printf("%d\n", val);	32	return 0;
15	close(a[0]);close(a[1]);	33	}
16	close(b[1]);close(c[0]);		
17	exit(0);		
18	}		

- Welchen Zweck erfüllt der c-pipe?
- Was wird auf der Konsole angezeigt?
- Was passiert, wenn man die Zeilen 11 und 12 überschneidet?
- Wie wirkt es sich auf den Prozess aus, wenn Zeile 7 zwischen die Zeilen 14-15 und Zeile 19 zwischen die Zeilen 26-27 verschoben wird?
- Was passiert, wenn die Zeile 17 entfernt wird?

### Aufgabe 2 (4p)

Beantworten Sie die folgenden Fragen zur Ausführung des unten stehenden UNIX-Shell-Skripts.

1	#!/bin/bash
2	
3	for F in `find /usr/sbin -type f`; do
4	if [ ! -x "\$F" ] && ls -l "\$F"   grep -q -E "^-({2}x . {5}x . {8}x) "
5	then
6	echo "\$F:" `ps -e -f   grep -E "\$F"   grep -E -v "grep"   wc -l`
7	fi
8	done

- Welche Werte kann die Variable F annehmen?
- Erklären Sie den regulären Ausdruck in Zeile 4.
- Welche Dateien erfüllen den ersten Teil der Bedingung in Zeile 4 (vor &&)?
- Welche Dateien erfüllen den zweiten Teil der Bedingung in Zeile 4 (nach &&)?
- Welchen Zweck erfüllt der zweite grep in Zeile 6?

# BAREM INFORMATICĂ

## VARIANTA 1

### Subiect Algoritmă și Programare

Oficiu – 1p

#### Problema 1. Punctaj - 1.5p

- Răspuns corect (șirul  $z$  va conține elementele șirului  $x$  în ordine inversă): **0.75 p**
- Justificare: **0.75p**

#### Problema 2. Punctaj - 3p

- Soluție având complexitatea timp în caz defavorabil  $O(n^3)$  - 3p
- Soluție având complexitatea timp în caz mediu  $\theta(n^3)$  - 1.75p
- Soluție având complexitatea timp  $O(n^3 \log_2 n)$  - 1.25p
- Soluție având complexitatea timp  $O(n^4)$  - 0.5p

#### Problema 3. Punctaj - 2.25p

- Clasa **MyObject (0.2p)**: abstractă, funcția abstractă **print**, virtual destructor
- Clasa **MyInteger (0.3p)**: mostenire, camp privat de tip *int* + constructor + funcția **print**
- Clasa **MyString (0.3p)**: mostenire, camp privat de tip *string* + constructor, funcția **print**
- Clasa **MyObjectList (0.85p)**:
  - vector de elemente *Object* (pointers) (0.2), funcția **add** (0.2), funcția **length** (0.1), destructor (0.15), funcție care accesează elementul de pe o poziția dată sau funcție **getObjects** sau friend class *MyListIterator* (0.2).
- Clasa **MyListIterator (0.6p)**:
  - câmpuri private de tip *MyObjectList* și *int* (elementul curent) + constructor (0.25), funcția **isValid** (0.1), funcția **next** (0.1), funcția **element** (0.15).

#### Problema 4. Punctaj - 2.25p

- Clasa **Persoana - 0.3p**
- Signatura funcției: lista/array + parametru de tip funcție (de comparare sau funcție care returnează cheia după care sortăm) - **0.3p**
- Implementare **Merge Sort** (partea recursivă/interclasare) -  $0.6+0.75 = 1.35p$
- Implementare funcții de comparare/cheie (sau funcții lambda) și apel corect pentru cele 3 variante de sortare - **0.3p**

### Subiect Baze de date

Oficiu – 1p

#### Problema 1. Punctaj - 4p

- relații cu atribute corecte, chei primare, chei candidat: **3p**
- legături modelate corect (chei externe): **1p**

#### Problema 2. Punctaj - 5p

- **a** - rezolvarea completă a interogării: **2.5p**

- **b1** - rezultat evaluare interogare:

CodJurnalist	Nr
2	1

- coloane – **0.5p**

- valori tuplu – **1p**

- **b2** - {CodJurnalist} → {Titlu} nu este satisfăcută – **0.25p**; **0.25p** explicație  
- {CodArticol} → {Text} este satisfăcută – **0.25p**; **0.25p** explicație

**Notă:** La specializările Informatică engleză și Informatică maghiară se iau în considerare versiunile traduse în limbile corespunzătoare.

## **Subiect Sisteme de operare**

**Oficiu – 1p**

**Problema 1. Punctaj – 5p**

- a) Previne procesul fiu de la a citi din a ceea ce a scris el însuși – 1p
- b) Se va afișa 0 și 3 în orice ordine – 1p
- c) Dacă procesul fiu citește din a ce a scris chiar el, se blochează, altfel funcționează ca la a) – 1p
- d) Funcționarea nu este afectată – 1p
- e) Procesul fiu își continuă execuția cu instrucțiunile de după IF – 1p

**Problema 2. Punctaj – 4p**

- a) Numele fișierelor normale din directorul /usr/bin găsite recursiv și în subdirectoare – 0.5p
- b) La început de linie, minus urmat de orice grup de caractere de lungime 2, 5 sau 8 urmat de x – 1p
- c) Fișierele care nu pot fi executate de utilizatorul care rulează scriptul – 0.5p
- d) Fișierele care au drept de execuție pentru owner, group sau others – 1p
- e) Elimină din lista de procese cele care conțin în linia de comandă \$F pe cele care conțin cuvântul `grep` – 1p