

СОФИЙСКИ УНИВЕРСИТЕТ  
“СВ. КЛИМЕНТ ОХРИДСКИ”



ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА

## ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”

### ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листове.
- Пишете само на предоставените листове, без да ги разкопчавате.
- Решението на една задача трябва да бъде на същия лист, на който е и нейното условие (т.е. може да пишете отпред и отзад на листа със задачата, но не и на лист на друга задача).
- Ако имате нужда от допълнителен лист, можете да поискате от квесторите.
- На един лист не може да има едновременно и чернова, и белова.
- Черновите трябва да се маркират, като най-отгоре на листа напишете “ЧЕРНОВА”.
- Ако решението на една задача не се побира на нейния лист, трябва да поискате нов бял лист от квесторите. Той трябва да се защити с телбод към листа със задачата.
- Всеки от допълнителните листове (белова или чернова) трябва да се надпише най-отгоре с вашия факултетен номер.
- Черновите също се предават и се защитват в края на работата.
- Времето за работа по изпита е 3 часа.
- Всяка напълно и коректно решена задача, съгласно критериите за оценяване, носи 10 точки.
- Оценката от практическата част се изчислява по формулата  $\frac{x}{10}$ , където  $x$  е сумата от получените точки от задачите. Максималната оценка е Отличен 6.00.
- За успешно полагане на изпита са необходими поне 30 точки, като при събрани под 30 точки оценката е Слаб 2.00.

*Изпитната комисия ви пожелава успешна работа!*

---

**Задача 1.** Решете задачата на езика C++. Решения на друг език носят нула точки. Отговорите на подточки 1А, 1Б, 1В и 1Г трябва да съвпадат с това, което би извела програмата. При несъответствие отговорът се оценява с нула точки. Решението на подточка 1Д трябва да бъде технически издържано (например не бива да изтича памет, трябва да се спазват добрите практики за структуриране на програмата и т.н.). Ако решението съдържа сериозни грешки, то се оценява с нула точки.

**1А) (1 точка)** Какъв ще бъде изходът от изпълнението на следния фрагмент:

```
int calc(int a[5])
{
    int sum = 8;
    for (int i = 0; a[i]; ++i)
        sum += i;
    return sum;
}
```

```
int a[7] = {1, 2, 3};
std::cout << calc(a);
```

Отговор: \_\_\_\_\_

**1Б) (1 точка)** Какъв ще бъде изходът от изпълнението на следния програмен фрагмент?

```
char text[] = "hello", *p = text;
while (*p) std::cout << ++*p++;
```

Изберете един от следните отговори:

- а) Грешка по време на компилация.
- б) Грешка по време на изпълнение.
- в) ifmmp
- г) el
- д) Безкраен цикъл.

**1В) (1 точка)** Какво ще изведе следният фрагмент (приемаме, че е част от валидна програма):

```
int *pt;
int a[3] = {4, 19, 13};
pt = &a[1];
pt += 1;
std::cout << *pt << std::endl;
```

Отговор: \_\_\_\_\_

**1Г) (1 точка)** В дадените по-долу празни места попълнете какви ще бъдат стойностите на елементите на двата масива А и В след обръщението към функцията f.

```
void f(int * arr1, const int * arr2)
{
    int *p1 = arr1;
    const int *p2 = arr2;

    while(*p2 >= 0)
    {
        *p1++ = *p2++;
    }
}
```

```
void main()
{
    int A[4] = {-1, -2, -3, -4};
    int B[4] = {10, 20, 30, -1};

    f(A, B);
}
```

Отговор:

A[0] = \_\_\_\_ A[1] = \_\_\_\_ A[2] = \_\_\_\_ A[3] = \_\_\_\_

B[0] = \_\_\_\_ B[1] = \_\_\_\_ B[2] = \_\_\_\_ B[3] = \_\_\_\_

**1Д) (6 точки)** Дадени са структура `Point`, описваща точка в декартова координатна система с координати `x` и `y` от тип `float`, и структура `Circle`, описваща окръжност с център `center` от тип `Point` и радиус `r` от тип `float`.

Да се дефинира функция `findRelativePosition`, която определя относителната позиция на две дадени окръжности една спрямо друга. Резултатът от изпълнението на функцията е стойност от изброения тип:

```
RelativePosition {NO_COMMON_POINTS, TOUCHING,  
                 INTERSECTING, SAME}.
```

със следния смисъл:

- `NO_COMMON_POINTS`: без общи точки
  - `TOUCHING`: допиращи се
  - `INTERSECTING`: пресичащи се
  - `SAME`: съвпадат
-

**Задача 2.** Задачата да се реши на езика C++. Отговорите си попълнете в посочените за тях полета. Оценява се само попълненото в полетата. Всичко друго писано по листа не носи точки.

**2А) (6 точки)** Разгледайте програмата. Някои от конструкциите във функцията main са номерирани в коментар. За всеки такъв ред посочете какво ще се изведе в резултат от неговото изпълнение. Отговор, който не съответства точно на това, което извежда програмата, носи нула точки. Ако смятате, че някой от редовете няма да изведе нищо, напишете "нищо". Ако смятате, че някой от тях ще предизвика грешка, опишете каква. Ако за някоя конструкция не бъде попълнено нищо, отговорът не носи точки.

```
#include <iostream>
using std::cout;

class test {
public:
    int var;

    test() { cout << "test()\n"; var = 0; }

    test &operator=(const test &other) {
        if (this != &other) {
            cout << "copy\n";
            var = other.var;
        } else {
            cout << "self-assignment\n";
        }
        return *this;
    }

    static test &instance() {
        static test obj;
        return obj;
    }

    test &self() { return *this; }
    void _oncreate() { cout << "_oncreate()\n"; }
    void _oncopy() { cout << "_oncopy()\n"; }
};

int main() {
    test &r1 = test::instance(); // 1
    test &r2 = test::instance(); // 2
    r1.var = 10; //3
    cout << r2.var << "\n"; // 4
    r1.self() = r2; // 5
    new test[3]; // 6
}
```

**Посочете отговорите си тук:**  
(1 точка) Ред //1 ще изведе:

(1 точка) Ред //2 ще изведе

(1 точка) Ред //3 ще изведе:

(1 точка) Ред //4 ще изведе

(1 точка) Ред //5 ще изведе

(1 точка) Ред //6 ще изведе

**2Б) (1 точка)** Възможно ли е в един клас да се дефинират няколко различни копиращи конструктора?

- а) Да
- б) Не

**2В) (3 точки)** Нека са дадени следните дефиниции:

```
class foo {
public:
    foo(int) {}
};

void g(foo) {}
void f(int) {}
```

Срещу всеки от изразите посочете (като напишете "да"или "не") дали ще се компилира:

```
g(5); _____
f(foo(5)); _____
foo('a'); _____
```

**Задача 3.** (10 точки) Софтуерна система е проектирана да приема резултати от независими тестове на машини. Всеки тест включва проверка на 5 операции. Всяка машина трябва да има изпълнени 5 индивидуални теста, като максималният резултат от всеки тест е 20. Системата изчислява и общия резултат от всички тестове. Резултатите от тестовете се сравняват по двойки. Машината се подлага на следващи тестове при следните условия:

- Ако два теста имат разлика по-голяма от 2 при проверка на една и съща операция;
- Ако разликата в крайните резултати от 2 теста е по-голяма от 10.

Опишете как ще се извърши тестването с класове на еквивалентност и анализ на граничните стойности, като включите следните стъпки:

1. Определете валидни и невалидни класове на еквивалентност.
2. Определете гранични стойности за тестване.
3. Дефиниране на тестови сценарии с конкретни стойности. Какъв е минималният брой на тестовите сценарии, които покриват класовете на еквивалентност и граничните стойности?

**Задача 4.** Дадена е базата от данни **Movies**.

Таблицата **Studio** съдържа информация за филмови студиа:

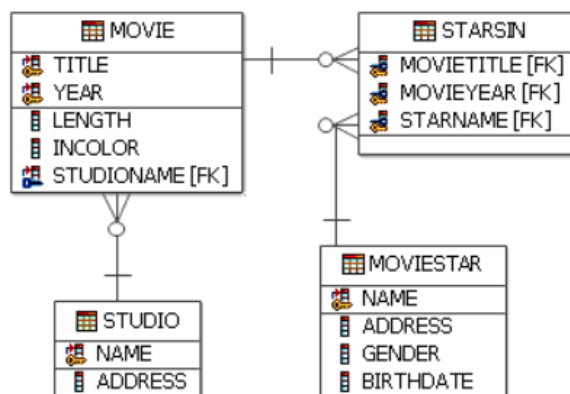
- *name* — име, първичен ключ;
- *address* — адрес.

Таблицата **Movie** съдържа информация за филми. Колоните *title* и *year* заедно формират първичния ключ.

- *title* — заглавие;
- *year* — година, в която филмът е заснет;
- *length* — дължина в минути;
- *incolor* — 'Y' за цветен филм и 'N' за черно-бял;
- *studioName* — име на студио, външен ключ.

Таблицата **MovieStar** съдържа информация за филмови звезди:

- *name* — име, първичен ключ;
- *address* — адрес;
- *gender* — пол, 'F' за жена и 'M' за мъж;
- *birthdate* — рождена дата.



Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите колони заедно формират първичния ключ. Колоните *movietitle* и *movieyear* образуват външен ключ към **Movie**.

- *movietitle* — заглавие на филма;
- *movieyear* — година на заснемане на филма;
- *starname* — име на филмовата звезда, външен ключ.

**Задача 4.1** (4 точки): Посочете заявката, която извежда имената на всички студиа, които имат поне един цветен филм и едновременно с това поне един филм с неизвестна дължина:

а) `SELECT studioName  
FROM Movie  
WHERE inColor != 'Y'  
AND studioName =  
(SELECT studioName  
FROM Movie  
WHERE length IS NULL);`

в) `SELECT studioName  
FROM Movie  
WHERE inColor = 'Y'  
INTERSECT  
SELECT studioName  
FROM Movie  
WHERE length IS NULL;`

б) `SELECT studioName  
FROM Movie  
WHERE inColor = 'Y'  
AND length = NULL;`

г) `SELECT name  
FROM Studio  
LEFT JOIN Movie  
ON name = studioName  
WHERE inColor = 'Y'  
OR length = NULL;`

**Задача 4.2** (6 точки): Посочете заявката, която за всяка актриса извежда името и броя на черно-белите филми, в които е участвала. Ако за дадена актриса няма информация в какви филми е участвала или е играла само в цветни филми, срещу нейното име да се изведе числото 0.

a)SELECT name, COUNT(DISTINCT title)  
FROM Movie  
RIGHT JOIN StarsIn  
ON title = movieTitle  
AND year = movieYear  
RIGHT JOIN MovieStar  
ON starName = name  
WHERE gender = 'F'  
GROUP BY name  
HAVING inColor = 'N';

б)SELECT name, COUNT(title)  
FROM MovieStar  
LEFT JOIN StarsIn ON name = starName  
LEFT JOIN Movie  
ON movieTitle = title  
AND movieYear = year  
AND inColor = 'N'  
WHERE gender = 'F'  
GROUP BY name;

в)SELECT name, COUNT(title)  
FROM MovieStar  
LEFT JOIN StarsIn ON name = starName  
LEFT JOIN Movie  
ON movieTitle = title  
AND movieYear = year  
HAVING inColor = 'N'  
AND gender = 'F'  
GROUP BY name;

г)SELECT starName, COUNT(\*)  
FROM StarsIn  
LEFT JOIN Movie  
ON title = movieTitle  
AND year = movieYear  
WHERE inColor = 'N' AND gender = 'F'  
GROUP BY name;

**Задача 5.** (10 точки) Направете декомпозиция на модулите от архитектурата на софтуерна система за *подпомагане на доставката и търговията с лекарствени продукти*, според дадените по-долу изисквания. Обосновете защо така проектираната архитектура удовлетворява изискванията.

**R1.** Потребители на системата са различни юридически лица — търговци (т.е. аптеки), доставчици, вносители и производители на лекарства

**R2.** Всички функционалности на системата са достъпни само за регистрирани потребители от изброените по-горе групи.

**R3.** Също така има и потребители с административен достъп, които може да одобряват и проверяват легитимността на регистрациите на описаните в R1 групи потребители.

**R4.** В профила на всеки потребител трябва да има възможност за интеграция с електронен подпис за целите на удостоверяване на самоличността при договаряне на поръчки и подписване на договори.

**R5.** Системата трябва да поддържа връзка с единен регистър на лечебните заведения, както и регистри на легитимните вносители, търговци, доставчици и производители на лекарствени средства. Достъпът до тези регистри става по специфичен за тях протокол.

**R6.** Системата да дава възможност за проследяване и изпълнение на електронни рецепти. За целта също трябва да се осигурява връзка с електронно здравно досие на пациентите.

**R7.** Системата трябва да поддържа електронен дневник на всички осъществени договори, поръчки и продажби – както от търговци към крайни потребители, така и между търговците и вносителите/доставчиците/производителите на лекарствени средства.

**R8.** Всички данни в описания в R7 регистър трябва да са защитени и достъпни само заинтересованите лица, които извършват съответните действия.

**R9.** Системата трябва да е 100% налична, 24 часа в денонощието, като не се допуска липса на услуга дори и при срыв в някой от компонентите ѝ.



**Задача 6.** (10 точки) Нека  $L$  е регулярен език над азбуката  $\Sigma$ . Вярно ли е, че езикът  $L' = \{uv \mid u, v \in \Sigma^* \text{ и } vu \in L\}$  е регулярен? Обосновете отговора си!

**Задача 7.** (10 точки) Намерете неопределения интеграл

$$\int \operatorname{arctg} \sqrt{x} dx, \quad x > 0.$$

**Чернова**