

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 защитават в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Задача 1. Задачата да се реши на езика C++.

1) Нека е дефиниран масивът

```
int arr[] = { 1, 2, 3 };
```

Срещу всеки от изразите да се посочи каква ще бъде неговата оценка.

```
arr[1] == *(arr+2) _____  
arr == &arr[0] _____  
(arr+1) == &arr[1] _____  
*arr == arr[0] _____
```

2) Нека е дадена следната дефиниция:

```
void mystery(const char* str)  
{  
    while (*str && *(str+1)) {  
        std::cout << *str;  
        str += 2;  
    }  
}
```

Да се посочи какво ще изведе на екрана обръщението:

```
mystery("abcdef");
```

3) Нека са дадени следните дефиниции:

```
char s1[] = "Hello";  
char s2[] = "world!";  
char result[80];
```

Да се довърши програмният фрагмент, така че след изпълнението му в `result` да се съхрани коректното представяне на низа "Hello world!". На празните места трябва да се попълнят имената на подходящи стандартни функции за работа с низове.

```
_____(result, s1);  
_____(result, " ");  
_____(result, s2);
```

4) Да се довърши кодът на рекурсивните функции, така че `f` да проверява дали символният низ, сочен от `word`, се съдържа като подниз в `text`. За определеност считаме, че празният низ се съдържа във всеки друг.

```
bool g(const char* text, const char* word)  
{  
    if (!*word) return true;  
    if (!*text) return _____;  
    if (*word != *text) return false;  
    return g(_____, _____);  
}
```

```
bool f(const char* text, const char* word)  
{  
    if (!*word) return _____;  
    if (!*text) return false;  
    return g(_____, _____) ||  
           f(_____, _____);  
}
```

5) Да се посочи какво ще изведе на екрана даденият по-долу фрагмент:

```
char arr[3][3] = { 'a', 'b', 'c',  
                  'd', 'e', 'f',  
                  'g', 'h', 'i' };  
for (int i = 0; i < 3; ++i)  
    std::cout << arr[2-i][i];
```

6) Да се посочи какво ще изведе на екрана даденият по-долу фрагмент:

```
double var = 5 / 2;  
std::cout << var;
```

Критерии за оценяване

- Точки се дават само за напълно коректно посочени отговори.
- В подточките, в които се изисква да се посочи какво ще се изведе, точки се дават само ако отговорът напълно съвпада с това, което извежда съответният код. В противен случай се дават 0 т.
- В задачите за посочване на стойност на израз, ако отговорът не съвпада напълно с коректното решение се дават нула точки.
- В задачите за посочване на стойност на израз, ако вместо булевите литерали true/false се посочат числата 1/0, точките се намаляват наполовина.
- В задачата за довършване на кода на функцията, ако написаното не е синтактично или логически коректно или е различно от коректния отговор, се дават 0 т.
- Сумата от точките се закръгля до цяло число.

Максималната оценка за всяка подточка е както следва:

- Подточка 1: 2 точки (по 0.5 точки за всеки напълно коректен отговор)
- Подточка 2: 1 точка
- Подточка 3: 1 точка (0.5 точки, ако коректно е посочен `strcpy` за първата функция; 0.5 точки, ако коректно е посочена `strcat` за вторите две извиквания).
- Подточка 4: 4 точки (по 0.5 точки за всеки напълно коректен отговор).
- Подточка 5: 1 точка
- Подточка 6: 1 точка (отговорът се счита за коректен, независимо дали е посочен с десетична точка; например 2, 2.0 или 2,0 са коректни отговори).

Примерно решение:

1)

```
arr[1] == *(arr+2) --> false
arr == &arr[0] --> true
(arr+1) == &arr[1] --> true
*arr == arr[0] --> true
```

2)

ace

3)

```
strcpy(result, s1);
strcat(result, " ");
strcat(result, s2);
```

4)

```
bool g(const char* text, const char* word)
{
    if (!*word) return true;
    if (!*text) return false;
    if (*word != *text) return false;
    return g(text + 1, word + 1);
}
```

```
bool f(const char* text, const char* word)
{
    if (!*word) return true;
    if (!*text) return false;
    return g(text, word) ||
           f(text + 1, word);
}
```

5)

ges

6)

ВЪЗМОЖНИ ОТГОВОРИ: 2, 2.0, 2,0 и т.н.

Задача 2. Задачата да се реши на езика C++.

1) Освен конструктора по подразбиране (default constructor), кои други функции влизат в “голямата четворка” (функциите от т.нар. “rule-of-3”)?
Да се попълнят имената им в полетата долу:

2) Нека е дадена дефиницията:

```
class foo {  
public:  
    virtual void f() {};  
    void g() {};  
};
```

Срещу всеки от редовете, които извикват f или g, да се запише “статично” или “динамично” според вида свързване, който ще се използва за тях.

```
foo obj;  
foo& ref = obj;  
obj.f(); _____  
obj.g(); _____  
ref.f(); _____  
ref.g(); _____
```

3) Нека са дадени следните дефиниции:

```
class base {  
public: int a;  
private: int b;  
};  
class derived : protected base { };
```

Да се посочи каква ще бъде видимостта на променливите a и b в класа derived – public, protected или private.

- Видимост на a: _____
- Видимост на b: _____

4) Нека класът X е абстрактен. Срещу всяко от твърденията да се посочи “да” или “не” според това дали е вярно:

- Могат да се създават обекти от тип X: _____
- Могат да се създават референции (reference) към обекти от тип X: _____

5) Нека е дадена следната дефиниция:

```
struct s {  
public:  
    static int var;  
    s() { var = 5; }  
};  
int s::var = 0;
```

Да се посочи какво ще изведе следният фрагмент:

```
std::cout << '(' << s::var << ')';  
s obj1;  
obj1.var = 10;  
s obj2;  
std::cout << '-' << s::var << '-';
```

6) Да се допълни дефиницията на класа test, така че функцията f да бъде чиста виртуална (pure-virtual) и класът да може коректно да се използва като основа на полиморфна йерархия.

```
class test {  
public:  
    _____ void f() _____;  
};
```

7) Да се допълни дефиницията на шаблона Array, така че функцията test да се компилира коректно и да извежда на стандартния изход 55.

```
_____ <_____>  
class Array {  
    static const size_t size = 10;  
    T data[size];  
public:  
    _____ at(size_t index) {  
        if (index _____)  
            throw std::out_of_range("error");  
        return data[index];  
    }  
};
```

```
void test() {  
    Array<int> a;  
    a.at(0) = 5;  
    std::cout << a.at(0);  
    Array<Array<int>> b;  
    b.at(0) = a;  
    std::cout << b.at(0).at(0);  
}
```

Критерии за оценяване

- Точки се дават само за напълно коректно посочени отговори.
- В подточките, в които се изисква да се посочи какво ще се изведе, точки се дават само ако отговорът напълно съвпада с това, което извежда съответният код. В противен случай се дават 0 т.
- В задачите за посочване на стойност на израз, ако отговорът не съвпада напълно с коректното решение, се дават нула точки.
- В задачите за посочване на стойност на израз, ако вместо булевите литерали true и false се посочат числата 1 и 0, точките се намаляват наполовина.
- В задачата за довършване на кода на функцията, ако написаното не е синтактично или логически коректно или е различно от коректния отговор, се дават 0 т.
- Сумата от точките се закръгля до цяло число.

Максималната оценка за всяка подточка е както следва:

1. 1 точка (1/3 точки за коректен отговор, -1/3 точки за грешно посочена функция, минимална оценка: 0 точки).
2. 2 точки (по 0.5 за коректен отговор).
3. 1 точка (по 0.5 за коректен отговор).
4. 1 точка (по 0.5 за коректен отговор).
5. 1 точка (по 0.5 съответно ако е посочено извеждането на 0 и на 5; Ако в отговора не са включени скобите и тиретата, оценката се намалява с 0.2 точки).
6. 2 точки (по 0.5 съответно за virtual и за = 0 във функцията f; 1 точка за виртуален деструктор).
7. 2 точки (по 0.5 за всяко коректно попълнено място).

Примерно решение:

1)

- “копиращ конструктор” или “конструктор за копиране” или “copy constructor”
- “копиращо присвояване” или “операция/оператор за присвояване” или “copy assignment” или “assignment operator” или “operator=”
- “деструктор” или “destructor”

2)

```
foo obj;  
foo& ref = obj;  
obj.f();    статично  
obj.g();    статично  
ref.f();    динамично  
ref.g();    статично
```

3)

- Видимост на a: protected
- Видимост на b: private или “няма видимост в производния клас”

4)

- Могат да се създават обекти от тип X: не
- Могат да се създават референции (reference) към обекти от тип X: да

5)

(0)-5-

6)

```
class test {  
public:  
    virtual void f() = 0;  
    virtual ~test() {}  
};
```

7)

```
template <typename T>  
class Array {  
    static const size_t size = 10;  
    T data[size];  
public:  
    T& at(size_t index) {  
        if (index >= size)  
            throw std::out_of_range("error");  
        return data[index];  
    }  
};
```

Задача 3. Разглежда се множество от системи линейни уравнения с реални коефициенти, където λ е реален параметър и свободните коефициенти b_1, b_2, b_3, b_4 са реални числа:

$$(*) \quad \left| \begin{array}{lcl} 4x_1 - 7x_2 + 6x_3 + 9x_4 & = & b_1 \\ -2x_1 + 11x_2 - 8x_3 + 3x_4 & = & b_2 \\ 3x_1 - 9x_2 + 7x_3 + 3x_4 & = & b_3 \\ x_1 - x_2 + x_3 + \lambda x_4 & = & b_4 \end{array} \right.$$

- а) В случая на хомогенна система (т.е. когато $b_1 = b_2 = b_3 = b_4 = 0$) да се реши системата и да се определи фундаментална система от решения (ФСР) в зависимост от параметъра λ .
- б) Нека B_3 е множеството, състоящо се от всички свободни стълбове, за които системата $(*)$ има решение, когато $\lambda = 3$, т.е.:

$$B_3 = \{(b_1, b_2, b_3, b_4) \mid \text{системата } (*) \text{ има решение при } \lambda = 3\}.$$

Да се докаже, че B_3 е линейно подпространство на 4-мерното пространство \mathbb{R}^4 , да се определи размерността му и да се намери базис на B_3 .

Примерно решение:

- а) (5 т.) Преобразуваме матрицата на системата (без стълба от свободните членове, защото е нулев).
Първо изваждаме третия ред от първия, както и прибавяме третия ред към втория ред.

$$A = \begin{pmatrix} 4 & -7 & 6 & 9 \\ -2 & 11 & -8 & 3 \\ 3 & -9 & 7 & 3 \\ 1 & -1 & 1 & \lambda \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 & 6 \\ 1 & 2 & -1 & 6 \\ 3 & -9 & 7 & 3 \\ 1 & -1 & 1 & \lambda \end{pmatrix}$$

Първия ред изваждаме от втория, първия ред вадим и от четвъртия и освен това първия ред умножен по 3 го вадим от третия. Третия ред го разделяме на 5, вадим от последния ред и подреждаме редовете. Получава се следната матрица.

$$A \sim \begin{pmatrix} 1 & 2 & -1 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & -15 & 10 & -15 \\ 0 & -3 & 2 & \lambda - 6 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 & 6 \\ 0 & -3 & 2 & -3 \\ 0 & 0 & 0 & \lambda - 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

В зависимост от стойността на параметъра λ имаме два случая:

Случай 1: Когато $\lambda \neq 3$. В този случай $\lambda - 3 \neq 0$ и матрицата има три линейно независими редове и нейния ранг е $r(A) = 3$. В този случай $x_4 = 0$, решението зависи от 1 реален параметър $x_3 = p$ и замествайки в първите два реда получаваме $x_2 = \frac{2p}{3}$ и $x_1 = \frac{-p}{3}$. Решението е едномерно линейно подпространство на четиримерното пространство \mathbb{R}^4 и общият вид на решението е

$$U_{\lambda \neq 3} = \left\{ \left(\frac{-p}{3}, \frac{2p}{3}, p, 0 \right) \mid p \in \mathbb{R} \right\}.$$

Фундаменталната система от решения е базис на подпространството от решения и се състои от един вектор и при $p = 3$ получаваме вектора $g = (-1, 2, 3, 0)$, който формира ФСР на решението в случая $\lambda \neq 3$.

Случай 2: Когато $\lambda = 3$. В този случай $\lambda - 3 = 0$ и матрицата има ранг 2. Решението зависи от два параметъра $x_3 = p$ и $x_4 = q$, заместваем и получаваме $x_2 = \frac{2p}{3} - q$ и $x_1 = -\frac{p}{3} - 4q$. Общият вид на решението, което е двумерно подпространство, е

$$U_{\lambda=3} = \left\{ \left(-\frac{1}{3}p - 4q, \frac{2}{3}p - q, p, q \right) \mid p, q \in \mathbb{R} \right\}.$$

Базис на решението, което е фундаментална система от решения, можем да получим когато дадем два набора независими стойности на параметрите, например при $p_1 = 0$ и $q_1 = 1$ получаваме $g_1 = (-4, -1, 0, 1)$ и при $p_2 = 3$ и $q_2 = 0$ получаваме $g_2 = (-1, 2, 3, 0)$. Векторите g_1, g_2 образуват фундаментална система от решения в случая $\lambda = 3$.

- б) (5 т.) *Първи начин:* Нека стълбовете на матрицата на системата са c_1, c_2, c_3, c_4 и свободния стълб е b , където

$$c_1 = \begin{pmatrix} 4 \\ -2 \\ 3 \\ 1 \end{pmatrix}, c_2 = \begin{pmatrix} -7 \\ 11 \\ -9 \\ -1 \end{pmatrix}, c_3 = \begin{pmatrix} 6 \\ -8 \\ 7 \\ 1 \end{pmatrix}, c_4 = \begin{pmatrix} 9 \\ 3 \\ 3 \\ 3 \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}.$$

От теоремата на Руше е известно, че системата (*) има решение тогава и само тогава, когато рангът на матрицата на системата е равен на ранга на разширената матрица. За произволна

матрица знаем, че нейният ранг е равен на ранга на стълбовете на матрицата, който пък е равен на размерността на линейната обвивка на вектор-стълбовете:

$$b \in B_3 \Leftrightarrow r(A) = r(\bar{A}) \Leftrightarrow r(c_1, \dots, c_4) = r(c_1, \dots, c_4, b) \Leftrightarrow \dim \ell(c_1, \dots, c_4) = \dim \ell(c_1, \dots, c_4, b).$$

Това е възможно точно когато стълбът от свободните членове принадлежи на линейната обвивка на стълбовете на системата:

$$b \in B_3 \Leftrightarrow b \in \ell(c_1, \dots, c_4) \Rightarrow B_3 = \ell(c_1, \dots, c_4).$$

Оттук получаваме, че множеството B_3 е линейно подпространство и от пресмятанията в т. а) получаваме

$$\dim B_3 = r(c_1, \dots, c_4) = r(A) = 2.$$

Базис на това двумерно подпространство ще намерим като вземем произволни два линейно независими вектора измежду стълбовете c_1, \dots, c_4 . Например един базис е

$$B_3 = \ell(c_1, c_2), \text{ където } c_1 = \begin{pmatrix} 4 \\ -2 \\ 3 \\ 1 \end{pmatrix}, c_2 = \begin{pmatrix} -7 \\ 11 \\ -9 \\ -1 \end{pmatrix}.$$

Втори начин: Преобразуваме матрицата заедно със стълба от свободните членове. Използвайки четвъртия ред анулираме всички елементи на първо място от останалите редове. С новополучения първия ред анулираме лявата част на втори и трети ред:

$$\left(\begin{array}{cccc|c} 4 & -7 & 6 & 9 & b_1 \\ -2 & 11 & -8 & 3 & b_2 \\ 3 & -9 & 7 & 3 & b_3 \\ 1 & -1 & 1 & 3 & b_4 \end{array} \right) \sim \left(\begin{array}{cccc|c} 0 & -3 & 2 & -3 & b_1 - 4b_4 \\ 0 & 9 & -6 & 9 & b_2 + 2b_4 \\ 0 & -6 & 4 & -6 & b_3 - 3b_4 \\ 1 & -1 & 1 & 3 & b_4 \end{array} \right) \sim \left(\begin{array}{cccc|c} 0 & -3 & 2 & -3 & b_1 - 4b_4 \\ 0 & 0 & 0 & 0 & 3b_1 + b_2 - 10b_4 \\ 0 & 0 & 0 & 0 & -2b_1 + b_3 + 5b_4 \\ 1 & -1 & 1 & 3 & b_4 \end{array} \right)$$

Системата има решение точно когато дясната част на втори и трети ред е равна на нула

$$b \in B_3 \Leftrightarrow \begin{cases} 3b_1 + b_2 - 10b_4 = 0 \\ -2b_1 + b_3 + 5b_4 = 0 \end{cases}$$

Това е хомогенна система и множеството B_3 е нейно решение, което е подпространство на \mathbb{R}^4 . Даваме два независими набора от стойности за b_1, b_4 и определяме стойностите на b_2, b_3 и по този начин намираме базис на подпространството B_3 :

$$\begin{aligned} \text{ако } b_1 = 1, b_4 = 0 &\Rightarrow b_2 = -3, b_3 = 2 \\ \text{ако } b_1 = 0, b_4 = 1 &\Rightarrow b_2 = 10, b_3 = -5 \end{aligned}$$

Получихме, че един базис на пространството B_3 образуват векторите $t_1 = (1, -3, 2, 0)$ и $t_2 = (0, 10, -5, 1)$.

Задача 4. Авиокомпания прилага следните правила при регистрация (check-in) на пътници:

1. Ако летите с икономичен билет, има възможност да преминете към бизнес класа, особено ако сте притежател на златна карта в програма за често пътуващи пътници на авиокомпанията.
2. Ако не сте притежател на златна карта, има възможност да не бъдете допуснати до полета, ако е препълнен и се регистрирате (check-in) късно.

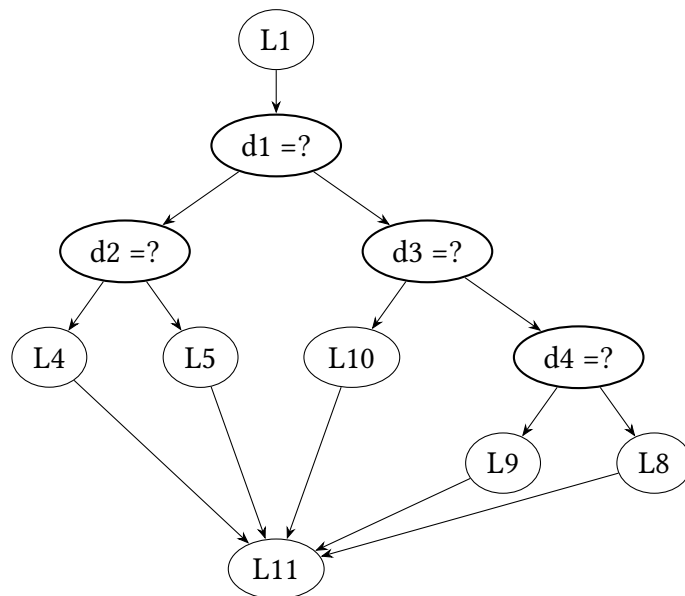
Да се опише с псевдо-код начинът на опериране на авиокомпанията при регистрация (check-in).
Да се конструира модел за тестване с граф на управляващия поток, от който да се дефинират тестови сценарии за структурно тестване (тестване по метода на бялата кутия).

Да се опишат основните стъпки при конструиране на модела.

Колко процента е покритието на следния тестов сценарий: “Притежател на златна карта, който използва възможност за преминаване към бизнес класа”? Да се дефинират тестови сценарии за постигане на 100% покритие.

Примерно решение:

- L1: input (start check-in)
- L2: if gold card then
- L3: if business class full then
- L4: economy boarding pass
- L5: else upgraded boarding pass
- L6: else if economy full then
- L7: if business class full then
- L8: bump off flight
- L9: else upgrade boarding pass
- L10: else economy boarding pass
- L11: output (end check-in)



Последователност на конструиране на граф на управляващия поток:

- асоцииране на обработващите възли с изразите за **присвояване, извикване на процедури или функции**;
- асоцииране на възлите за взимане на решение с изразите за **условен преход** “if-then-else” или “if-then”, или **множествено разклонение** “switch-case”;
- създаване на специален тип възли за разклонение и асоциирането им с **изразите за цикъл**;
- асоцииране на началния и крайния възел на графа с **първия и последния израз в програмата**.

Покритието на следния тестов сценарий: „Притежател на златна карта, който използва възможност за преминаване към бизнес класа“ е **20%**.

За постигане на 100% покритие се очаква дефиниране на 5 тестови сценария:

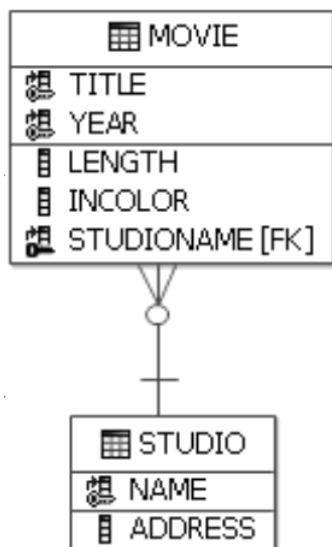
- ТС1: Притежател на златна карта, който използва възможност за преминаване към бизнес класа.
- ТС2: Пътник без златна карта, който остава в икономична класа.
- ТС3: Пътник, който не е допуснат до полет.
- ТС4: Притежател на златна карта, който остава в икономична класа.
- ТС5: Пътник без златна карта, който използва възможност за преминаване към бизнес класа.

Критерии за оценяване:

- Псевдо-код на функционалността за опериране на авиокомпанията при регистрация (check-in): 1,5 т.
- Модел за тестване с граф на управляващия поток: 4 т.
- Описани стъпки за конструиране на модела: 2 т.
- Определен процент на покритие на примерния тестов сценарий: 0,5 т.
- Дефинирани тестови сценарии за 100% покритие: 2 т.

Сумата от постигнатите точки се закръгля до цяло число.

Задача 5. Дадена е базата от данни Movies, в която се съхранява информация за филми и филмови студиа, които ги произвеждат.



Таблицата Studio съдържа информация за филмови студиа:

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

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

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

1) Да се напише заявка, която извежда заглавията и дължините в минути на всички цветни филми без най-дългия цветен и без тези с неизвестна дължина. Ако има няколко филма с максимална дължина, нито един от тях не трябва да бъде изведен.

2) Да се посочи коя от следните заявки извежда имената на тези студиа, които нямат филми или са снимали филми само в една единствена година:

A) `SELECT name
FROM Studio
WHERE name NOT IN (SELECT studioname
FROM Movie)
OR COUNT(DISTINCT year) = 1;`

B) `SELECT name
FROM Studio
LEFT JOIN Movie ON name = studioname
GROUP BY name
HAVING COUNT(DISTINCT year) <= 1;`

В) `SELECT studioname
FROM (SELECT studioname, year
FROM Movie
GROUP BY studioname, year) Years
GROUP BY studioname
HAVING COUNT(*) <= 1;`

Г) `SELECT name
FROM Studio
WHERE NOT EXISTS (SELECT * FROM Movie)
UNION
SELECT studioname
FROM Movie
GROUP BY studioname
HAVING COUNT(DISTINCT year) = 1;`

Примерно решение на подзадача 1:

```
SELECT title, length
FROM Movie
WHERE incolor = 'Y'
      AND length < (SELECT MAX(length)
                    FROM Movie
                    WHERE incolor = 'Y');
```

Критерии за оценяване:

- 1) Общо 5 т., от които
 - 1 т. за коректни SELECT и FROM клаузи на външната заявка, както и WHERE клауза на външната заявка без проверка на дължина;
 - 3 т. за коректна подзаявка за намиране на най-дългите цветни филми;
 - ако подзаявката намира най-дългите филми изобщо, се дава само 1 т. за този критерий;
 - ако подзаявката намира точно един най-дълъг филм, се дават само 2 т. за този критерий;
 - 1 т. за правилна релация с резултата от подзаявката (в конкретния пример това са <, <>, !=, NOT IN).

- 2) Единственият верен отговор е В). Посочването на този отговор носи 5 т., а посочването на грешен отговор или комбинация от отговори носи 0 т.

Задача 6. Вашата фирма е сключила договор с клиент за разработване на облачна система за управление на доставките. Задачата ви е да подготвите план за работа, при което:

1. да се дефинират работни пакети и да се посочат задачите в тях, като е необходимо всяка задача да води до специфичен краен резултат, който може да се валидира и контролира от клиента;
2. за всеки работен пакет да се определят какви материални и нематериални ресурси и експертиза са необходими за този проект (директни разходи на фирмата).

Примерно решение:

1. Работни пакети:

- **РП1 — Анализ на областта:** проучване на теорията и практиката за управление на веригите на доставка; моделиране на бизнес процесите на клиента за управление на доставките; формулиране на изисквания към системата;
- **РП2 — Разработване на софтуер:** проектиране на системата; програмиране и тестване на системата; разработване и попълване на база данни с информация на клиента;
- **РП3 — Комуникация с клиента:** семинар за валидиране на изискванията към системата; тестване на прототип; семинар за обучение на клиента;
- **РП4 — Управление на проекта:** планиране и контрол; финансово управление; срещи и комуникация на ръководния екип; отчетност и документация.

2. Материални и нематериални ресурси и експертиза:

- **РП1:** бизнес анализатори, ИТ експерти, експерти по управление на доставките, транспорт до клиента, офис консумативи;
- **РП2:** програмисти (може да изброяват: архитекти, дизайнери, тестери и др.), достъп до облачна среда;
- **РП3:** организатори, лектори, зали с мултимедия, кетъринг, материали за участниците, офис консумативи;
- **РП4:** мениджър, финансист, административен персонал, офис консумативи.

Критерии за оценяване:

1. Дефинирани работни пакети с посочени задачи в тях, като всяка задача води до специфичен краен резултат, който може да се валидира и контролира от клиента – макс. 6 точки.
2. За всеки работен пакет са определени необходимите за проекта материални и нематериални ресурси и експертиза – макс. 4 точки.
 - *Забележка:* Не се дават точки за втората част от задачата, ако не са дефинирани работни пакети и задачи в първата част.

Задача 7. Нека $\Sigma = \{0, 1\}$. За език $L \subseteq \Sigma^*$ казваме, че дума $w \in (\Sigma \cup \{\#\})^*$ е *L-половинчата*, ако всяка поддума на w от вида $v = \#u\#$ с $u \in \Sigma^*$ има свойството, че $u \cdot u \in L$.

Вярно ли е, че за всеки регулярен език $L \subseteq \Sigma^*$, езикът:

$$L_{\text{half}} = \{w \in (\Sigma \cup \{\#\})^* \mid w \text{ е } L\text{-половинчата}\}$$

е регулярен? Отговорът да се обоснове.

Примерно решение:

Да, вярно е!

Нека L е фиксиран регулярен език над Σ . Тъй като регулярните езици са затворени относно допълнение, достатъчно е да докажем, че $\overline{L_{\text{half}}} = (\Sigma')^* \setminus L_{\text{half}}$ е регулярен, където $\Sigma' = \{0, 1, \#\}$.

За дума $w \in \Sigma'^*$ е ясно, че $w \in \overline{L_{\text{half}}}$ точно когато $w \notin L_{\text{half}}$, тоест точно когато w съдържа поддума $v = \#u\#$, за която $u \in \Sigma^*$, но $u \cdot u \notin L$.

Тогава, ако означим с $L_2 = \{u \in \Sigma^* \mid u \cdot u \in L\}$, а $\overline{L_2} = \Sigma^* \setminus L_2$ получаваме, че:

$$\overline{L_{\text{half}}} = \Sigma'^* \cdot \{\#\} \cdot \overline{L_2} \cdot \{\#\} \cdot \Sigma'^*. \quad (1)$$

Тъй като регулярните езици са затворени относно конкатенация и итерация, а очевидно Σ' и $\{\#\}$ са регулярни, остава да покажем, че $\overline{L_2}$ е регулярен.

Но регулярните езици са затворени относно допълнение, така че е достатъчно да покажем, че L_2 е регулярен. За целта, нека $\mathcal{A} = \langle \Sigma, Q, s, \delta, F \rangle$ е тотален краен детерминиран автомат за езика L . От Теоремата на Клини такъв има, защото L е регулярен.

Тогава $u \in L_2$ точно когато $u \cdot u \in L$, тоест $\delta^*(s, u \cdot u) \in F$, точно когато $\delta^*(\delta^*(s, u), u) \in F$, откъдето получаваме, че

$$u \in L_2 \iff \exists q \in Q (\delta^*(s, u) = q \& \delta^*(q, u) \in F). \quad (2)$$

Нека $\overleftarrow{\mathcal{A}}_q = \langle \Sigma, Q, q, \delta, F \rangle$, а $\overrightarrow{\mathcal{A}}_q = \langle \Sigma, Q, s, \delta, \{q\} \rangle$. Тогава $\delta^*(s, u) = q$ е еквивалентно на $u \in L(\overleftarrow{\mathcal{A}}_q)$, а $\delta^*(q, u) \in F$ – на $u \in L(\overrightarrow{\mathcal{A}}_q)$.

Така получихме, че:

$$u \in L_2 \iff \exists q \in Q (u \in L(\overleftarrow{\mathcal{A}}_q) \& u \in L(\overrightarrow{\mathcal{A}}_q)) \iff u \in \bigcup_{q \in Q} (L(\overleftarrow{\mathcal{A}}_q) \cap L(\overrightarrow{\mathcal{A}}_q)). \quad (3)$$

Отново от Теоремата на Клини, езиците $L(\overleftarrow{\mathcal{A}}_q)$ и $L(\overrightarrow{\mathcal{A}}_q)$ са регулярни и тъй като регулярните езици са затворени относно допълнение и обединение, то езикът L_2 е регулярен. Оттук $\overline{L_2}$, а значи и $\overline{L_{\text{half}}}$ и L_{half} са регулярни.

Критерии за оценяване:

- 1 т. — за свеждане към $\overline{L_{\text{half}}}$;
- 2 т. — за (1), от които 1 т. — за описание с думи, съответно доказателство;
- 1 т. — за свеждане на $\overline{L_2}$ към L_2 ;
- 2 т. — за доказателство на (2);
- 1 т. — за въвеждане на $\overleftarrow{\mathcal{A}}_q$ и $\overrightarrow{\mathcal{A}}_q$;
- 2 т. — за доказателство на (3);
- 1 т. — за довършване.

Алтернативна схема за първата част: Езикът L_{half} може да се изрази и така:

$$\begin{aligned} L_{\text{half}} &= \Sigma^* \cup \Sigma^* \cdot \{\#\} \cdot (L_2 \cdot \{\#\})^* \cdot \Sigma^*, \text{ или} \\ L_{\text{half}} &= \Sigma^* \cup \Sigma^* \cdot (\{\#\} \cdot L_2)^* \cdot \{\#\} \cdot \Sigma^*, \text{ или} \\ L_{\text{half}} &= \Sigma^* \cdot (\{\varepsilon\} \cup \{\#\} \cdot (L_2 \cdot \{\#\})^*) \cdot \Sigma^* \text{ и т.н.} \end{aligned}$$

Изразявания като горните се оценяват с 4 точки, които съответстват на първите 4 точки от горната схема. Тези точки се разпределят така:

- 2 т. — за декларация на съответното регулярно изразяване;
- 2 т. — за доказателство, че съответното изразяване е коректно.

Забележка: Точки **не са дават** при грешно изразяване, например:

$$\begin{aligned}L_{\text{half}} &= \Sigma^* \cdot \{\#\} \cdot (L_2 \cdot \{\#\})^* \Sigma^*, \text{ или} \\L_{\text{half}} &= \Sigma^* \cup \Sigma^* \cdot (\{\#\} \cdot L_2 \cdot \{\#\})^* \cdot \Sigma^*, \text{ или} \\L_{\text{half}} &= \Sigma^* \cdot (\{\varepsilon\} \cup \{\#\} \cdot (L_2 \cdot \{\#\})^*) \text{ и т.н.}\end{aligned}$$

Решението обаче може да получи до 6 точки според оценяването за регулярността на езика L_2 .

Забележка: Конструкции на крайни автомати, за които се твърди, че разпознават L_{half} , без едно от двете: (i) ясна семантика на състоянията или (ii) пълно доказателство за коректност, се оценяват с 0 точки. Самият едносричен отговор носи 0 точки.

Задача 8. Да се намери неопределеният интеграл

$$\int (x + \operatorname{tg}^2 x) \sin^2 x \, dx, \quad x \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

Примерно решение:

Започваме със свеждането

$$\int (x + \operatorname{tg}^2 x) \sin^2 x \, dx = \int x \sin^2 x \, dx + \int \frac{\sin^4 x}{\cos^2 x} \, dx. \quad (*)$$

Намираме всеки от двата неопределени интеграла горе вдясно. За първия имаме

$$\begin{aligned} \int x \sin^2 x \, dx &= \frac{1}{2} \int x(1 - \cos 2x) \, dx = \frac{1}{2} \int x \, dx - \frac{1}{4} \int x \cos 2x \, d(2x) \\ &= \frac{x^2}{4} + \operatorname{const} - \frac{1}{4} \int x \, d(\sin 2x) \quad (\text{вносяме } \cos 2x \text{ под знака на диференциала}) \\ &= \frac{x^2}{4} + \operatorname{const} - \frac{1}{4} \left(x \sin 2x - \int \sin 2x \, dx \right) \quad (\text{интегрираме по части}) \\ &= \frac{x^2}{4} - \frac{1}{4} x \sin 2x - \frac{1}{8} \cos 2x + \operatorname{const}. \end{aligned}$$

За втория интеграл вдясно на (*) имаме

$$\begin{aligned} \int \frac{\sin^4 x}{\cos^2 x} \, dx &= \int \frac{(1 - \cos^2 x)^2}{\cos^2 x} \, dx \\ &= \int \frac{dx}{\cos^2 x} - 2 \int dx + \int \cos^2 x \, dx \\ &= \operatorname{tg} x - 2x + \operatorname{const} + \frac{1}{2} \int (1 + \cos 2x) \, dx \\ &= \operatorname{tg} x - 2x + \operatorname{const} + \frac{1}{2} \int dx + \frac{1}{4} \int \cos 2x \, d(2x) \\ &= \operatorname{tg} x - 2x + \frac{x}{2} + \frac{1}{4} \sin 2x + \operatorname{const} \\ &= \operatorname{tg} x - \frac{3x}{2} + \frac{1}{4} \sin 2x + \operatorname{const}. \end{aligned}$$

Окончателно получаваме

$$\int (x + \operatorname{tg}^2 x) \sin^2 x \, dx = \frac{x^2}{4} - \frac{3x}{2} - \frac{1}{4} x \sin 2x + \frac{1}{4} \sin 2x - \frac{1}{8} \cos 2x + \operatorname{tg} x + \operatorname{const}.$$

Критерии за оценяване: Общо 10 т., от които:

- за намиране на $\int x \sin^2 x \, dx$: 4 т., в това число:
 - за понижаване на степента на \sin : 1 т.,
 - за намиране на $\int x \, dx$: 1 т.,
 - интегриране по части: 2 т.;
- за намиране на $\int \frac{\sin^4 x}{\cos^2 x} \, dx$: 4 т., в това число:
 - подходящо разлагане на подинтегралната функция: 1 т.,
 - за намиране на $\int \frac{dx}{\cos^2 x}$: 1 т.,
 - за намиране на $\int dx$: 1 т.,
 - за намиране на $\int \cos^2 x \, dx$: 1 т.;
- окончателен отговор: 2 т. (при отсъствие на интеграционната константа **не** се присъждат).

Чернова