

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.** Да се дефинира рекурсивна функция на езика C++

```
void print_backwards(const char* begin, const char* end).
```

Параметрите `begin` и `end` са указатели към елементи на буфер от символи, като  $end \geq begin$ . Символен низ с начало `begin` и край `end` наричаме последователността от символи, започваща със символа, намиращ се на адрес `begin` и завършваща със символа на адрес `end-1`, включително, дори и ако някой от тези символи е с код 0. Дума в такъв символен низ наричаме всяка непразна подпоследователност от стандартни символи (с код  $> 32$ ), ограничена от двете страни или от някой от краищата на низа, или от специален символ (с код  $\leq 32$ ).

Функцията `print_backwards` да извежда на стандартния изход всички думи в низа с начало `begin` и край `end` в ред, обратен на срещането им в низа, и разделени с точно един интервал.

Да се дефинира функция

```
void print_backwards(const char* string),
```

която прилага рекурсивната функция `print_backwards` върху символния низ `string`.

Да се демонстрира извикването на последната функция в кратка програма.

Пример: Извикването `print_backwards("I\tneed a break!");` извежда низа "break! a need I".

---

**Задача 2.** а) Да се дефинира клас `Interval`, описващ интервала от цели числа  $[a, a + k]$  (с тип на елементите `int`) по зададени чрез конструктора на класа числа  $a$  и  $k \geq 0$ . Чрез обекти от клас `Interval` да може да се обхождат последователно елементите на интервала с помощта на следните методи и операции:

1. Префиксна операция `*` за получаване на текущия елемент на интервала.
2. Префиксна и съответна постфиксна операция `++` за преминаване към следващия елемент на интервала.
3. Метод `reset`, който задава за текущ първия елемент от интервала.
4. Булев метод `last` за проверка дали е достигнат най-големият елемент на интервала.

Обхождането на интервала започва с най-малкия му елемент. Всяко изпълнение на операцията `++` води до преминаване към следващия елемент в интервала. По този начин, ако операцията `++` е изпълнена точно  $i$  ( $0 \leq i \leq k$ ) пъти, чрез операцията `*` ще бъде достъпен елементът  $a + i$ . Ако обхождането на интервала вече е достигнало най-големия му елемент, операцията `++` да няма ефект.

Пример: Следният цикъл

```
Interval i(0,10);
std::cout << *i << std::endl;
do {
    ++i;
    std::cout << *i << std::endl;
} while (!i.last());
```

ще изведе числата от 0 до 10.

б) Да се дефинира клас `SquaredInterval`, наследник на клас `Interval`, който предефинира операцията `*` така, че с нея да се получава втората степен на текущия елемент на интервала. Пример, аналогичен на горния, но за клас `SquaredInterval`, ще изведе числата 0, 1, 4, ..., 100.

в) Да се дефинира функция, която по подаден масив от интервали от тип `SquaredInterval` го сортира възходящо спрямо максималната стойност, която може да бъде получена чрез прилагането на операцията `*` в съответния интервал. Да се демонстрира използването на тази функция в кратка програма. Няма ограничения за използваните метод, алгоритъм или функции на стандартната библиотека.

Решение:

**Задача 3.** Даден е следният псевдо-код:

```
1      Program BestInterest
2      Interest, Base Rate, Balance: Real
3
4      Begin
5      Base Rate = 0.035
6      Interest = Base Rate
7
8      Read (Balance)
9      If Balance > 1000
10     Then
11         Interest = Interest + 0.005
12     If Balance < 10000
13         Then
14             Interest = Interest + 0.005
15         Else
16             Interest = Interest + 0.010
17         Endif
18     Endif
19
20     Balance = Balance * (1 + Interest)
21
22     End
```

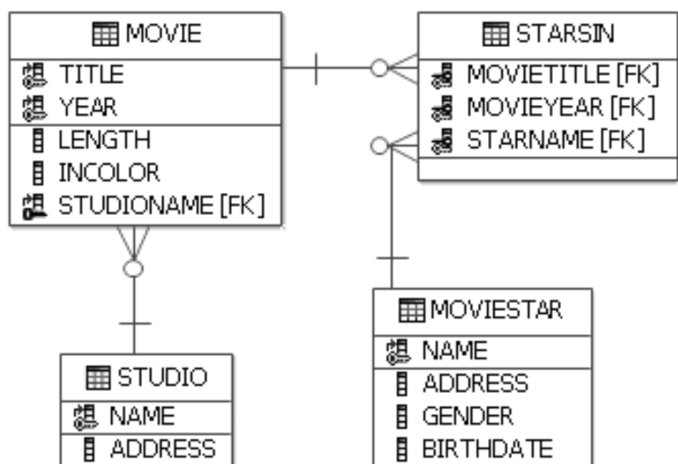
- а) Да се конструира граф на управляващия поток за псевдо-кода.
- б) Да се опишат основните стъпки при конструиране на графа.
- в) Да се дефинират тестови сценарии, като се използва конструираният граф, за да се получи пълно покритие на изразите (statements).
- г) Даден е следният псевдо-код:

```
1      Begin
2      Read Time
3      If Time < 12 Then
4          Print(Time, "am")
5      Endif
6      If Time > 12 Then
7          Print(Time -12, "pm")
8      Endif
9      If Time = 12 Then
10         Print(Time, "noon")
11     Endif
12     End
```

Да се определи какво ще бъде тестовото покритие на решенията (decisions) в проценти при тестови сценарии Time = 11 и Time = 15. Отговорът да се обоснове.

---

**Задача 4.** Дадена е базата от данни Movies, в която се съхранява информация за филми, филмови студия, които ги произвеждат, както и актьорите, които участват в тях.



Таблицата Studio съдържа информация за филмови студия:

- 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 чернобял
- studioname — име на студио, външен ключ към Studio.name;

Таблицата MovieStar съдържа информация за филмови звезди:

- name — име, първичен ключ
- address — адрес
- gender — пол, 'M' за мъж (актьор) и 'F' за жена (актриса)
- birthdate — рождена дата.

Таблицата StarsIn съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ. Атрибутите 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 към MovieStar.name.

а) Да се напише заявка, която извежда имената и рождените дати на всички филмови звезди, чието име не съдържа "Jr. " и са играли в поне един цветен филм. Първо да се изведат най-младите звезди, а звезди, родени на една и съща дата, да се изведат по азбучен ред.

б) Да се напише заявка, която извежда следната информация за всяка актриса, играла в най-много 6 филма:

- име;
- рождена година (напр. ако актрисата е родена на 1.1.1995 г., в колоната да пише 1995);
- брой различни студия, с които е работила.

Ако за дадена актриса няма информация в какви филми е играла, за нея също да се изведе ред с горната информация, като за брой студия се изведе 0.

**Задача 5.** Софтуерна система свързва потребители и застрахователна брокерска компания за сключване на застраховка Гражданска отговорност (ГО). Системата работи и с оператори, които са оторизирани представители на застрахователни дружества.

1. През системата нерегистриран потребител има възможност да поиска оферта за ГО.
2. За генериране на оферта за ГО от потребителя се изискват данни за превозното средство (марка, модел, година, номер, мощност), както и данни за шофьора (име, телефон, номер на шофьорска книжа).
3. При генериране на оферта за ГО се прави справка в системата на МВР за нарушения на шофьора.
4. Нерегистриран потребител може да сключи ГО онлайн след електронно заплащане на съответната сума.
5. Системата поддържа плащане с ePay и банкова карта.
6. При сключване на ГО е необходим електронен подпис на шофьора, като за целта се извършва проверка за валидност в националната система за електронни подписи.
7. Системата дава възможност на потребителите да се регистрират чрез имейл адрес и парола.
8. Регистрираните потребители получават 10% отстъпка при сключване на три и повече ГО чрез системата.
9. Регистриран потребител може да поиска SMS напомняне от системата за изтичане на ГО.
10. Операторите се удостоверяват в системата използвайки корпоративната система на съответното застрахователно дружество.
11. Оператор има право да променя параметрите на генерираните оферти.

Да се състави диаграма на случаите на употреба (Use Case) на системата за дадените изисквания. Да се отбележат съществуващите връзки между потребителските случаи.

---

**Задача 6.** Вярно ли е, че за всеки регулярен език  $L \subseteq \{0, 1\}^*$ , езикът:

$$\{w^{|w|} \mid w \in L\}$$

е регулярен? Отговорът да се обоснове.

**Задача 7.** В урна има 5 бели, 7 зелени и 3 червени топки. На всеки опит вадим от урната едновременно две топки, записваме цвета им, след което връщаме топките обратно в урната. Дефинираме събитие

$$A = \{\text{Изтеглени са една бяла и една зелена топка}\}.$$

а) Да се определи вероятността на  $A$  при извършване на един опит.

б) Нека  $X$  е броят на сбъдванията на събитието  $A$  при провеждане на 5 опита, да се пресметнат:  $P(X = 3)$ , математическото очакване  $EX$  и дисперсията  $DX$ .

в) Нека белите топки са 5, зелените 7, но броят на червените е  $Z$ . Каква трябва да бъде стойността на  $Z$ , така че средният брой на неуспешните опити до първото сбъдване на събитието  $A$  да бъде точно пет? Отговорът да се обоснове.



**Чернова**