

**СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”**



**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

**ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАТИКА”**

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
10.09.2009 г.**

Време за работа – 3 часа

Драги абсолвенти:

- Пишете само на предоставените ви листове без да ги разкопчавате;
- Попълнете факултетния си номер горе вдясно на всички страници;
- Решението на всяка от задачите се разполага само в пространството от края на условието на тази задача до началото на условието на следващата задача (могат да се използват и двете страни на листата).

Изпитната комисия ви пожелава успешна работа.

Задача 1. (9т.) Да се дефинира на C++ функцията

`void merge (int a1[], int a2[], int n, int result[]);`

където a_1 и a_2 са масиви с n елемента, а $result$ е масив с големина поне $2 \times n$. Елементите на a_1 и a_2 са подредени в ненамаляващ ред. Функцията да запише всички елементи на a_1 и a_2 в масива $result$ така, че елементите на $result$ също да са подредени в ненамаляващ ред.

Пример: ако a_1 съдържа в посочения ред елементите 1, 3, 3 и 5, а a_2 съдържа в посочения ред елементите 0, 2, 4, 6, то след изпълнение на функцията, $result$ ще съдържа елементите 0, 1, 2, 3, 3, 4, 5, 6 в този ред.

Задача 2. (8т.) Да се попълни в празните полета изходът от съответните програмни конструкции.

<pre>#include <iostream> using namespace std; class Base { public: Base () {cout << "Base::Base()\n"; } Base (const Base&) {cout << "Base::copy()\n"; } virtual void f () {cout << "Base::f()\n"; } virtual ~Base () {cout << "Base::~\n"; } }; class Der : public Base { public: Der () {cout << "Der::Der()\n"; } Der (const Der&) {cout << "Der::Copy()\n"; } void f () {cout << "Der::f()\n"; } ~Der () {cout << "Der::~\n"; } }; void g1 (Base a) { cout << "F: "; a.f(); } void g2 (Der a) { cout << "F: "; a.f(); } void g3 (Base& a) { cout << "F: "; a.f(); }</pre>	<pre>void main () { cout << "1:\n"; Der d; <div style="border: 1px solid black; height: 50px; width: 100%;"></div> cout << "2:\n"; g1 (d); <div style="border: 1px solid black; height: 50px; width: 100%;"></div> cout << "3:\n"; g2 (d); <div style="border: 1px solid black; height: 50px; width: 100%;"></div> cout << "4:\n"; g3 (d); <div style="border: 1px solid black; height: 50px; width: 100%;"></div> cout << "5:\n"; }</pre>
---	--

Задача 3. (12т.) Компресирано представяне на безкраен поток от числа наричаме такъв друг поток, за който неколкото повторение на числото 0 е заменено от точкова двойка с първи елемент 0 и втори – броят на повторенията.

Например, ако началото на един поток е (0 0 0 1 0 0 7 3 ...), компресираното му представяне има начало ((0.3) 1 (0.2) 7 3 ...)

Реализирайте двойка функции на Scheme, които по зададен безкраен поток, съдържащ някакви стойности, намират компресираното му представяне и обратно.

Задача 4. (15т.) Да означим с φ_1 , φ_2 и φ_3 следните предикатни формули:

$\varphi_1: \forall X \forall Y \forall Z (p(X, Y) \& p(Y, Z) \Rightarrow p(X, Z))$

$\varphi_2: \exists X \exists Y (p(X, Y) \& p(Y, X))$

$\varphi_3: \forall X \neg p(X, X)$.

А. Покажете, че множеството $\{\varphi_1, \varphi_2\}$ е изпълнимо (т.е. има структура, в която са верни едновременно φ_1 и φ_2).

Б. С метода на резолюцията докажете, че множеството $\{\varphi_1, \varphi_2, \varphi_3\}$ не е изпълнимо.

Задача 5. (15т.) Дадено е множеството от булеви функции:

$$A = \{f_1(x, y, z) = x \rightarrow y; f_2(x, y, z) = x \equiv y; f_3(x, y, z) = xy \oplus yz\}$$

а) Попълнете в таблицата вектор-стълба на всяка от горните функции:

x	y	z	$f_1(x, y, z)$	$f_2(x, y, z)$	$f_3(x, y, z)$

б) Попълнете таблицата, като със звездичка означите кога съответната функция не принадлежи на съответното множество:

	T_0	T_1	S	M	L
$f_1(x, y, z)$					
$f_2(x, y, z)$					
$f_3(x, y, z)$					

в) Определете дали множеството A е базис в множеството на всички булеви функции.

Задача 6. (10 т.) Напишете вясно какво ще бъде изведено на стандартния изход като резултат от изпълнението на дадения по-долу фрагмент от програма на C:

```
main()
{
.....
    int i, pid, flag = 2, status;

    pid = fork();
    if ( pid ) { - - flag; wait(&status); }
    else for ( i=1; i <= 2; i++ ) ++flag;
    ++flag;
    if ( flag == 3 ) exit ( - - i );
    printf ( "flag = %d\n",flag);
.....
}
```


Задача 7. (10т.) В базата от данни със схема:

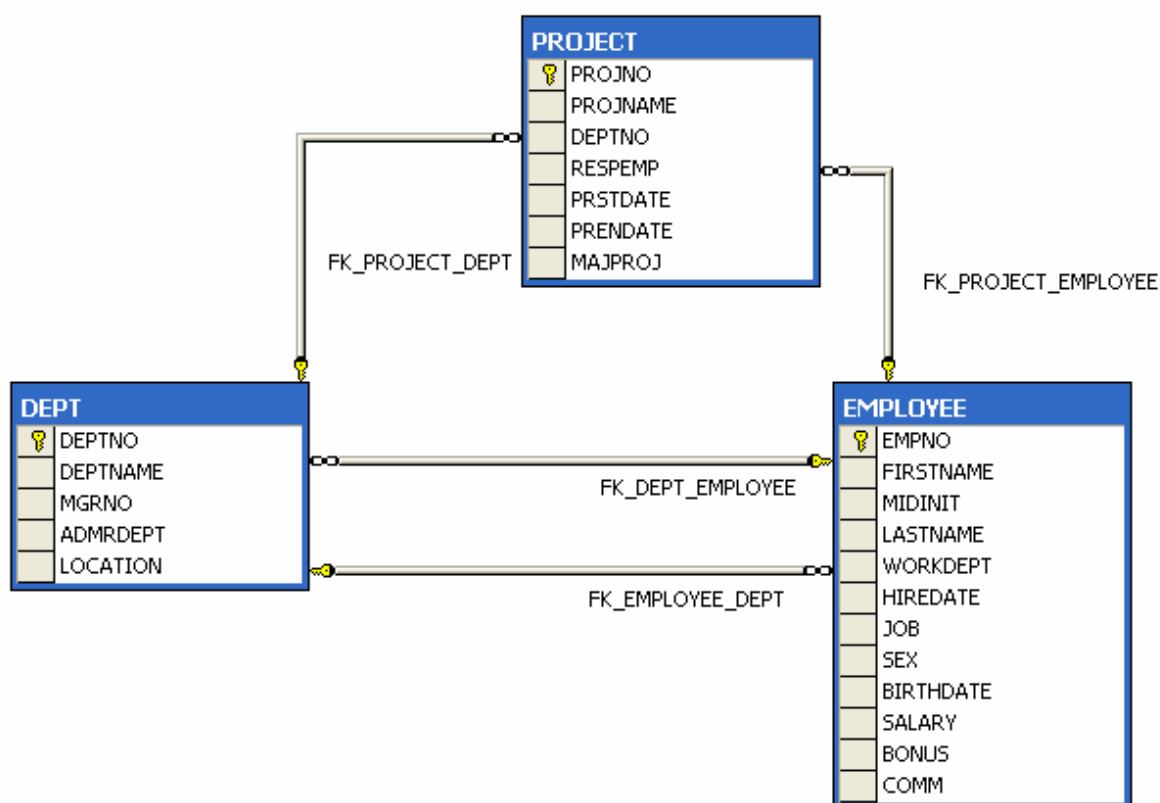
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

```
EMPLOYEE (EMPNO, FIRSTNAME, MIDNAME, LASTNAME, WORKDEPT, HIREDATE, JOB, SEX, BIRTHDATE, SALARY, BONUS, COMM);
```

```
PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE, PRENDATE, MAJPROJ);
```

се съхранява информация за департаменти, работници и проекти:

- Таблицата DEPT съдържа информация за номер и име на департамент и за номер на работник (MGRNO) – менажер за този департамент;
- Таблицата EMPLOYEE съдържа информация за номер на работник, лични данни, длъжност (JOB), заплата (SALARY) и номера на департамента, към който работи (WORKDEPT);
- Таблицата PROJECT съдържа информация за номер и име на проект, начална и крайна дата на проекта, департамента, към който е проектът (DEPTNO), и номера на работника – менажер на проекта (RESPEMP).



А. Напишете заявка, която извежда името на департамента, броя на проектите и сумарната заплата на менажерите на проекти от този департамент.

Б. Като използвате заявката от т. А, напишете заявка, която извежда името на департамента с най-много проекти.

Забележка: Ако има повече департаменти с равен максимален брой на проекти, те също да се изведат.

Задача 8. (5т.) Даден е лабиринт от стаи. Всяка стая има номер, който я идентифицира еднозначно. Търси се най-кратък път от стая X до стая Y.

Да се напише програма на C++/C#/Java, която по зададени числа X и Y да връща номерата на стаите, през които минава най-краткият път от X до Y. Приемат се за дадени функциите:

- `public static int getEast(int id);` - връща номера на стаята, която е на изток от стаята с номер `id`.
- `public static int getWest(int id);` - връща номера на стаята, която е на запад от стаята с номер `id`.
- `public static int getNorth(int id);` - връща номера на стаята, която е на север от стаята с номер `id`.
- `public static int getSouth(int id);` - връща номера на стаята, която е на юг от стаята с номер `id`.

В случай, че не съществува стая в съответната посока, всяка от тези функции връща стойност 0. Ако някоя от функциите се извика с аргумент 0, върнатата стойност е 0.

1	34	45	67	29	48
65				78	73
77	23		57	44	28
66	46				26
34	65	76	99	96	93

Примерен вход: X=1, Y=48

Изход: 1, 34, 45, 67, 29, 48

Примерен лабиринт

Спецификация на входа и изхода на програмата:

Вход:

- X – номер на стая, от която се стартира търсенето;
- Y – номер на стая, до която трябва да се достигне.

Изход:

- Най-кратък път от X до Y, изписан като поредица от стаите, през които трябва да се мине. При повече от един най-кратки пътища, като резултат се изписва само един от тях.

