

ПРИМЕРНИ ЗАДАНИЯ

Дисциплина: Компютърни мрежи и комуникации

Задание 1

Малка компания е получила от Интернет доставчик адрес 192.30.50.0. Необходимо е да се раздели мрежата на 14 подмрежи с по 12 хоста във всяка подмрежа. Политиката на фирмата е адресът на сървъра в подмрежата да е първият използваем хост адрес, а адресът на рутера (default gateway), да е последният използваем адрес.

а) Каква е областта от адреси за седмата подмрежа:

б) Задайте конфигурацията на сървъра за седмата подмрежа:

IP адрес:

Подмрежова маска:

Default Gateway:

в) Какъв адрес е 192.30.50.207 – валиден хост адрес, адрес на подмрежа или broadcast адрес:

Да се приложат и направените изчисления.

Задание 2

Компания е получила от Интернет доставчик адрес 172.50.0.0. Необходимо е да се раздели мрежата на 50 подмрежи с по 820 хоста във всяка подмрежа. Политиката на фирмата е адресът на рутера (default gateway) да е първият използваем хост адрес, адресът на сървъра в подмрежата да е последният използваем адрес.

а) Каква е областта от адреси за седмата подмрежа:

б) Задайте конфигурацията на сървъра за седмата подмрежа:

IP адрес:

Подмрежова маска:

Default Gateway:

в) Какъв адрес е 172.50.50.50 – валиден хост адрес, адрес на подмрежа или broadcast адрес:

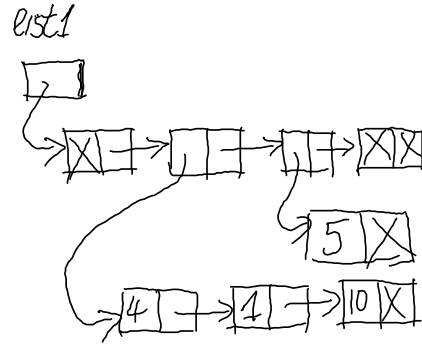
Да се приложат и направените изчисления.

Разработил: гл. ас. М. Радев

Дисциплина: Програмиране и структури от данни

Задание 1

Да се напише програма, в която чрез подпрограма се създава динамичен линеен свързан списък (ЛСС) list1, елементите на който са динамичен целочислен ЛСС (фиг.1). Да се напишат още две подпрограми, първата от които намира сумата на елементите на списък list1, а втората извежда списъка list1 върху екрана. Например, списъкът от фиг. 1 трябва да се изведе като (0,(4,1,10),(5),(0)).



Фиг.1. Пример за динамичен ЛСС list1

Трябва да се има предвид, че данните между главната програма и подпрограмите се обменят само чрез параметри.

Задание 2

Да се напише програма със следните три подпрограми:

- първата подпрограма създава целочислен динамичен линеен свързан списък;
- втората подпрограма е рекурсивна и извежда значенията на списъка в обратен ред;
- третата подпрограма също е рекурсивна и проверява дали зададено значение се съдържа в списъка.

Трябва да се има предвид, че данните между главната програма и подпрограмите се обменят само чрез параметри.

Разработил: доц. д-р Ф. Филипов

Дисциплина: Въведение в програмирането

Заданията могат да включват едно или няколко условия от следния вид:

- анализ на готови фрагменти от код или цели програми – откриване на заложения алгоритъм и неговите цели; намиране на резултата, който би се получил; откриване на грешки
- разработване (на лист) на кратки фрагменти програмен код, реализиращи поставен проблем (напр. кратка потребителска функция).

Задание 1

1. Дефинирайте функция, която да проверява дали масив от числа е симетричен относно средата си. Функцията да приема 2 параметъра – брой на елементите и указател към началото на масива и да връща стойност 1, ако редицата е симетрична и 0 – ако не е.

2. Опишете какво изчислява следният фрагмент от код:

```
int n, f=1;
cin>>n;
while(n) f*= n--;
cout<<f;
```

3. Дефинирайте рекурсивна функция, която намира сумата на числата от 1 до n.

Разработил: доц. д-р В. Сълов

Дисциплина: Обектноориентирано програмиране

Задача 1.

На кой ред/редове се нарушава приятелството?

```
01) class B;
02) class A {
03)     friend class B;
04)     private:
05)         void f() {
06)             B b;
07)             b.f();
08)         }
09) };
10) class B {
11)     private:
12)         void f() {
13)             A a;
14)             a.f();
15)         }
16) };
```

- a) 6 и 7
- б) 7
- в) 14
- г) 7 и 14

Задача 2.

Какво ще се изведе?

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<int> v;
    v.push_back(9);
    v[v.size()-1]=10;
    v.insert(v.begin(),0);
    v.insert(v.begin(),20);
    v.erase(v.begin()+1);
    for(int i=0; i<v.size(); i++) cout<<v[i];
    return 0;
}
```

- a) 2009
- б) 200910
- в) 2010
- г) в кода има логическа грешка

Задача 3.

Какво ще се изведе?

```
string s = "object oriented programming 2009 in C++";  
s.replace(2,4, "varna");  
cout<<s.substr(s.find("0"), 2)  
    <<s.substr(s.find_first_of("p"), 1);
```

- a) varna
- б) 2009
- в) 00p
- г) съобщение за грешка

Задача 4.

Кои от редовете във функцията main() са допустими?

```
class RealEstate {  
public:  
    void setPrice(double newPrice) { price=newPrice; }  
    void setProfit(double newProfit) { profit=newProfit; }  
    double getPrice() { return price; }  
private:  
    double price;  
    double profit;  
    double getProfit() { return profit; }  
} app, house;  
  
int main() {  
1)    app.price = 45000.99;  
2)    house.setPrice(120000.97);  
3)    double price, profit;  
4)    price = house.getPrice();  
5)    profit = house.getProfit();  
6)    profit = app.getProfit();  
7)    app = house;  
}
```

- a) 3, 4, 6
- б) 2, 5, 7
- в) 2, 3, 7
- г) 1, 2, 3

Разработил: доц. д-р П. Петров

Дисциплина: Базы от данни

(Проектиране на реляционен модел на база от данни; използване на SQL операторите за дефиниране на обекти (схеми, таблици, изгледи) на базата от данни и манипулиране с данните)

Задание 1

Компания за обучение провежда курсове, които се водят от инструктори. За всеки курс се съхранява информация за номер на курса, наименование и продължителност на курса в дни. Един и същи курс може да се води от различни инструктори. Един инструктор може да има компетенции да води няколко курса, но за даден период от време може да води само 1 курс. За всеки инструктор се пазят данни за него – номер, име и фамилия, град, адрес, телефонен номер и за кои курсове има компетенции да ги води. В каталога с обявите на курсовете се посочва кой курс в кой период от време ще се провежда и при кой инструктор. Един и същи курс може да се обявява в различно време на провеждане и за един и същи период могат да се провеждат едни и същи курсове от различни инструктори. Курсистите избират кой курс, в кой период от време и при кой инструктор да посещават. Курсистите могат да ходят на няколко курса и много от тях го правят. За всеки курсист се пазят данни за него – номер, име и фамилия, град, адрес, телефонен номер. Някои от курсистите и инструкторите не дават телефонните си номера.

Проектирайте реляционен модел на база от данни за тази дейност на компанията и опишете релациите (таблиците) и връзките между тях.

Задание 2

За управление на доставките и продажбите на стоки в търговска фирма е създадена база от данни, която съдържа следните таблици:

- Products (**ProductId**, Name, GroupId) – Стоки; GroupId външен ключ.
- Groups (**GroupId**, Name) – Групи стоки
- Customers(**CustomerId**, CompanyName, City, Address) - Клиенти
- Invoices(**InvoiceId**, InvoiceNumber, InvoiceDate, CustomerId, InvoiceType) – Фактури; CustomerId външен ключ; InvoiceType – вид на фактурата в зависимост дали е за доставка или продажба на стоката приема стойности 1 или 2 съответно).
- InvoicesDetails(**InvoiceId**, **ProductId**, Quantity, Price) – детайлни редове за доставените или продадени стоки по всяка фактура; InvoiceId – външен ключ за фактури, ProductId – външен ключ за Стоки.

За всяка група стоки да се изведе справка, която да показва стойността на наличното количество от стоките (изчислява се като разлика между общата стойност на доставените стоки и общата стойност на продадените стоки за тези стоки, за които наличното количество е > от нула).

Разработил: доц. д-р М. Кашева