

Обектно-ориентирано програмиране

спец. Информатика, 2013/2014

Задачи

за задължителна самоподготовка

№3

[1] **Задача 1.4.24.** Да се дефинира функция, която създава копие на символен низ. Функцията да се грижи за заделянето на памет за новия низ.

[2] **Задача 1.4.25.** Да се дефинира функция, която преобразува положително цяло число в съответния му символен низ.

[3] **Задача 1.4.30.** Обединение на два символни низа s_1 и s_2 наричаме всеки символен низ, който съдържа без повторение всички символи на s_1 и s_2 . Да се дефинира функция, която намира обединението на два символни низа.

[4] **Задача 1.4.36.** За работа със символни низове могат да бъдат използвани следните основни функции:

`char car(const char* x)`, която връща първия символ (елемент) на низа x ;

`char* cdr(char* x)`, която връща останалата част от низа x след отделянето на първия елемент на низа x ;

`char* cons(char x, const char* y)`, която връща указател към символен низ, разположен в динамичната памет и съдържащ конкатенацията на символа x със символния низ y ;

`bool eq(const char* x, const char* y)`, която връща `true`, ако низовете x и y са равни и `false`, в противен случай.

Да се дефинират описаните функции. Като се използват тези функции да се дефинират следните функции:

а) `char* reverse(char* x)`, която връща указател към символен низ, разположен в динамичната памет и съдържащ символите на `x`, записани в обратен ред;

б) `char* copy(char* x)`, която връща указател към символен низ, разположен в динамичната памет и съдържащ копие на символния низ `x`;

в) `char* car_n(char* x, int n)`, която връща указател към символен низ, разположен в динамичната памет и съдържащ първите `n` символа на символния низ `x`;

г) `char* cdr_n(char* x, int n)`, която връща останалата част от низа `x` след отделянето на първите `n` символа. Предварително е известно, че `x` притежава поне `n` символа;

д) `int number_of_char(char* x, char ch)`, която намира колко пъти символът `ch` се среща в символния низ `x`;

е) `int number_of_substr(char* x, char* y)`, която намира колко пъти символният низ `y` се среща в символния низ `x`;

ж) `char* delete_substr(char* x, char* y)`, която връща указател към символен низ, разположен в динамичната памет и съдържащ символите на низа `x`, от който са изтрети всички срещания на символния низ `y`.

[5] Задача 2.2.3. Да се намери и обясни резултатът от изпълнението на програмата:

```
#include <iostream.h>
class C
{ private:
    int a, b;
public:
    void init(int, int);
```

```

        void display() const;
    };
    void C::init(int x, int y)
    { a = x;
b = y; }

    void C::display() const
    { cout << a << " " << b << endl;
    }
    int main()
    { C c;
        c.init(5, 8);
        c.display();
        return 0;
    }

```

[6] Задача 2.2.8. Да се намери и обясни резултатът от изпълнението на програмата:

```

#include <iostream.h>
class E
{ private:
        int a, b;
    public:
        void init();
        void init(int);
        void init(int, int);
        void display() const;
};
void E::init()
{ a = b = 0;
}
void E::init(int x)
{ a = x;
}
void E::init(int x, int y)

```

```

        { a = x;
b = y; }

void E::display() const
{ cout << a << " " << b << endl;
}
int main()
{ E e1, e2, e3;
  e1.init();
  e2.init(1);
  e3.init(1, 2);
  e1.display();
  e2.display();
  e3.display();
  return 0;
}

```

[7] Задача 2.2.6. Да се дефинира клас `angle`, чрез който се намират синусът, косинусът и тангенсът на ъгъл, зададен в градуси.

[8] Задача 2.2.7. Да се дефинира клас `Employee`, който пресмята заплата на чиновник, дефиниран чрез:

- име, зададено чрез структура с полета `first` (собствено име) и `last` (фамилно име) и
- работа, зададена чрез структура с полета `hours` (часове) и `salary` (заплащане за 1 час).