

Задачи за самостоятелна работа

1. Да се дефинира структурата **Complex**, която реализира комплексно число.

Структурата да съдържа **методи** за:

- създаване на комплексно число;
- аритметика с комплексни числа – събиране, изваждане, умножение и деление;
- връщане на реалната част на комплексното число;
- връщане на имажинерната част на комплексното число;
- проверка дали комплексното число е 0;
- инициализиране на данните на структурата от потока за вход in;
- инициализиране на данните на структурата от потока за вход in, който ще представлява отворен двоичен файл;
- записване на данните на структурата в потока за изход out;
- записване на данните на структурата в потока за изход out, който ще представлява отворен двоичен файл.

Да се дефинират:

- булева функция, която установява дали числа x и y са равни;
- функция, която сортира подаден масив от комплексни числа.

Упътване: Използвайте задача 3 от седмица 5, която решихме в час.

2. Да се дефинира структура **Rectangle**, която определя правоъгълник по двете му страни.

Да се дефинират **методи**, които:

- създават и извеждат правоъгълник;
- намират радиуса на описаната около правоъгълника окръжност;
- намират лицето и периметъра на правоъгълника;
- инициализира данните на структурата от потока за вход in, който ще представлява отворен двоичен файл;
- записва данните на структурата в потока за изход out, който ще представлява отворен двоичен файл.

Да се напише **програма**, която тества така създадената структура, като:

- създава правоъгълник по зададени страни;
- извежда правоъгълника;
- намира и извежда радиуса на описаната около правоъгълника окръжност, лицето и периметъра му.

3. Структурата **Fruit**, определяща плод, е дефинирана по следния начин:

```
enum Color {ORANGE, YELLOW, RED, GREEN};
```

```
struct Fruit {  
    char* mName; // име на плод  
    Color mColor; // цвят на плод  
    bool mIsAnnual; // едногодишен плод  
    bool mIsPerennial; // многогодишен плод  
    bool mIsTropical; // тропически плод  
};
```

Да се дефинира **функция**, която:

- създава едномерен масив от плодове;
- извежда едномерния масив от плодове;
- извежда едногодишните плодове с даден цвят, записани в масива, сортирани във възходящ ред по име на плода;
- извежда многогодишните тропически плодове с даден цвят, сортирани в низходящ ред по име на плода.

4. Да се напише програма, която извежда на екрана съдържанието на подаден файл **inFile**. (командата cat в UNIX)
5. Да се напише програма, която копира файла **inFile** във файла **outFile**.
6. Да се напише програма, която конкатенира съдържанието на файловете **inFile** и **outFile** и го записва в **outFile**.
7. Да се напише програма, която намира поредния номер на най-дългия ред на даден текстов файл. Номерацията започва от 1. В случай, че има повече от един ред с дължина, равна на дължината на най-дългия ред, програмата да намира първия ред.
8. В текстов файл е записан списък от програми. Програмите са съставени от командите **MOV(X, Y)**, **ADD(X, Y)**, **MUL(X, Y)** и **SUB(X, Y)**. За всички команди X е променлива (малка латинска буква), а Y е вече дефинирана променлива или цяло число с или без знак. Една променлива се нарича *дефинирана*, ако е участвала в предишна команда MOV като първи аргумент. За всички команди без командата MOV се изисква и първият аргумент да е дефинирана променлива.

Командите извършват следните действия:

MOV(X, Y) присвоява стойността на Y на променливата X. X става дефинирана, дори да не е срещана преди това;

ADD(X, Y) прибавя стойността на Y към X;

MUL(X, Y) умножава X и Y;

SUB(X, Y) изважда Y от X.

Всички команди запазват стойността на втория си аргумент и променят стойността на първия.

Всяка програма започва с „PROGRAM <name>” и завършва с “END”, където <name> е произволна редица от малки и/или главни латински букви.

Да се напише програма, която по даден файл с програми извежда на екрана за всяка програма „Грешка”, ако съответната програма не е коректна (напр. съдържа непозната команда или се опитва да работи с недефинирана променлива). Ако в некоректна програма има няколко реда, които са грешни, се извежда едно съобщение „Грешка” за всички грешни редове. Ако дадена програма е коректна, да се изведе името ѝ и списък на всички дефинирани променливи и техните стойности след края на програмата.

Вход:

PROGRAM First
MOV(a, 10)
MOV(d, 5)
SUB(a, 7)
MUL(a, d)
ADD(d, a)
END.
ADD(d, 4)
MOV(c, a)
PROGRAM SecondProgram
MOV(a, 3)

Изход:

First
a = 15
d = 20

Грешка

Грешка

ThirdProgram
a = 3
z = 8

```
MOV(z, 5)
ADD(3, z)
END.
PROGRAM ThirdProgram
MOV(a, 3)
MOV(z, 5)
ADD(z, 3)
END.
```

В примера първата грешка е заради командите извън блок PROGRAM – END, а втората – заради опита да се събере число с променлива.