

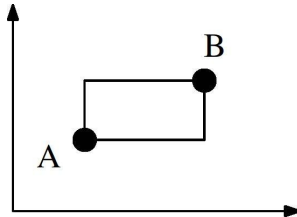
ЗАДАЧА 1: - 10 т.

Направете структура **rectangle_t** (правоъгълник в равнината), за която да се съхраняват следните данни:

- **point_t** A;
- **point_t** B;

Правоъгълникът е описан със срещуположните две точки - A и B.

Всяка структура **point_t** описва точка в равнината - две реални числа **x** и **y**.



Да се създаде свързан списък от правоъгълници. Да се добавят в списъка **COUNT** (предпорцеорна директива) на брой правоъгълници с произволни координати на точките - между -10.0 и 10.0.

- Да се направи функция, която изчислява лицето на всички правоъгълници.

double areaRects(node_t* list);

- Да се направи функция, която изтрива от списъка всички възли, които имат лица в определен интервал - [min - max]:

void filterRects(node_t list, double min, double max);**

- Да се изведе филтрирания списък в подходящ формат.

ЗАДАЧА 2: - 10т.

Да се създаде програма, която преобразува двоичен в текстови файл и обратното.

Вариант 1: Да се прочетат **цели положителни двубайтови числа** от двоичен файл и да се запишат в текстови файл - по едно на ред.

Вариант 2: Да се прочетат **цели положителни числа в интервала от 0 и $2^{16}-1$** от текстови файл записани по едно на ред и да се запишат в двоичен файл - по 2 байта за число.

Имената на файловете и режима на програмата да се подават чрез аргументи от командния ред.

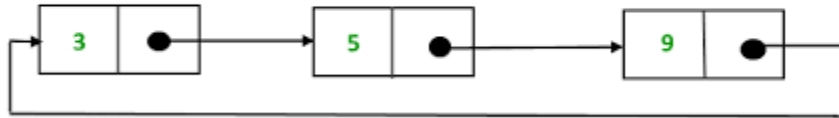
./fileConvert *source_file* *destination_file* *mode*

Mode: **-BT** (от двоичен в текстови Вариант 1), **-TB** (от текстови в двоичен Вариант 2).

Да се направи валидация на входните аргументи и да се направят проверка при отварянето на файловете.

ЗАДАЧА 3: - 10 т.

Циклический связанный список (**cyclic linked list**) — это связанный список, в котором последний элемент указывает (**pointer**) на первый элемент.



Напишите структуру **Node**, за която да се съхраняват следните данни::

- `int data;`
- `struct Node *next;` (Указател към следващия елемент на циклическия свързан списък)

A. (3т.) От стандартния вход се въвеждат числото **N>0** и след това **N цели числа, разделени със SPACE**. Създайте циклический связанный список в динамичната памет, чиито елементи са введените числа. Погрижете се за освобождаването на динамичната памет.

B. (3т.) Напишете функцията `void print(struct Node *c11)`, която приема указател към елемент на циклический связанный список (**c11**) като параметър. Функцията печата всичките елементи на **c11** на стандартния изход, точно по веднъж..

Тествайте функцията, като я извикате в **main**.

C. (4т.) Напишете функцията:

```
struct Node* insertAfter(struct Node *c11, int skipCount, int newElem).
```

Функцията приема следните параметри:

- указател към циклический связанный список (**c11**)
- брой позиции, които ще бъдат пропуснати (**0 <= skipCount <= 1000000**)
- нов елемент, който да бъде добавен.

Функцията започва от **c11**, обхожда следващите **skipCount** на брой елемента и след това добавя новата стойност (**newElem**). Функцията връща указател към новодобавения елемент.

Тествайте функцията, като я извиквате в **main**, с различни тестови аргументи.