

ЗАДАЧА 1: - 10 т.

Да се създаде програма, която генерира масив от реални числа с COUNT на брой елементи в обхвата [MIN, MAX]. COUNT, MIN и MAX са предпроцесорни константи, които да се дефинират по време на компилация (с опция -D на gcc).

/4-те compare функции са такива каквито се подават на стандартната **qsort()** функция/

1. Да се направи функция компаратор, която сортира масива във възходящ ред.
2. Да се направи функция компаратор, която сортира масива в низходящ ред.
3. Да се направи функция компаратор, която сортира масива по сумата на цифрите на цялата част на числото.
/Напр. 45.67 -> 45 -> 4 + 5 -> 9/
4. Да се направи функция компаратор, която сортира масива по абсолютната стойност на числата.

Да се създаде масив от указатели към компаратори, в който да се включат 4-те compare функции. От **stdin** се чете едно цяло число в интервала 1-4 и спрямо него масива да сортира с една от 4те функции с помощта на **qsort()**. След това сортираният масив да се изведе в **stdout**. Ако от **stdin** не се прочетат валидни данни да се изведе съобщение за грешка в **stderr** и да се прекрати действието на програмата.

ЗАДАЧА 2: - 10 т.

Направете програмата от задача 1 с разделно компилиране. Създайте следните файлове:

- **main.c** - с главната функция.
- **comparators.h** - файл с декларациите на компараторите;
- **comparators.c** - файл с дефиницията на компараторите;

Компилирайте програмата като преминете през всички етапи на компилацията за всеки C файл. Прикачете всички междинни файлове. Опишете кратко всеки един от етапите през които се създава програмата.

1. Създайте Makefile с който удобно да се компилира вашата задача.
2. Създайте static library от comparators.c. Компилирайте програмата като включите статичната библиотека.
3. Създайте dynamic library от comparators.c. Компилирайте програмата като включите динамичната библиотека.

/Запишете в текстови файл командите с които правите всичко това/

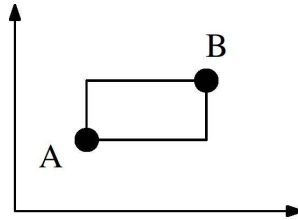
ЗАДАЧА 3: - 10 т.

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**.



Създайте масив от 10 произволни точки с координати по x и y $[-100; 100]$. Да изведе в `stdout` най-голямото разстояние между точките.