

Домашно 1

За всяка от задачите която решите добавете тестове. Документирайте кодът си където сметнете за нужно. Не е задължително да предадете всички задачи заедно.

Задача 1.

Напишете клас `LargeFloat` който представлява реално число с 48 бита. Като отделите 1 бит за знак, 7 бита експонента и 40 бита за мантия. Размера на обектите от вашият клас трябва да е точно 6 байта. След всяка операция поддържайте числото в нормализиран вид.

Трябва да имплементирате поне тези 2 оператора за вашият клас:

```
std::ostream &operator<<(std::ostream &stream, const LargeFloat &value);  
LargeFloat operator+(const LargeFloat &a, const LargeFloat &b);
```

Както и конструктор получаващ един аргумент от тип `double` който инициализира вашият клас на съответната стойност.

```
LargeFloat::LargeFloat(double value);
```

Задача 2.

Напишете функция която приема 3 аргумента:

- Две реални числа (`double`) `A`, и `B` които дефинират интервал
- Указател към функция която приема един аргумент `double` и връща `double`

Вашата функция трябва да интегрира получената функция в подадения интервал и да върне стойността на интеграла. Изберете подходяща точност и големина на стъпката за да интегрирате функцията. Не използвайте стохастичен алгоритъм.

```
double integrateFunction(double A, double B, double (*F)(double));
```

Задача 3.

Разширете задача 2 така че да може да се интегрират функции с произволен брой аргументи. Параметри ще бъде броя аргументи, интервали за всеки от аргументите на функцията и указател към функцията.

За улеснение може да приемете че функциите които интегрирате ще спазват следният прототип:

```
double someFunction(double *arguments, int count);
```

Hint: Направете клас който се инициализира с броя аргументи и указател към функцията и предоставя метод който получава масив от двойки числа които са границите за всеки от аргументите на функцията.