Задача 4.1

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 1$, $b = 0.63068461011847427257$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.2

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 2$, $b = 0.64290468798623146946$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.3

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 3$, $b = 0.65516613322874853863$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.4

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 4$, $b = 0.66746930183853711164$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.5

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 5$, $b = 0.67981455165879275715$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.6

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 6$, $b = 0.69220224244001234377$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.7

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 7$, $b = 0.70463273589956080574$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.8

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 8$, $b = 0.71710639578423441222$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.9

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 9$, $b = 0.72711218527473842103$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".

Задача 4.10

По формуле трапеций вычислите интеграл

$$\int_a^b dx \, \frac{1}{\sqrt{\cosh(x)}}$$

Здесь $a = 0.01 * 10$, $b = 0.73714409702367322749$. Используйте сетку с $N = 2\,000\,000\,000$ узлами. (Это **неправильный** способ считать интегралы!) Измерьте время счета (используйте функцию "`time`"). Ускорьте счет, использовав все 4 ядра CPU. Для этого создайте 4 программных потока. Измерьте время счета и найдите коэффициент ускорения.

Потоки создаются так:

1. Нужен дополнительный инклудовский файл:

   ```
   #include <pthread.h>
   ```

2. Нужны дополнительные переменные:

   ```
   pthread_t th[4];
   void **ret;
   double *xx[4];
   ```

3. Нужно аллоцировать все 4 массива `xx[i]`

   ```
   for(i=0;i<4;i++) xx[i]=(double*)malloc(M*sizeof(double));
   ```

   Эти массивы позволяют родительской программе и детям (потокам) обмениваться данными в обе стороны. В `xx[i]` могут находиться как входные данные, так и результаты счета, исходя из этого надо выбрать размер массивов "`M`".

4. Функцию, вычисляющую четвертинку интегральной суммы, надо определить как

   ```
   static void *f(void *y)
   {
   double *yy =(double*)y;
   // это способ передачи массива xx[i], т.е. yy[k] - это будет xx[i][k]
   ...
   }
   ```

5. Потоки надо создать (запустить) и дождаться их завершения

   ```
   for(i=0;i<4;i++) pthread_create(&th[i],NULL,&f,(void *)xx[i]);
   for(i=0;i<4;i++) pthread_join(th[i],ret);
   ```

Компилировать программу надо с дополнительным флагом "`-pthread`".