**Проблем 1: Инлайн функции и препроцесорни директиви**

**Задача:**

Напишете програма, която изчислява площта на правоъгълник, използвайки инлайн функция. Освен това, дефинирайте препроцесорен макрос, който отпечатва дебъг информация, когато програмата се компилира в дебъг режим.

**Изисквания:**

* Реализирайте инлайн функция calculateArea(int width, int height), която връща площта.
* Използвайте препроцесорна директива (#define DEBUG), за да отпечатвате дебъг информация при компилация в дебъг режим.
* Демонстрирайте използването на инлайн функцията и препроцесорната директива в main().

**Проблем 2: Демонстрация на Name Mangling**

**Задача:**

Напишете програма, която дефинира две функции със същото име, но с различни параметри в C++. Компилирайте програмата и инспектирайте имената на символите, използвайки nm или objdump (Linux) или dumpbin (Windows).

**Изисквания:**

* Дефинирайте две претоварени функции: void print(int) и void print(double).
* Компилирайте програмата и проверете генерираните имена на символите, използвайки nm (objdump) a.out | grep print (или dumpbin за Windows).
* Обяснете разликите в генерираните имена.

**Проблем 3: Сравнение на скоростта на изпълнение на макрос, инлайн и обикновена функция**

**Задача:**

Напишете програма, която сравнява времето за изпълнение на три различни подхода за изчисляване на квадрата на число:

1. Макрос (#define SQUARE(x))
2. Инлайн функция (inline int squareInline(int x))
3. Обикновена (не-инлайн) функция (int squareNotInline(int x))

Програмата трябва да изпълнява всеки метод в цикъл от 10 милиона итерации, да измерва времето за изпълнение, използвайки std::chrono, и да отпечатва резултатите.

**Изисквания:**

* Дефинирайте макрос SQUARE(x), който изчислява квадрата на x.
* Реализирайте инлайн функция squareInline(int x).
* Реализирайте обикновена функция squareNotInline(int x).
* Използвайте std::chrono::high\_resolution\_clock, за да измерите времето за изпълнение на всеки метод.
* Стартирайте всеки метод в цикъл от 10 милиона итерации.
* Сравнете и отпечатайте времената за изпълнение.

**Проблем 4: Логер за извикване на функции с използване на препроцесорни макроси**

**Задача:**

Напишете програма, която дефинира препроцесорен макрос за:

* Автоматично извикване на функция.
* Отпечатване на името на функцията преди изпълнението й.

**Изисквания:**

* Дефинирайте макрос CALL\_FUNC(func, args...), който:
  + Отпечатва "Извикване на функция: func" преди извикването на func(args...).
  + Извиква функцията с подадените аргументи.
* Реализирайте поне две различни функции (напр. add(int, int) и multiply(int, int)).
* Използвайте макроса, за да извикате и двете функции в main().

**Примерен изход:**

Извикване на функция: add

Резултат: 7

Извикване на функция: multiply

Резултат: 12

**Проблем 5: Организация на калкулаторна програма с няколко файла**

**Задача:**

Създайте проста калкулаторна програма, която поддържа събиране, изваждане, умножение и деление, като организирате кода в няколко файла:

1. Файл за операциите на калкулатора
2. Главен (main) файл
3. Хедър файл за декларации

Програмата трябва да включва хедър файл за декларации на функциите и да използва няколко сорс файла за имплементация на операциите на калкулатора.

**Изисквания:**

* Създайте три файла:
  + calculator.h: Хедър файл, съдържащ декларации на функциите (add, subtract, multiply, divide).
  + calculator.cpp: Сорс файл, където се дефинират реалните функции.
  + main.cpp: Главен файл, където потребителят може да избере операция и да въведе числа.

**Реализирани функции:**

* int add(int a, int b) – Връща сумата на a и b.
* int subtract(int a, int b) – Връща разликата между a и b.
* int multiply(int a, int b) – Връща произведението на a и b.
* double divide(int a, int b) – Връща частното на a и b. Ако се опита деление на нула, връща съобщение.

**Използване на хедър файла:**

* Хедър файлът трябва да декларира функциите, а сорс файловете да го включват.

**Детайли по имплементацията:**

* Главната програма трябва да подканя потребителя да избере операция и да въведе две числа.
* В зависимост от избора, съответната функция от calculator.cpp трябва да бъде извикана.
* Отпечатайте резултата от операцията.