Należy stworzyć program z użyciem języka C++ oraz asemblera masm w visual studio 2022 (plik dll dołączany dynamicznie) w którym zajmiemy się obliczaniem otoczki wypukłej z użyciem algorytmu Grahama. W celu integracji biblioteki dll z C++ utworzono i skonfigurowano projekt w visual studio tworząc dwa projekty w solucji (JAproj gdzie umieszczamy kod C++ oraz JAAsm gdzie umieszczony jest plik .asm oraz .def). Podaję główne założenia programu:

Projekt ten dotyczy zagadnienia z obszaru geometrii obliczeniowej, polegającego na wyznaczeniu najmniejszej otoczki wypukłej, która obejmuje wszystkie punkty zadanego zbioru w przestrzeni dwuwymiarowej. Implementacja będzie opierać się na algorytmie Grahama, jednym z powszechnie stosowanych algorytmów do wyznaczania otoczki wypukłej. Program przyjmie dane wejściowe w postaci pliku tekstowego (.txt), zawierającego współrzędne punktów w układzie kartezjańskim (𝑥,𝑦). W części zaimplementowanej w języku asemblera obliczane będą kąty, jakie tworzą poszczególne punkty z prostą przechodzącą przez punkt początkowy. Następnie punkty zostaną posortowane według tych kątów. Dodatkowo w ramach obliczeń zostanie uwzględnione wyznaczenie iloczynu wektorowego, niezbędnego do poprawnego działania algorytmu. Wyjściem programu będzie lista współrzędnych punktów tworzących otoczkę wypukłą, zapisana w pliku .txt, z możliwością graficznej wizualizacji wyników. Złożoność czasowa zastosowanego algorytmu wynosi 𝑂(𝑛 log𝑛) gdzie 𝑛 to liczba punktów w zbiorze wejściowym. Złożoność ta wynika głównie z procesu sortowania punktów, który dominuje czas wykonania algorytmu. Projekt ma na celu opracowanie efektywnego rozwiązania problemu otoczki wypukłej dla punktów w przestrzeni dwuwymiarowej, z zastosowaniem algorytmu Grahama. Kluczowym elementem będzie optymalizacja części obliczeniowej przy wykorzystaniu asemblera, co pozwoli na zwiększenie wydajności programu poprzez bezpośrednią manipulację na poziomie sprzętowym.

Projekt jednak ma wymagania co do używanych mechanizmów języka, czy technologii:  
  
**Załozenia projektu:**

1. Projekt jest pracą indywidualną
2. Projekt z programowania w Językach Asemblerowych dotyczy architektury procesorów Intel X64
3. Projekt realizowany jest w środowisku Visual Studio
4. Projekt składa się z dwóch powiązanych projektów:
   1. Aplikacja Interfejsu Użytkownika w dowolnym języku wysokiego poziomu (C/C++/C#)
   2. Dwie tożsame biblioteki DLL - jedna w języku wysokiego poziomu (C/C++) druga w ASM x64
5. Projekt powinien wykorzystywać wielowątkowość
6. Projekt powinien umożliwiać pomiar czasu wykonania
7. W projekcie należy (sensownie!) wykorzystać instrukcje wektorowe.

**Wymagania:**

1. Każdy plik źródłowy C/C++/C#, ASM ma rozpoczynać się od komentarza zawierającego:
   * Imię Nazwisko autora, data wykonania projektu Rok/Semestr
   * Temat projektu
   * Krótki opis algorytmu
   * Aktualną wersję programu i historię zmian funkcjonalności w kolejnych wersjach
2. Wszystkie procedury muszą być odpowiednio skomentowane:
   * Nazwa i krótki opis procedury
   * Parametry wejściowe z opisem ich przeznaczenia i zakresem wartości
   * Parametry wyjściowe z opisem i zakresem wartości
   * Rejestr i flagi które ulegają zmianie po wykonaniu procedury
3. Każda zmienna i stała również mus zostać odpowiednio skomentowana
4. Linie kodu ASM powinny dodatkowo zostać zaopatrzone komentarzami informującymi o ich przeznaczeniu

**Zasada działania programu:**

1. Program powinien umożliwiać dynamiczną zmianę biblioteli DLL w trakcie działania aplikacji
2. Aplikacja powinna być zabezpieczona przed próbą użycia nieprawidłowych danych wejściowych
3. Aplikacja powinna działać w wątkach. Ilość wątków powinna być ustalana w interfejsie graicznym. Zmiana ilości wątków ma następować bez konieczności przeładowania aplikacji

**Wyniki działania programu:**

1. Program należy przetwstować dla różnych rodzajów danych wejściowych (np. mały, średni i duży obrazek)
2. Należy zmieżyć czas działania programu dla każdego rodzaju danych wejśiowych, dla obu bibliotek (np. uśredniony czas 5 wywołań)
3. Dodatkowo należy uwzględni czas wykonania programu dla kodu bez optymalizacji oraz z optymalizacją (czasową i/lub pamięciową).
4. Zebrane wyniki należy przedtsawić za pomocą wykresów (np. w programie Excel) i przedstawić w raporcie końcowym wraz z wnioskami.

Chciałbym na podstawie powyższych założeń, abyś zaproponował strukturę plików, funkcji i innych istotnych kwestii organizacyjnych na początku projektu abym mógł dobrze przygotować strukturę projektu a potem zająć się tylko imlementacją i testowaniem programu. Jak na razie chcę użyć aplikacji konsolowej do wyświetlania i debugowania dla mojego algorytmu a aplikację graficzną stworzę później (napisz czy to dobry pomysł i czy dam radę to przenieść). Daj mi istotne wskazówki jak z tym wszystkim zacząć (chodzi głównie o strukturę projektu, sam algorytm i naturę problemu przestudiowałem osobno).