Podstawy OpenCL

1. Napisz program z użyciem OpenCL, który wykonuje dodawanie dwóch wektorów. Suma wektorów zapisywana jest w trzecim wektorze, którego wartość wyświetlana jest na ekranie. Program proszę przetestować dla różnych wielkości wektorów – uwzględniając fakt, że pojedynczy wątek będzie musiał wykonać operację dodawania więcej niż jednej wartości w przypadku dużych rozmiarów wektorów. Przyjąć globalną liczbę wątków równą 32, natomiast rozmiar grupy wątków równy 8.

2. Napisz program z użyciem OpenCL, który wyświetla informacje o dostępnych platformach OpenCL korzystając z funkcji `clGetPlatformInfo` oraz następujących stałych:

   - `CL_PLATFORM_VERSION`
   - `CL_PLATFORM_NAME`
   - `CL_PLATFORM_VENDOR`
   - `CL_PLATFORM_EXTENSIONS`

   Następnie korzystając z funkcji `clGetDeviceInfo` należy wyświetlić informacje o wybranym urządzeniu używając kolejno jako parametrów stałych:

   - `CL_DEVICE_NAME`,
   - `CL_DEVICE_MAX_COMPUTE_UNITS`,
   - `CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS`,
   - `CL_DEVICE_MAX_WORK_ITEM_SIZES`,
   - `CL_DEVICE_MAX_WORK_GROUP_SIZE`,
   - `CL_DEVICE_LOCAL_MEM_SIZE`.

3. Napisać program generujący wizualizację zbioru Julii na podstawie poniższej funkcji, która dla zadanych współrzędnych sprawdza, czy punkt ten należy do zbioru. Punkty należące do zbioru należy pokolorować na biało, natomiast pozostałe na czarno.

   Wygenerowany obraz zapisać w formacie PNG za pomocą biblioteki `lodepng`.

```
// size - szer / wys obrazu
char in_julia(int x, int y, int size) {
    const float scale = 1.5f;
    const float jx = scale * (float)(size / 2 - x)/(size / 2);
    const float jy = scale * (float)(size / 2 - y)/(size / 2);
```

```
    float c[] = { -0.8f, 0.156f };
    float a[] = { jx, jy };
    short i = 0;
    for (i = 0; i < 200; ++i) {
        float a_next[] = { a[0]*a[0] - a[1]*a[1] + c[0],
                           a[0]*a[1] + a[1]*a[0] + c[1] };
        a[0] = a_next[0];
        a[1] = a_next[1];
        if (a[0] * a[0] + a[1] * a[1] > 1000) {
            return 0;
        }
    }
    return 1;
}
```