

## **STRESZCZENIE**

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

**Słowa kluczowe:** słowa kluczowe

**Dziedziny nauki i techniki zgodne z wymogami OECD:** 1.2 Nauki o komputerze i informatyce

## **ABSTRACT**

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

**Keywords:** keywords

**OECD field of science and technology:** 1.2 Computer and information sciences

## SPIS TREŚCI

Wykaz ważniejszych oznaczeń i skrótów .....	6
1. Wprowadzenie .....	7
1.1. Cel pracy.....	7
2. Wstęp do Api OpenCL .....	8
2.1. Linkowanie biblioteki OpenCL na platformie Android .....	8
2.2. Typowy przebieg aplikacji OpenCL.....	9
2.3. Możliwości i ograniczenia sprzętowe .....	11
2.4. Pomiary czasu wykonywania kerneli .....	14
3. Specyfika testowanych Urządzeń .....	15
3.1. Pierwsza sekcja .....	15
3.1.1. Podsekcja.....	15
3.1.2. I kolejna podsekcja.....	16
3.2. Druga sekcja .....	17
3.2.1. Podsekcja sekcji drugiej.....	17
3.2.2. Kod.....	18
4. Testy Performancowe OpenCL na Mobilkach .....	20
4.1. Pomiar Mocy Obliczeniowej.....	20
4.2. Przepływ pamięci.....	21
4.3. Czas Oczekiwania na wykonanie kernela .....	22
5. Aplikacje wykorzystujące OpenCL Przypadki Użycia .....	24
5.1. Pierwsza sekcja .....	24
5.1.1. Podsekcja.....	24
5.1.2. I kolejna podsekcja.....	25
5.2. Druga sekcja .....	26
5.2.1. Podsekcja sekcji drugiej.....	26
5.2.2. Kod.....	27
6. Analiza Wyników, Wnioski.....	29
6.1. Pierwsza sekcja .....	29
6.1.1. Podsekcja.....	29
6.1.2. I kolejna podsekcja.....	30
6.2. Druga sekcja .....	31
6.2.1. Podsekcja sekcji drugiej.....	31
6.2.2. Kod.....	32
Wykaz literatury .....	34
Wykaz rysunków.....	34
Wykaz tabel .....	35
Dodatek A. Przykładowy dodatek.....	36

## **WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW**

**PG** – Politechnika Gdańska

**WETI** – Wydział Elektroniki, Telekomunikacji i Informatyki

## **1. WPROWADZENIE**

### **1.1. *Cel pracy***

## 2. WSTĘP DO API OPENCL

OpenCL czyli Open Compute Language to standard tworzony obecnie przez grupę Khronos, służący do pisania programów, które mogą zostać wykonane na różnych platformach takich jak CPU, GPU czy FPGA. Specyfikacja OpenCL definiuje interfejs w języku c++, który umożliwia zaprogramowanie aplikacji by ta wykonała konkretny kod na wybranym urządzeniu. Standard OpenCL jest głównie wykorzystywany do równoległych obliczeń takie jak wektorowe operacje matematyczne czy przetwarzanie obrazów. Za implementację sterownika który wystawia api zgodne z określoną wersją specyfikacji odpowiedzialny jest producent urządzenia. Dzięki temu, że standard jest otwarty a jego implementację posiada większość producentów możemy stworzyć kod który możemy uruchomić niezależnie od architektury czy producenta posiadanego procesora głównego czy graficznego. Jest to duża zaleta w porównaniu na przykład do CUDA, która to jest interfejsem implementowanym jedynie przez NVidie. Standard OpenCL jest rozwijany i modyfikowany, przez co api zdefiniowane jest w kilku wersjach. Najnowsza wersja specyfikacji to wersja 3.0. Wszystkie wersje są kompatybilne wstecznie. Dodatkowo zdefiniowane są rozszerzenia api takie jak `cl_khr_gl_sharing` czy definiujący api do sharingu obiektów między OpenCL a OpenGL, takie dodatkowe api jest też specyfikowane przez grupę Khronos w ramach określonej wersji OpenCL, jednak nie jest obowiązkowe. Istnieją także rozszerzenia api wyspecyfikowane przez konkretnego producenta np `cl_intel_mem_force_host_memory`, które jest dostępna na urządzeniach intel'a, lub `cl_qcom_android_native_buffer_host_ptr` dostępny na procesorach Qualcom'a z systemem Android. Takie dodatkowe api uzupełnia podstawę, umożliwiając lepsze dopasowanie specyfikacji do konkretnego sprzętu. OpenCL na platformie Android dostępny jest jedynie z poziomu natywnej biblioteki w języku c++.

### 2.1. *Linkowanie biblioteki OpenCL na platformie Android*

Aplikacje zwykle nie linkują się bezpośrednio ze sterownikiem posiadającym kompletną implementację api. Do tego wykorzystywana jest dodatkowa biblioteka która wyszukuje implementację sterownika dla wszystkich platform na urządzeniu. Dzięki wykorzystaniu takiej ładującej biblioteki Aplikacja może używać każdej dostępnej platformy wspierającej to Api, oraz nie jest sztywno połączona z jednym sterownikiem w określonej wersji. Niestety nie istnieje binarna wersja takiej biblioteki dla systemu Android. Istnieje możliwość połączenia natywnej biblioteki ze sterownikiem znajdującym się w urządzeniu, jednak wadą takiego rozwiązania jest konieczność pobrania z urządzenia pliku binarnego z implementacją OpenCL oraz wszystkich zależnych od niej sterowników. Takie rozwiązanie powoduje, że skompilowana aplikacja będzie działać jedynie na urządzeniu, z którego zostały pobrane biblioteki. Innym rozwiązaniem jest pobranie źródeł z kodem biblioteki ładującej, zbudowanie biblioteki i połączenie jej z aplikacją. Sterownik który będzie łączył program z implementacją OpenCL ma zapisane domyślne ścieżki w których mogą znajdować się biblioteka OpenCL. Istnieje również możliwość zdefiniowania zmiennej środowiskowej w której wskażemy lokalizację z której chcemy by sterownik został załadowany. Wadą takiego rozwiązania

jest konieczność kompilacji takiej biblioteki, natomiast dzięki temu możemy zbudować aplikację działającą na wielu urządzeniach. <https://github.com/krrishnarraj/libopencl-stub>

## 2.2. Typowy przebieg aplikacji OpenCL

Poniżej widać jak wygląda przebieg prostego programu wykorzystującego Api OpenCL do inkrementowania każdego elementu bufora pamięci.

```
1  cl_int err = 0;
2  std::unique_ptr<cl_platform_id> platforms;
3  cl_device_id device_id = 0;
4  cl_uint platformsCount = 0;
5  cl_context context = NULL;
6  cl_command_queue queue = NULL;
7  cl_program program = NULL;
8  cl_kernel kernel = NULL;
9  cl_mem buffer = NULL;
10 const size_t bufferSize = sizeof(int) * 1024;
11
12 cl_uint dimension = 1;
13 size_t offset[3] = {0, 0, 0};
14 size_t gws[3] = {bufferSize, 1, 1};
15 size_t lws[3] = {4, 1, 1};
16
17 err = clGetPlatformIDs(0, NULL, &platformsCount);
18 platforms = std::make_unique<cl_platform_id>(platformsCount);
19 err = clGetPlatformIDs(platformsCount, platforms.get(), NULL);
20 cl_device_type deviceType = CL_DEVICE_TYPE_GPU;
21 err = clGetDeviceIDs(platforms.get()[0], deviceType, 1, &device_id,
    NULL);
22 context = clCreateContext(0, 1, &device_id, NULL, NULL, &err);
23 queue = clCreateCommandQueue(context, device_id, 0, &err);
24
25 program = clCreateProgramWithSource(context, 1, &kernelStrings, 0,
    &err);
26 err = clBuildProgram(program, 1, &device_id, nullptr, nullptr,
    nullptr);
27 kernel = clCreateKernel(program, "increment", &err);
28 cl_mem_flags flags = CL_MEM_READ_WRITE;
29 buffer = clCreateBuffer(context, flags, bufferSize, nullptr, &err);
30 void *ptr = clEnqueueMapBuffer(queue, buffer, CL_TRUE, CL_MAP_READ,
    0, bufferSize, 0, nullptr, nullptr, &err);
31 memset(ptr, 13, bufferSize);
32 err = clEnqueueUnmapMemObject(queue, buffer, ptr, 0, nullptr,
    nullptr);
33 err = clSetKernelArg(kernel, 0, sizeof(cl_mem), &buffer);
34 err = clEnqueueNDRangeKernel(queue, kernel, dimension, offset, gws,
    lws, 0, 0, nullptr);
35 err = clFinish(queue);
```

Pierwszym krokiem jest zwołanie **clGetPlatformIDs**. Podając drugi argument czyli `cl_platform_id` × jako null, do trzeciego argumentu jakim jest `cl_uint` zostanie wpisana liczba dostępnych na urządzeniu platform wspierających OpenCL. Drugie wywołanie `clGetPlatformIDs` wpisze informacje o podanej liczbie platform i zapisze je w tablicy podanej w drugim argumencie. W tej pracy w testowanych aplikacjach na sytem android wykorzystywany będzie `CL_DEVICE_TYPE_GPU`, który jako jedyny jest dostępny na testowanych przezemnie urządzeniach. Posiadając obiekt device,wołając **clCreateContext** możemy stworzyć context w ramach którego możliwe jest zarządza-

nie pozniej stworzonymi obiektami na określonych przy tworzeniu contextu urządzeniach. Dalej w przykładowym kodzie stworzona jest kolejka **clCreateCommandQueue** kolejka powstaje w ramach contextu na konkretny device. Później na tym obiekcie kolejgowane będą zadania takie jak transfery pamieci czy wykonywane funkcje. W zależności czy kolejka zostanie stworzona z flagą **CL\_QUEUE\_OUT\_OF\_ORDER\_EXEC\_MODE\_ENABLE** lub bez niej, zadania te będą mogły być wykonywane równolegle, lub jedno po drugim w kolejnosci dodania do kolejki. Kolejnym krokiem w powyższym kodzie jest stworzenie obiektu programu używając **clCreateProgramWithSource** lub **clCreateProgramWithBinary** pierwsza stworzy program zawierający nieskompilowany kod w języku OpenCL C, natomiast druga stworzy obiekt z binarnej wersji, wcześniej skompilowanej. Do stworzenia programu ze źródeł przekazywany jest ciąg znaków zawierający kernele, czyli funkcje które mogą zostać wykonane na urządzeniu. Do wykonania inkrementacji kazdego elementu bufora w przykładownie użyty zostanie następujący kernel

```
1 __kernel void increment(__global int* in){
2     int i= get_global_id(0);
3     in[i]++;
4 }
```

W funkcji tej zostaje pobrany unikalny numer aktualnie wykonywanego kernela w ramach globalnej work grupy, następnie element bufora pod tym indeksem jest inkrementowany. Po stworzeniu programu zawierającego kernele, należy zawołać **clBuildProgram** by kod kerneli w języku OpenCL C został skompilowany dla wskazanego urządzenia. W przypadku stworzenia programu ze źródeł w formie binarnej, tego kroku się nie wykonuje. Następnym wykonanym krokiem jest stworzeniem obiektu bufora poprzez **clCreateBuffer**. Stworzony zostaje obiekt reprezentujący obszar pamięci o podanym rozmiarze, który może być wykorzystany przy wykonywaniu kernela. W podanym kodzie powstanie bufor o rozmiarze 4096 bajtów czyli 1024 elementów typu int. Następnie zostaje wykonane **clEnqueueMapBuffer** funkcja ta zmapuje konkretny bufor na obszar pamięci dostępny z poziomu aplikacji. W tym przypadku w pamięci pod zwróconym wskaźnikiem ustawiamy w każdym bajcie wartość 13. By przesłać pamięć z powrotem do obiektu bufora dostępnego z poziomu urządzenia na którym będzie wykonywany kernel wołamy **clEnqueueUnmapMemObject**. Tak przygotowany bufor z danymi możemy ustawić jako argument kernela wołając **clSetKernelArg** podając w argumentach kernel, który w którym chcemy ustawić argument, index argumentu, jego typ oraz wskaźnik na obiekt który będzie argumentem funkcji. Funkcja która uruchomi wykonanie kernela na wskazanym wcześniej urządzeniu jest **clEnqueueNDRangeKernel**, która umieści w kolejce do wykonywania wskazany kernel. W tym momencie podane jest także w ilu wymiarach odbędzie się wykonywanie, podana jest wielkość lokalnej i globalnej work grupy. Na samym końcu zawołane zostaje **clFinish**, jest to funkcja blokująca po wykonaniu której mamy pewność, że wszystkie zakolejkowane na konkretnej kolejce operacje zostały wykonane. W wyniku działania takiego kodu w buforze znajdować się będzie 1024 elementy o wartości 13131314.



### 2.3. Możliwości i ograniczenia sprzętowe

Każde urządzenie posiada ograniczenia związane ze specyfiką implementacji sterownika oraz barkiem zasobów sprzętowych. By dowiedzieć się jakie są maksymalne dostępne wartości np dotyczące rozmiaru pamięci, ilości poszczególnych obiektów w kernelu czy maksymalnej liczbie dostępnych work itemów w ramach lokalnej work grupy możemy odpytać sterownik wołając `clGetDeviceInfo` podając konkretny parametr. Dzięki temu wykonywana aplikacja może dostosować się do ograniczeń sprzętowych. Oto wynik działania aplikacji "clinfo", która wypisuje wszystkie dostępne informacje o urządzeniu. W tym wypadku jest to telefon Xiaomi Mi a2 lite z procesorem graficznym Adreno 506

```

1 Number of platforms 1
2   Platform Name      QUALCOMM Snapdragon(
   TM)
3   Platform Vendor    QUALCOMM
4   Platform Version   OpenCL 2.0 QUALCOMM
   build: commit #2df12b3 changeid #I07da2d9908 Date: 10/04/18 Thu
   Local Branch: Remote Branch:
5   Platform Profile   FULL_PROFILE
6   Platform Extensions
7
8   Platform Name      QUALCOMM Snapdragon(
   TM)
9 Number of devices    1
10  Device Name         QUALCOMM Adreno(TM)
11  Device Vendor       QUALCOMM
12  Device Vendor ID    0xbf4d3c4b
13  Device Version      OpenCL 2.0 Adreno(TM)
   506
14  Driver Version      OpenCL 2.0 QUALCOMM
   build: commit #2df12b3 changeid #I07da2d9908 Date: 10/04/18 Thu
   Local Branch: Remote Branch: Compiler E031.36.02.00
15  Device OpenCL C Version OpenCL C 2.0 Adreno(
   TM) 506
16  Device Type         GPU
17  Device Profile      FULL_PROFILE
18  Device Available    Yes
19  Compiler Available   Yes
20  Linker Available    Yes
21  Max compute units   1
22  Max clock frequency 1MHz
23  Device Partition    (core)
24    Max number of sub-devices 1
25    Supported partition types  None
26    Supported affinity domains (n/a)
27  Max work item dimensions 3
28  Max work item sizes  1024x1024x1024
29  Max work group size  1024
30  Preferred work group size multiple (kernel) 1024
31  Preferred / native vector sizes
32    char               1 / 1
33    short              1 / 1
34    int                1 / 1
35    long               1 / 0
36    half               1 / 1
   cl_khr_fp16)
37    float              1 / 1

```

```

38     double                                0 / 0      (
        n/a)
39 Half-precision Floating-point support      (cl_khr_fp16)
40     Denormals                             No
41     Infinity and NaNs                     Yes
42     Round to nearest                       Yes
43     Round to zero                          No
44     Round to infinity                      Yes
45     IEEE754-2008 fused multiply-add        No
46     Support is emulated in software         No
47 Single-precision Floating-point support    (core)
48     Denormals                             No
49     Infinity and NaNs                     Yes
50     Round to nearest                       Yes
51     Round to zero                          No
52     Round to infinity                      Yes
53     IEEE754-2008 fused multiply-add        No
54     Support is emulated in software         No
55     Correctly-rounded divide and sqrt operations No
56 Double-precision Floating-point support    (n/a)
57 Address bits                              64, Little-Endian
58 Global memory size                        1875912704 (1.747GiB)
59 Error Correction support                  No
60 Max memory allocation                     468978176 (447.3MiB)
61 Unified memory for Host and Device        Yes
62 Shared Virtual Memory (SVM) capabilities  (core)
63     Coarse-grained buffer sharing          Yes
64     Fine-grained buffer sharing            No
65     Fine-grained system sharing           No
66     Atomics                              No
67 Minimum alignment for any data type       128 bytes
68 Alignment of base address                 1024 bits (128 bytes)
69 Page size (QCOM)                          4096 bytes
70 External memory padding (QCOM)            0 bytes
71 Preferred alignment for atomics
72     SVM                                    128 bytes
73     Global                                0 bytes
74     Local                                 0 bytes
75 Max size for global variable               65536 (64KiB)
76 Preferred total size of global vars        1048576 (1024KiB)
77 Global Memory cache type                  Read/Write
78 Global Memory cache size                  16384 (16KiB)
79 Global Memory cache line size             64 bytes
80 Image support                             Yes
81     Max number of samplers per kernel      16
82     Max size for 1D images from buffer     134217728 pixels
83     Max 1D or 2D image array size          2048 images
84     Base address alignment for 2D image buffers 64 bytes
85     Pitch alignment for 2D image buffers    64 pixels
86     Max 2D image size                      16384x16384 pixels
87     Max 3D image size                      16384x16384x2048
        pixels
88     Max number of read image args          128
89     Max number of write image args         64
90     Max number of read/write image args     64
91     Max number of pipe args                16
92     Max active pipe reservations           4096
93     Max pipe packet size                   1024
94     Local memory type                     Local
95     Local memory size                     32768 (32KiB)
96     Max number of constant args            8
97     Max constant buffer size               65536 (64KiB)

```

```

98 Max size of kernel argument 1024
99 Queue properties (on host)
100 Out-of-order execution Yes
101 Profiling Yes
102 Queue properties (on device)
103 Out-of-order execution Yes
104 Profiling Yes
105 Preferred size 655376 (640KiB)
106 Max size 655376 (640KiB)
107 Max queues on device 1
108 Max events on device 1024
109 Prefer user sync for interop No
110 Profiling timer resolution 1000ns
111 Execution capabilities
112 Run OpenCL kernels Yes
113 Run native kernels No
114 printf() buffer size 1048576 (1024KiB)
115 Built-in kernels (n/a)
116 Device Extensions
    cl_khr_3d_image_writes cl_img_egl_image
    cl_khr_byte_addressable_store cl_khr_depth_images
    cl_khr_egl_event cl_khr_egl_image cl_khr_fp16 cl_khr_gl_sharing
    cl_khr_global_int32_base_atomics
    cl_khr_global_int32_extended_atomics
    cl_khr_local_int32_base_atomics
    cl_khr_local_int32_extended_atomics cl_khr_image2d_from_buffer
    cl_khr_mipmap_image cl_khr_srgb_image_writes cl_khr_subgroups
    cl_qcom_create_buffer_from_image cl_qcom_ext_host_ptr
    cl_qcom_ion_host_ptr cl_qcom_perf_hint cl_qcom_read_image_2x2
    cl_qcom_android_native_buffer_host_ptr cl_qcom_protected_context
    cl_qcom_priority_hint cl_qcom_compressed_yuv_image_read
    cl_qcom_compressed_image
117
118 NULL platform behavior
119 clGetPlatformInfo(NULL, CL_PLATFORM_NAME, ...) No platform
120 clGetDeviceIDs(NULL, CL_DEVICE_TYPE_ALL, ...) No platform
121 clCreateContext(NULL, ...) [default] No platform
122 clCreateContext(NULL, ...) [other] Success [P0]
123 clCreateContextFromType(NULL, CL_DEVICE_TYPE_DEFAULT) Success (1)
124 Platform Name QUALCOMM Snapdragon(TM)
125 Device Name QUALCOMM Adreno(TM)
126 clCreateContextFromType(NULL, CL_DEVICE_TYPE_CPU) No devices found
    in platform
127 clCreateContextFromType(NULL, CL_DEVICE_TYPE_GPU) Success (1)
128 Platform Name QUALCOMM Snapdragon(TM)
129 Device Name QUALCOMM Adreno(TM)
130 clCreateContextFromType(NULL, CL_DEVICE_TYPE_ACCELERATOR) No devices
    found in platform
131 clCreateContextFromType(NULL, CL_DEVICE_TYPE_CUSTOM) Invalid device
    type for platform
132 clCreateContextFromType(NULL, CL_DEVICE_TYPE_ALL) Success (1)
133 Platform Name QUALCOMM Snapdragon(TM)
134 Device Name QUALCOMM Adreno(TM)

```

## **2.4. Pomiary czasu wykonywania kerneli**

Standard OpenCL zapewnia mechanizm do odczytywania stempli czasu z poszczególnych etapów wykonywania kernela. Służy do tego obiekt typu `cl_event` stworzony przez funkcję `clCreateUserEvent`. Obiekt taki może zostać przekazany jako argument funkcji, która zostanie wykonana na urządzeniu np `clEnqueueNDRangeKernel` czy `clWnqueueWriteBuffer`. Po wykonaniu kernela z obiektu eventa można odczytać stemple czasu z jego wykonania. By wydobyć wartości należy zwołać `clGetEventProfilingInfo` przekazując jako argument obiekt eventu oraz jeden z czterech paramterów

- `CL_PROFILING_COMMAND_QUEUED` wartość opisuje czas urządzenia w którym komenda została dodana do kolejki.
- `CL_PROFILING_COMMAND_SUBMIT` wartość opisuje czas urządzenia w którym komenda została wysłana do urządzenia na którym zostanie wykonana.
- `CL_PROFILING_COMMAND_START` wartość opisuje czas urządzenia w którym rozpoczęte zostało wykonywanie komenty na urządzeniu.
- `CL_PROFILING_COMMAND_END` wartość opisuje czas urządzenia w którym wykonywanie komendy zostaje zakończone.

### 3. SPECYFIKA TESTOWANYCH URZĄDZEŃ

#### 3.1. Pierwsza sekcja

##### 3.1.1. Podsekcja

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Rys. 3.1. Logo WETI

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

**Tabela 3.1.** Przykład krótkiej tabeli

Nagłówek 1	Nagłówek 2	Nagłówek 3
1	lubię plaki	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
2	a ja nie	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Rys. 3.2. Logo PG

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

### 3.1.2. I kolejna podsekcja

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

### 3.2. Druga sekcja

#### 3.2.1. Podsekcja sekcji drugiej

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

**Tabela 3.2.** Przykład długiej tabeli

Nagłówek 1	Nagłówek 2
1	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
2	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
3	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nagłówek 1	Nagłówek 2
4	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
5	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
6	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

### 3.2.2. Kod

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << [](ostream& out) -> decltype(auto) { return out << "Hello
        world!\n"; } << endl;

```



```
7     return 0;  
8 }
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## 4. TESTY PERFORMANCOWE OPENCL NA MOBILKACH

### 4.1. Pomiar Mocy Obliczeniowej

Pomiary mocy obliczeniowej zostaną przeprowadzone za pomocą następującego testu. Wykonany zostanie jeden z poniższych kerneli. Wykożystane zostaną wektorowe typy danych. Dla każdego z tych kerneli liczba wykonanych operacji zmienno przecinkowych powinna być taka sama i wynosić 4096 dla pojedynczego work itemu. W przykładowo kernelu flops\_float1 operacja mad zostanie wykonana 2048 razy funkcja ta składa się z pojedynczego mnożenia i dodawania.

```
1 #define MAD_4(x, y)      x = mad(y, x, y);    y = mad(x, y, x);    x = mad
   (y, x, y);    y = mad(x, y, x);
2 #define MAD_16(x, y)    MAD_4(x, y);          MAD_4(x, y);          MAD_4(x
   , y);          MAD_4(x, y);
3 #define MAD_64(x, y)    MAD_16(x, y);          MAD_16(x, y);          MAD_16(
   x, y);          MAD_16(x, y);
4
5
6 __kernel void flops_float1(__global float *ptr, float _A)
7 {
8     float x = _A;
9     float y = (float)get_local_id(0);
10    int gid = get_global_id(0);
11
12    for(int i=0; i<128; i++)
13    {
14        MAD_16(x, y);
15    }
16
17    ptr[gid] = y;
18 }
19
20
21 __kernel void flops_float2(__global float *ptr, float _A)
22 {
23     float2 x = (float2)(_A, (_A+1));
24     float2 y = (float2)get_local_id(0);
25     int gid = get_global_id(0);
26
27     for(int i=0; i<64; i++)
28     {
29         MAD_16(x, y);
30     }
31
32     ptr[gid] = (y.S0) + (y.S1);
33 }
34
35 __kernel void flops_float4(__global float *ptr, float _A)
36 {
37     float4 x = (float4)(_A, (_A+1), (_A+2), (_A+3));
38     float4 y = (float4)get_local_id(0);
39
40     for(int i=0; i<32; i++)
41     {
42         MAD_16(x, y);
43     }
44
45     ptr[get_global_id(0)] = (y.S0) + (y.S1) + (y.S2) + (y.S3);
46 }
```

```

47
48
49 __kernel void flops_float8(__global float *ptr, float _A)
50 {
51     float8 x = (float8)(_A, (_A+1), (_A+2), (_A+3), (_A+4), (_A+5), (_A
52         +6), (_A+7));
53     float8 y = (float8)get_local_id(0);
54
55     for(int i=0; i<16; i++)
56     {
57         MAD_16(x, y);
58     }
59
60     ptr[get_global_id(0)] = (y.S0) + (y.S1) + (y.S2) + (y.S3) + (y.S4)
61         + (y.S5) + (y.S6) + (y.S7);
62 }
63
64 __kernel void flops_float16(__global float *ptr, float _A)
65 {
66     float16 x = (float16)(_A, (_A+1), (_A+2), (_A+3), (_A+4), (_A+5), (_
67         _A+6), (_A+7),
68         (_A+8), (_A+9), (_A+10), (_A+11), (_A+12), (_A+13),
69         (_A+14), (_A+15));
70     float16 y = (float16)get_local_id(0);
71
72     for(int i=0; i<8; i++)
73     {
74         MAD_16(x, y);
75     }
76
77     float2 t = (y.S01) + (y.S23) + (y.S45) + (y.S67) + (y.S89) + (y.SAB
78         ) + (y.SCD) + (y.SEF);
79     ptr[get_global_id(0)] = t.S0 + t.S1;
80 }

```

Argument kernela `_A` to przykładowa, zmiennie przecinkowa wartość początkowa. W przeprowadzonych testach rozmiar lokalnej work grupy to maksymalny możliwy rozmiar dla kernela. Natomiast rozmiar globalnej work grupy to największy możliwy rozmiar lokalnej grupy przemnożony przez liczbę dostępnych jednostek wykonawczych razy 2048.

Uzyskany wynik przedstawiony zostaje w jednostce FLOPS jest to jednostka określająca liczbę wykonanych operacji zmiennie przecinkowych na sekundę. W tym teście wartość w FLOPS otrzymamy przez pomnożenie liczby globalnych work itemów przez liczbę wykonywanych zmiennie przecinkowych operacji w każdym z nich, a następnie podzielenie uzyskanej wartości przez czas w jakim te się wykonywały. Do zmierzenia czasu wykorzystano obiekt typu `cl_event`. Po wykonaniu kernela zostały odczytane wartości `CL_PROFILING_COMMAND_START` i `CL_PROFILING_COMMAND_END`. Różnica tych wartości to czas wykonywania funkcji na urządzeniu.

Analogiczne kernele zostaną wykorzystane do przetestowania innych typów danych takich jak `integer half` i `double`, jeśli te są wspierane przez testowane urządzenie.

## 4.2. Przepływ pamięci

Zbadane zostało jak szybko dane zostają kopiowane pomiędzy różnymi obszarami pamięci. Do przetestowania został użyty prosty kernel.

```

1 #ifdef FLOAT1
2 typedef float Type;
3 #endif
4 #ifdef FLOAT2
5 typedef float2 Type;
6 #endif
7
8 #ifdef FLOAT4
9 typedef float4 Type;
10 #endif
11
12 #ifdef FLOAT8
13 typedef float8 Type;
14 #endif
15
16 #ifdef FLOAT16
17 typedef float16 Type;
18 #endif
19
20 __kernel void readFloatType(__global Type *dst, __global Type *src){
21     uint gid = get_global_id(0);
22     dst[gid] = src[gid];
23 }

```

W wykonywanym kernelu dla pojedynczego work itemu kopiowana jest jedna komórka pamięci z bufora src do dst. Typ pojedynczego elementu bufora jest definiowany na etapie kompilacji. W tym przykładzie może być to jedna z wektorowych wersji typu float.

W testach stworzone zostają dwa bufor. Pierwszy posiada inicjalne dane, a drugi jest pusty. Po wykonaniu kernela w drugim buforze znajdują się dane z pierwszego. Zebrane informacje o czasie z obiektu typu `cl_event` pozwalają nam obliczyć z jaką prędkością w bajtach na sekundę dochodzi do transferu pamięci. Analogicznie kernel używające buforów pamięci o typie danych integer half czy double zostaną także przetestowane.

#### 4.3. Czas Oczekiwania na wykonanie kernela

W celu sprawdzenia czasu oczekiwania na rozpoczęcie wykonywania kernela, wykonany został następujący test. Wykonany jest dowolny kernel w testowanym scenariuszu następujący.

```

1 __kernel void increment(__global int* in){
2     int i= get_global_id(0);
3     in[i]++;
4 }

```

Po wykonaniu kernela odczytane zostały wartości `CL_PROFILING_COMMAND_QUEUED` i `CL_PROFILING_COMMAND_START`. Różnica tych dwóch to czas potrzebny przesłanie kernela do urządzenia i rozpoczęcie jego wykonania.

## 5. APLIKACJE WYKORZYSTUJĄCE OPENCL PRZYPADKI UŻYCIA

### 5.1. Pierwsza sekcja

#### 5.1.1. Podsekcja

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Rys. 5.1. Logo WETI

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

**Tabela 5.1.** Przykład krótkiej tabeli

Nagłówek 1	Nagłówek 2	Nagłówek 3
1	lubię plaki	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
2	a ja nie	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Rys. 5.2. Logo PG

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

#### 5.1.2. I kolejna podsekcja

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## 5.2. Druga sekcja

### 5.2.1. Podsekcja sekcji drugiej

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

**Tabela 5.2.** Przykład długiej tabeli

Nagłówek 1	Nagłówek 2
1	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
2	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
3	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nagłówek 1	Nagłówek 2
4	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
5	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
6	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

### 5.2.2. Kod

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << [](ostream& out) -> decltype(auto) { return out << "Hello
      world!\n"; } << endl;

```



```
7     return 0;  
8 }
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## 6. ANALIZA WYNIKÓW, WNIOSKI

### 6.1. Pierwsza sekcja

#### 6.1.1. Podsekcja

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Rys. 6.1. Logo WETI

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

**Tabela 6.1.** Przykład krótkiej tabeli

Nagłówek 1	Nagłówek 2	Nagłówek 3
1	lubię plaki	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
2	a ja nie	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Rys. 6.2. Logo PG

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

#### 6.1.2. I kolejna podsekcja

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## 6.2. Druga sekcja

### 6.2.1. Podsekcja sekcji drugiej

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

**Tabela 6.2.** Przykład długiej tabeli

Nagłówek 1	Nagłówek 2
1	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
2	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
3	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nagłówek 1	Nagłówek 2
4	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
5	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.
6	Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

#### 6.2.2. Kod

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << [](ostream& out) -> decltype(auto) { return out << "Hello
        world!\n"; } << endl;

```

```
7   return 0;  
8 }
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## WYKAZ RYSUNKÓW

3.1.	Logo WETI.....	15
3.2.	Logo PG.....	16
5.1.	Logo WETI.....	24
5.2.	Logo PG.....	25
6.1.	Logo WETI.....	29
6.2.	Logo PG.....	30

## WYKAZ TABEL

3.1.	Przykład krótkiej tabeli .....	16
3.2.	Przykład długiej tabeli .....	17
5.1.	Przykład krótkiej tabeli .....	25
5.2.	Przykład długiej tabeli .....	26
6.1.	Przykład krótkiej tabeli .....	30
6.2.	Przykład długiej tabeli .....	31



## **Dodatek A: PRZYKŁADOWY DODATEK**

### ***A.1 Sekcja***