



Laborator 11

Exercițiile din acest lab se vor rezolva folosind OpenCL, rezultatele vor fi adăugate în README.txt.

1. Implementați soluția OpenCL folosind pipeline pentru sortarea elementelor unui vector (**sort.c**). Se poate presupune că numărul de elemente din pipeline e mai mic decât numărul maxim de work items dintr-un work group. Practic se va programa ca și cum am avea un singur work group și ne putem folosi de toate beneficiile: bariere, shared memory, atomice locale, samd..
2. Implementați soluția OpenCL folosind pipeline pentru calculul unui polinom (**polynomialFunction.c**). Se poate presupune că numărul de elemente din pipeline e mai mic decât numărul maxim de work items dintr-un work group. Practic se va programa ca și cum am avea un singur work group și ne putem folosi de toate beneficiile: bariere, shared memory, atomice locale, samd..

Exercițiile de la 1 la 2 sunt obligatorii. Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

Vă recomandăm, pentru a crește șansele de a obține o notă cât mai mare să explorați și următoarele exerciții:

3. Implementați programele precedente presupunând că numărul de elemente din pipeline va depăși numărul de work items dintr-un work group. În acest caz nu vă puteți baza pe facilitățile care sunt disponibile doar la nivel de work group.

HINT:

Pentru primul exercițiu pipeline-ul va funcționa în felul următor (a se observa că sunt 2N iterații și că nu toate thread-urile sunt mereu active):

De sortat vectorul 82,34,68,0

```
I (LOCAL_VAL_THREAD0 IS_ACTIVE) -> BUFFER_VAL -> (LOCAL_VAL_THREAD1
0 (999 1) -> 999 -> (999 0) -> 999 -> (999 0) -> 999 -> (999 0) -> 0 ->
1 ( 82 1) -> 999 -> (999 0) -> 999 -> (999 0) -> 999 -> (999 0) -> 0 ->
2 ( 34 1) -> 82 -> (999 1) -> 999 -> (999 0) -> 999 -> (999 0) -> 0 ->
3 ( 34 1) -> 68 -> ( 82 1) -> 999 -> (999 0) -> 999 -> (999 0) -> 0 ->
4 (  0 0) -> 34 -> ( 68 1) -> 82 -> (999 1) -> 999 -> (999 0) -> 0 ->
5 (  0 0) -> 34 -> ( 34 0) -> 68 -> ( 82 1) -> 999 -> (999 0) -> 0 ->
6 (  0 0) -> 34 -> ( 34 0) -> 68 -> ( 68 0) -> 82 -> (999 1) -> 0 ->
7 (  0 0) -> 34 -> ( 34 0) -> 68 -> ( 68 0) -> 82 -> ( 82 0) -> 999 ->
```