Faculté des Sciences & Techniques — Université de Limoges

Master's 1st year

GPGPU's Dev

Examination — november 2021

Duration : 1h30 — Permitted access to documents

1 – 14pts We want to search for **words** in a **text** :

▷ the text is contained in an array `char *T` of N characters :

a	a	c	c	a	e	a	a	d	h	d	j	···	m	j	b	k	f	g	t

▷ the text to search is contained in the array `char *pattern`, its length is **variable**, indicated by the variable `int pattern_length` :

m	o	t	i	f

pattern_length

▷ the array `int positions` will contain the positions of the pattern in the text.
The value `-1` will be the initialization value of an undetermined position.

We will set the size of the table T to $N = 2^{20} = 1048576$.

Questions :

a. Give a **sequential version in C** of a function that performs the **search for the pattern** in the text and returns its **position** in the text. *(1,5pts)*
We will search only the first position of the pattern in the text.

We would now like to determine **in parallel** with CUDA, **all positions** where the pattern is located in the text.

b. Describe how you will perform the **search for the pattern in parallel** ? *(1,5pts)*

How the **positions** of each found pattern will be recorded ?

Can **conflicts** between threads occur ?
You will indicate :
- ◇ *the work done by each thread and the data it uses ;*
- ◇ *the size of the grid and the blocks.*

c. In your solution, are the **global memory accesses** of the graphics card **optimal** ? *(2pts)*

Is the use of the type `char` ill-advised ? What solution do you propose ?

d. Give a CUDA version **not using shared memory**. *(3pts)*

e. Can the use of **shared memory** be interesting ? *(2pts)*

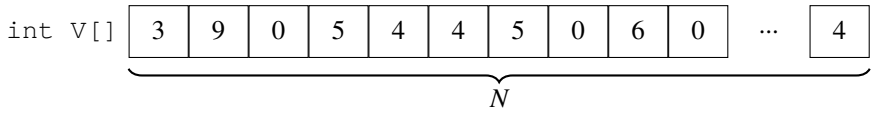
How can you use the shared memory to search the **different positions** of the pattern ?

Is it interesting to be able to adjust the size of the shared memory **dynamically** ?

Is the existence of the « *banks* » troublesome for your work ?

f. Give a CUDA version using the **shared memory**. *(4pts)*

2 – 6pts We want to find the value **min** and **max** of an array `int V[N]` of integer values.



We will choose $N = 1048576$.

Questions :

a. Describe a parallel solution in CUDA : *(2pts)*

- ◇ Is the use of the **shared memory** interesting ? Explain how you can exploit it.
- ◇ Can **conflicts** occur between the different threads ?
- ◇ How will you **divide the work** between the different threads ?
  You will indicate the grid and the blocks you will use.
- ◇ Are the accesses to the global memory **optimal** ?

b. Give a CUDA program that performs the search of the **min** and the **max** in the array `V[]` using the **shared memory**. *(4pts)*