

GPUs and Heterogeneous Systems – A.Y. 2023-24

Scuola di Ingegneria Industriale e dell'Informazione
Prof. Antonio Miele



POLITECNICO
MILANO 1863

Example of Exam - **SECOND PART OF THE EXAM**

Surname:	Name:	Personal Code:
----------	-------	----------------

Question	1	2	3	OVERALL
Max score	5	5	6	16
Score				

Instructions:

- This second part of the exam is “open book”. The students are allowed to use any material and notes.
- The students are allowed to use the laptop and the tablet. No extra devices (e.g., phones) are allowed. Please, shut down and store not allowed electronic devices.
- Students are not allowed to communicate with any other one or use Internet.
- Students can write in pen or pencil, any color, but avoid writing in red.
- Students can also use the laptop to code the test solution. In this case, please pay attention to the instructor’s instructions to submit the test solution.
- Any violation of the above rules will lead to the invalidation of the test.
- **Duration: 1 hour and 15 minutes**

Question 1

Implement a basic CUDA kernel function to accelerate the compute-intensive function in the following C program.

Question 2

Modify the main function to execute the CUDA kernel function defined in the former question. Set block size to 32.

Question 3

Implement a new CUDA kernel function to accelerate the compute-intensive function in the following C program by using the shared memory.

The source code can be downloaded from: XXX

```

/*
 * The kernel function to accelerate receives in input a vector of positive integers,
 * called A, together with its size, and a second empty vector of integers, B, of the
 * same size.
 * For each element i in A, the function saves in B[i] the value 1 if A[i] is greater
 * than all the neighbor values with an index between (i-DIST) and (i+DIST), bounds
 * included and if they exist; 0 otherwise. DIST is a constant value defined with a
 * macro.
 * The main function is a dummy program that receives as an argument the vector size,
 * instantiates and populates randomly A, invokes the above function, and shows
 * results.
 */

#include <stdio.h>
#include <stdlib.h>

#define MAXVAL 100
#define DIST 10

void compute(int *V, int *R, int num);

//kernel function: identify peaks in the vector
void compute(int *V, int *R, int num) {
    int i, j, ok;
    for(i=0; i<num; i++){
        for(j=-DIST, ok=1; j<=DIST; j++){
            if(i+j>=0 && i+j<num && j!=0 && V[i]<=V[i+j])
                ok=0;
        }
        R[i] = ok;
    }
}

int main(int argc, char **argv) {
    int *A;
    int *B;
    int dim;
    int i;

    //read arguments
    if(argc!=2){
        printf("Please specify sizes of the input vector\n");
        return 0;
    }
    dim=atoi(argv[1]);

    //allocate memory for the three vectors
    A = (int*) malloc(sizeof(int) * dim);
    B = (int*) malloc(sizeof(int) * dim);

    //initialize input vectors
    /*code omitted for the sake of space*/

    //execute on CPU
    compute(A, B, dim);

    //print results
    /*code omitted for the sake of space*/

    free(A);
    free(B);

    return 0;
}

```