Lab 2.1

Given the program **merge.c,** implement a threaded mergesort program, by replacing the recursive calls to **mergeSort** with thread activations**.**

Lab 2.2

Given your previous solution, implement a more efficient solution that does not create a new thread if the size of the sub-vector to be sorted is less than a given threshold, but just sort it. The threshold is given as an additional command line argument.

Lab 2.3

Implement a sequential program in C that takes a single argument **k** from the command line. The program creates two vectors (**v1** and **v2**) of dimension **k**, and a matrix (**mat**) of dimension **k**x**k**, which are filled with random numbers in the range **[-0.5 0.5]**, then it performs the product $\mathbf{v1^T * mat * v2}$, and print the result. This is an example for **k=5**:

$\mathbf{v1^T}$ **= [-0.0613 -0.1184 0.2655 0.2952 -0.3131]**

mat=[-0.3424 -0.3581 0.1557 0.2577 0.2060

0.4706 -0.0782 -0.4643 0.2431 -0.4682

0.4572 0.4157 0.3491 -0.1078 -0.2231

-0.0146 0.2922 0.4340 0.1555 -0.4538

0.3003 0.4595 0.1787 -0.3288 -0.4029]

$\mathbf{v2^T}$ **= [-0.3235 0.1948 -0.1829 0.4502 -0.4656]**

Result: 0.0194

Perform the product operation as **v = mat * v2**, which produces a new vector **v**, and $\mathbf{result = v1^T * v}$

Then, write a concurrent program using threads that performs the same task. The main thread creates the vectors, the matrix, and **k** threads. Then it exits without waiting the other threads.

Each thread **i** performs the product of the **i-th** row vector of **mat** and **v2,** which produces the **i-th** element of vector **v**.

One of the threads, the **last** terminating its product operation, performs the final operation $\mathbf{result= v1^T * v}$, and prints the result.

Create a compressed **`tgz`** archive file with this syntax:

Lab <LAB_number>_<Your_Id_number>_<YourLastname>. Tgz

Example:

`Lab02_123456_Smith.tgz`

which includes all the source programs, and if necessary the input data, for each assignment.

For example: **`Lab2-123456_Smith.tgz`** must include **`Lab2.1.c, Lab2.2.c, Lab2.3.c`** and any other useful file.

Drop your **`tgz file`** in the folder **Elaborati** of the SDP site of Portale della didattica.