Test the performance of the following codes

C code

In the Lab you can use MinGW Development Environment, installed under Inicio > Programas > Desarrollo de software.

Exercise C1: "compact" expressions

```
sum += x * y;  vs  sum = x * y + sum;

sum+=*p++;   vs   sum += p[i];
```

Exercise C2: Use of cache

Compare the performance of the following sorting algorithms

```
//Sort1: Bubble method (close exchanges)
for(x=0; x<n-1; x++)  {
   for(y=0; y<n-x-1; y++)  {
      if(array[y]>array[y+1])  {
         temp = array[y+1];
         array[y+1] = array[y];
         array[y] = temp;
      }
   }
}

//Sort2: Long exchanges
for(x=0; x<n-1; x++)  {
   for(y=x+1; y<n; y++)  {
      if(array[x]>array[y])  {
         temp = array[x];
         array[x] = array[y];
         array[y] = temp;
      }
   }
}
```

Exercise C3: Memory allocation

Analyze the degradation of performance due to memory reallocation.

```
    numbers=NULL;
    //numbers = (int*) realloc (numbers, N * sizeof(int)); //Good 1.8 s
    for (i=1; i<N; i++)  {
        numbers = (int*) realloc (numbers, i * sizeof(int)); //Bad 17 s

        if (numbers!=NULL) {
            numbers[i-1]=rand();
        }
        else {
            //free (numbers);
            puts ("Error (re)allocating memory");
            exit (1);
        }
    }
    free(numbers);
```

Matlab code

Use tic and toc to make execution-time measurements.

You may need to use a loop to make accurate measurement. For example:

```
>> tic, for t=1:100, prueba2, end, toc
Elapsed time is 10.846000 seconds
```

Exercise M1: Use of cache

Compare the performance of the following sorting algorithms

```
//Add by columns
suma=0;
for i=1:1000
    for j=1:1000
        suma=suma+z(i,j);
    end
end

//Add by rows
suma=0;
for i=1:1000
    for j=1:1000
        suma=suma+z(j,i);
    end
end

//Add internal loops. How many cores does it use?
suma=sum(z(:));
```

Exercise M2: Virtual memory

Try the following code for different values of sz. Use Task Manager to monitor memory use.

(Matlab may stop responding when you hit the memory limit)

```
sz=100000000;
x=rand(sz,1); tic, x=sort(x); toc
tic, x=rand(sz,1); x=sort(x); toc
```

Exercise M3: Cache and Virtual memory conflicts

Run two or more matlab sessions in parallel (or C programs in parallel) compare the performance with running the same programs one after another.

Programs block each other while trying to access memory. More remarkable (but more difficult to simulate) is while accessing the hard drive.

Another way to test disk access conflicts is executing to copy commands in parallel.

Exercise M4: Memory allocation

Measure execution time in the following cases.

Use Mlint to detect pre-allocation suggestions in Matlab

```
//Without memory allocation. Matlab uses realloc
for i=1:1000
    for j=1:1000
        z2(j,i)=z(j,i);
    end
end


//With memory allocation
z2=ones(size(z));
for i=1:1000
    for j=1:1000
        z2(j,i)=z(j,i);
    end
end
```

Webpage analysis

Measure loading times with Google Chrome by activating developer tools.

-Try loading different pages, ex. www.iit.upcomillas.es and www.upcomillas.es

-Try reloading, and reloading without cache.

-Try setting mobile behavior: Resolution and Network speed.

Browser comparison

Run tests in different browsers (may include mobile phone) and compare results.

Exercise B1- Javascript

https://www.webkit.org/perf/sunspider/sunspider.html

Exercise B2- CSS performance

http://ie.microsoft.com/TEStdrive/Performance/MazeSolver/Default.html

Laboratory report

- Program listings
- Time measurements