Київський національний університет імені Тараса Шевченка

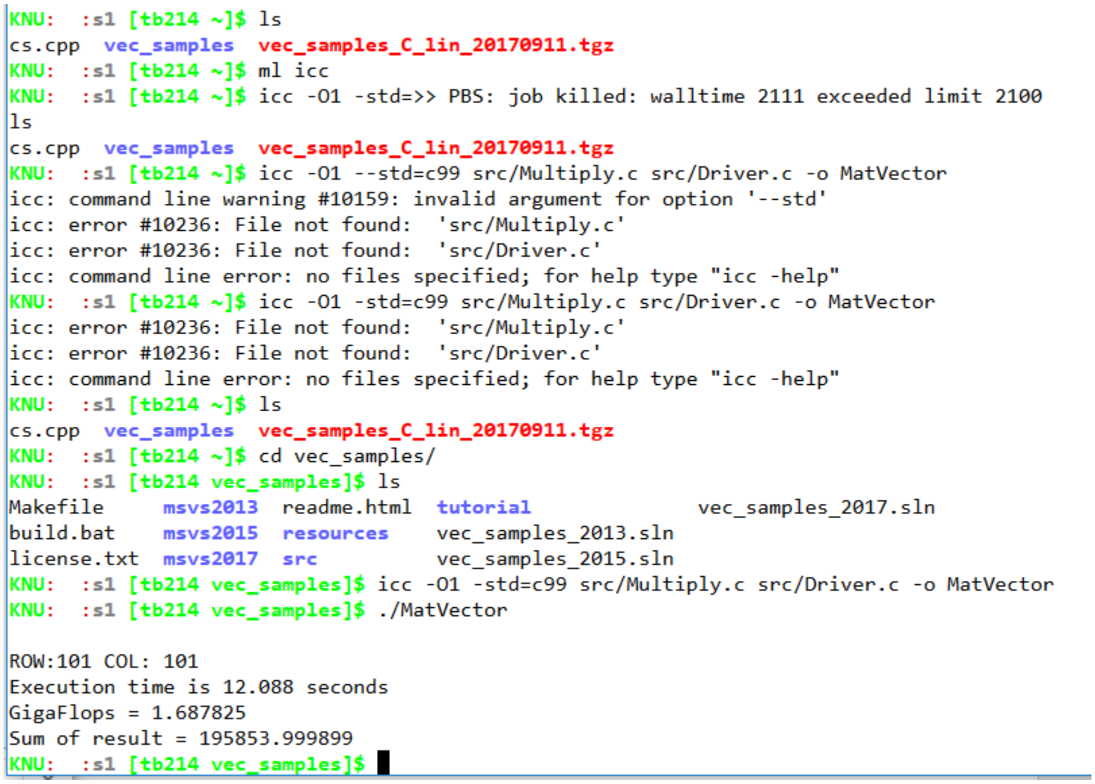
Факультет радіофізики, електроніки та компютерних систем

Лабораторна робота №3
Дослідження оптимізації коду з використанням векторних розширень CPU

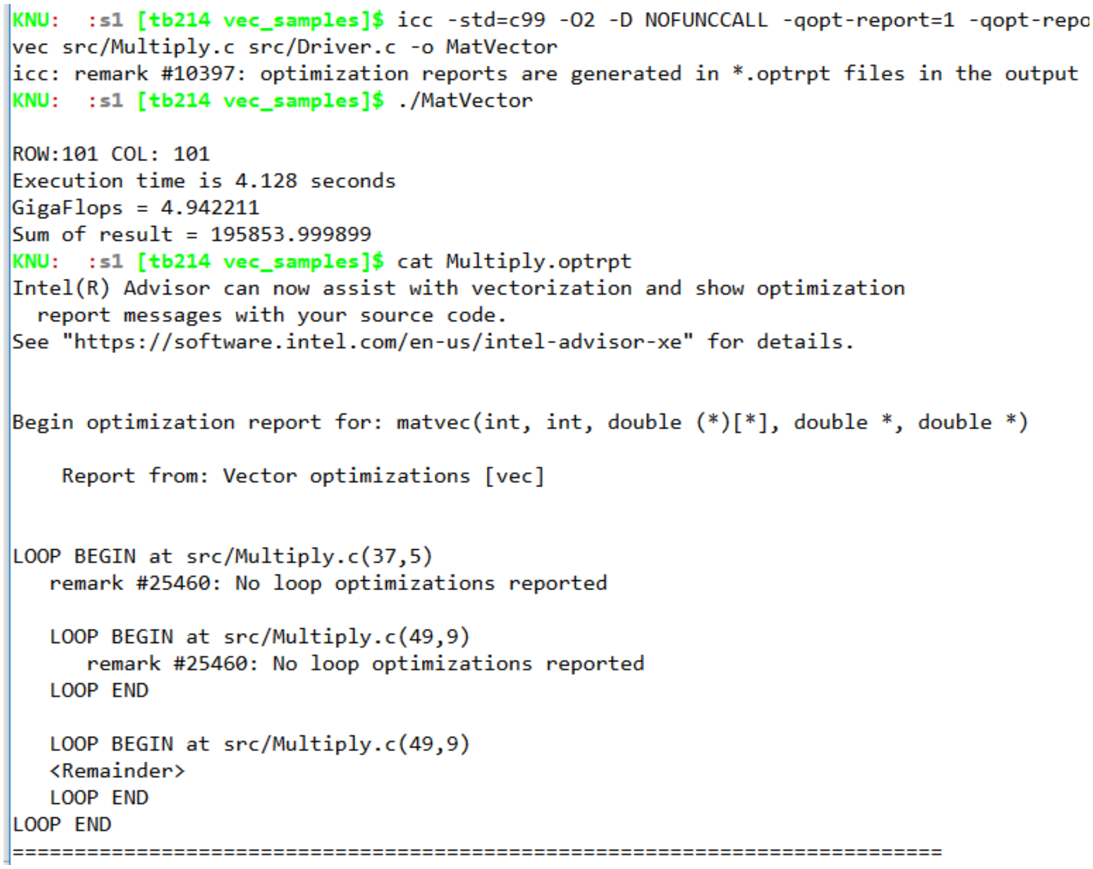
Виконав студент

3 курсу СА-КІ

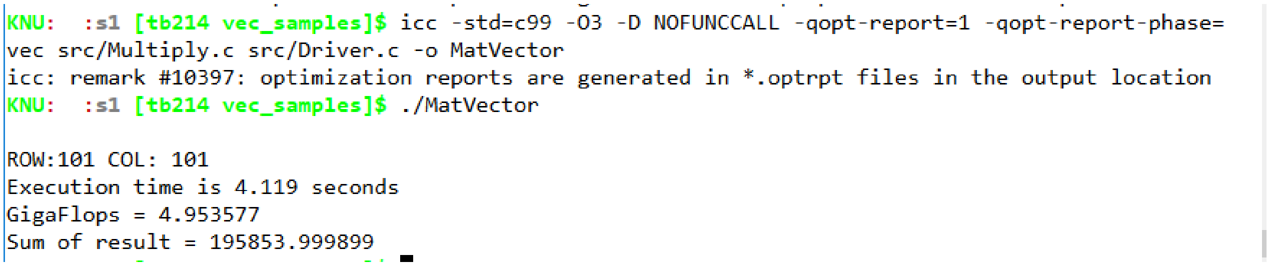
Глушко Гліб



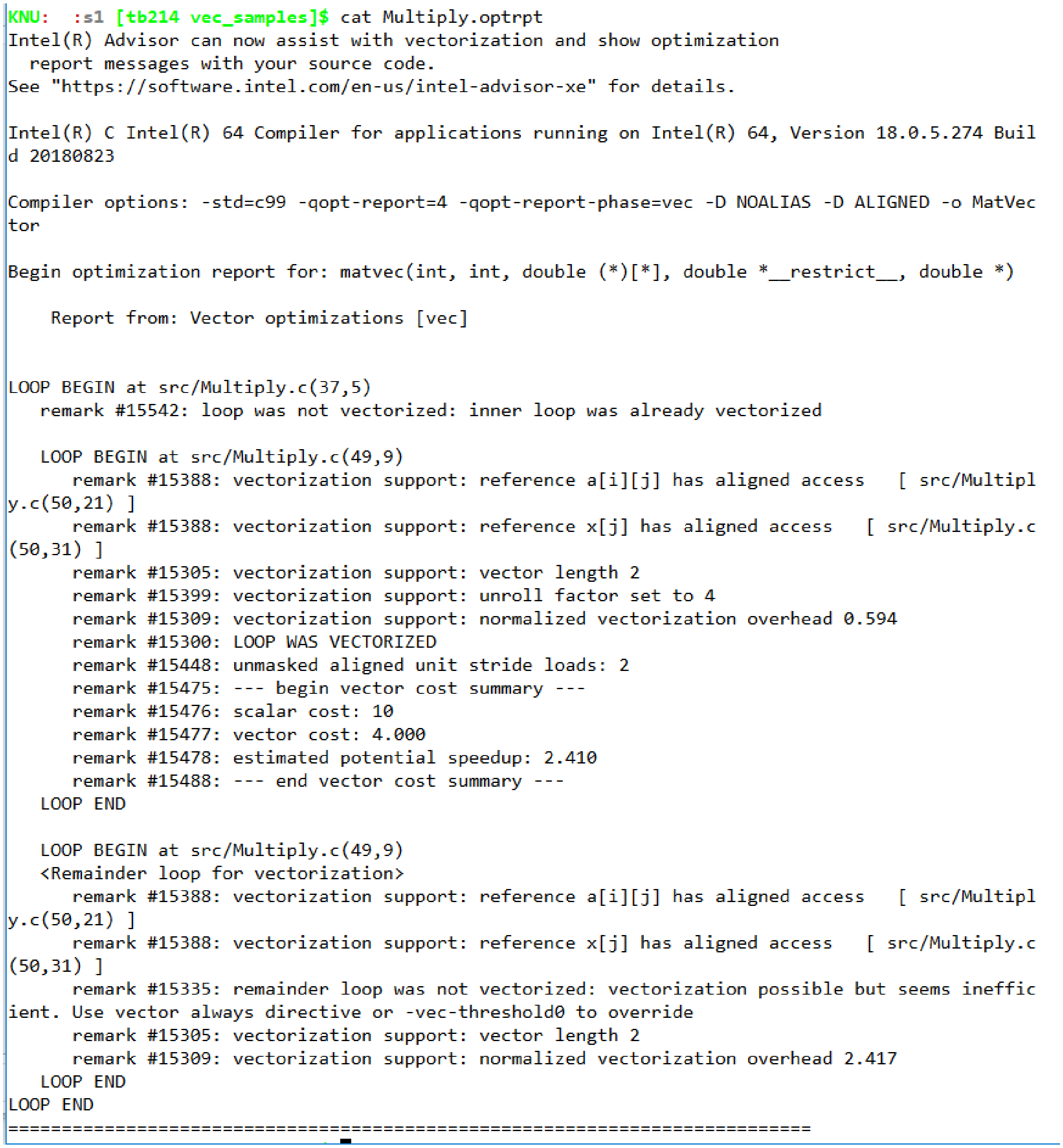
```
KNU:  :s1 [tb214 ~]$ ls
cs.cpp  vec_samples  vec_samples_C_lin_20170911.tgz
KNU:  :s1 [tb214 ~]$ ml icc
KNU:  :s1 [tb214 ~]$ icc -O1 -std=>> PBS: job killed: walltime 2111 exceeded limit 2100
ls
cs.cpp  vec_samples  vec_samples_C_lin_20170911.tgz
KNU:  :s1 [tb214 ~]$ icc -O1 --std=c99 src/Multiply.c src/Driver.c -o MatVector
icc: command line warning #10159: invalid argument for option '--std'
icc: error #10236: File not found:  'src/Multiply.c'
icc: error #10236: File not found:  'src/Driver.c'
icc: command line error: no files specified; for help type "icc -help"
KNU:  :s1 [tb214 ~]$ icc -O1 -std=c99 src/Multiply.c src/Driver.c -o MatVector
icc: error #10236: File not found:  'src/Multiply.c'
icc: error #10236: File not found:  'src/Driver.c'
icc: command line error: no files specified; for help type "icc -help"
KNU:  :s1 [tb214 ~]$ ls
cs.cpp  vec_samples  vec_samples_C_lin_20170911.tgz
KNU:  :s1 [tb214 ~]$ cd vec_samples/
KNU:  :s1 [tb214 vec_samples]$ ls
Makefile     msvs2013  readme.html  tutorial              vec_samples_2017.sln
build.bat    msvs2015  resources    vec_samples_2013.sln
license.txt  msvs2017  src          vec_samples_2015.sln
KNU:  :s1 [tb214 vec_samples]$ icc -O1 -std=c99 src/Multiply.c src/Driver.c -o MatVector
KNU:  :s1 [tb214 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 12.088 seconds
GigaFlops = 1.687825
Sum of result = 195853.999899
KNU:  :s1 [tb214 vec_samples]$ █
```



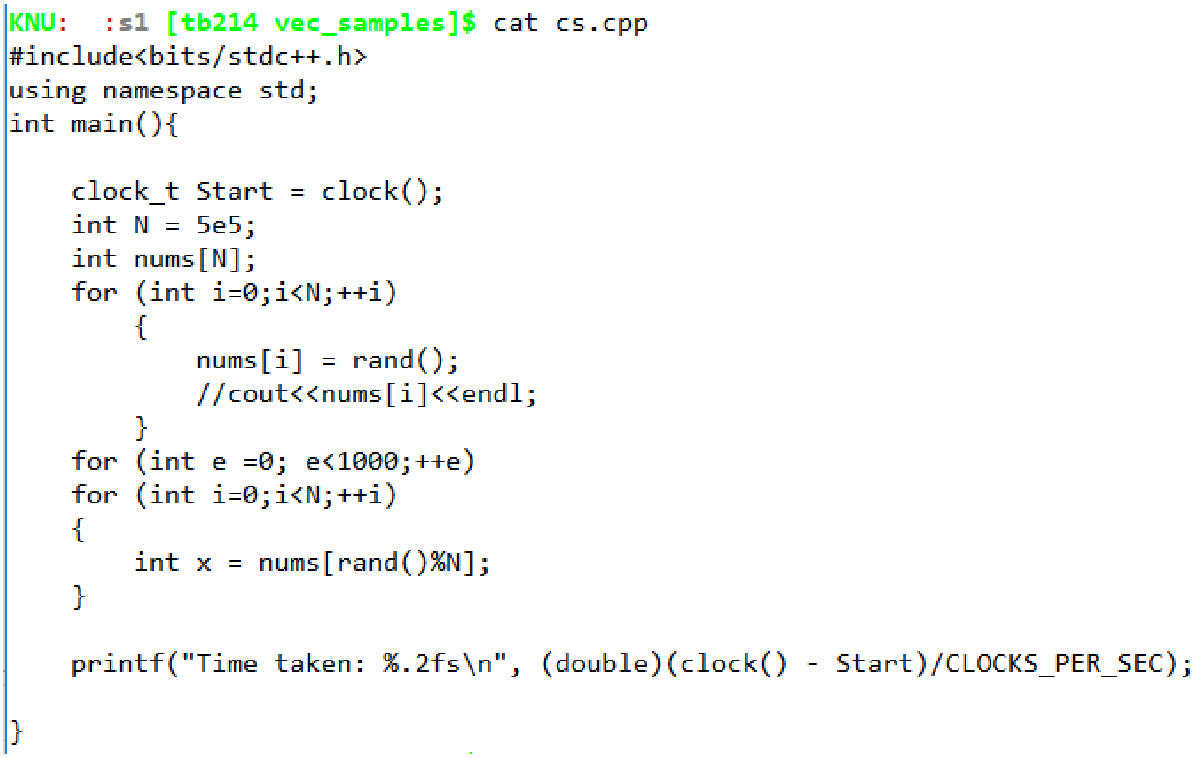
```
KNU:  :s1 [tb214 vec_samples]$ icc -std=c99 -O2 -D NOFUNCCALL -qopt-report=1 -qopt-repo
vec src/Multiply.c src/Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output
KNU:  :s1 [tb214 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 4.128 seconds
GigaFlops = 4.942211
Sum of result = 195853.999899
KNU:  :s1 [tb214 vec_samples]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.


Begin optimization report for: matvec(int, int, double (*)[*], double *, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Multiply.c(37,5)
   remark #25460: No loop optimizations reported

   LOOP BEGIN at src/Multiply.c(49,9)
      remark #25460: No loop optimizations reported
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
   <Remainder>
   LOOP END
LOOP END
===========================================================================
```



```
KNU:  :s1 [tb214 vec_samples]$ icc -std=c99 -O3 -D NOFUNCCALL -qopt-report=1 -qopt-report-phase=
vec src/Multiply.c src/Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output location
KNU:  :s1 [tb214 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 4.119 seconds
GigaFlops = 4.953577
Sum of result = 195853.999899
```

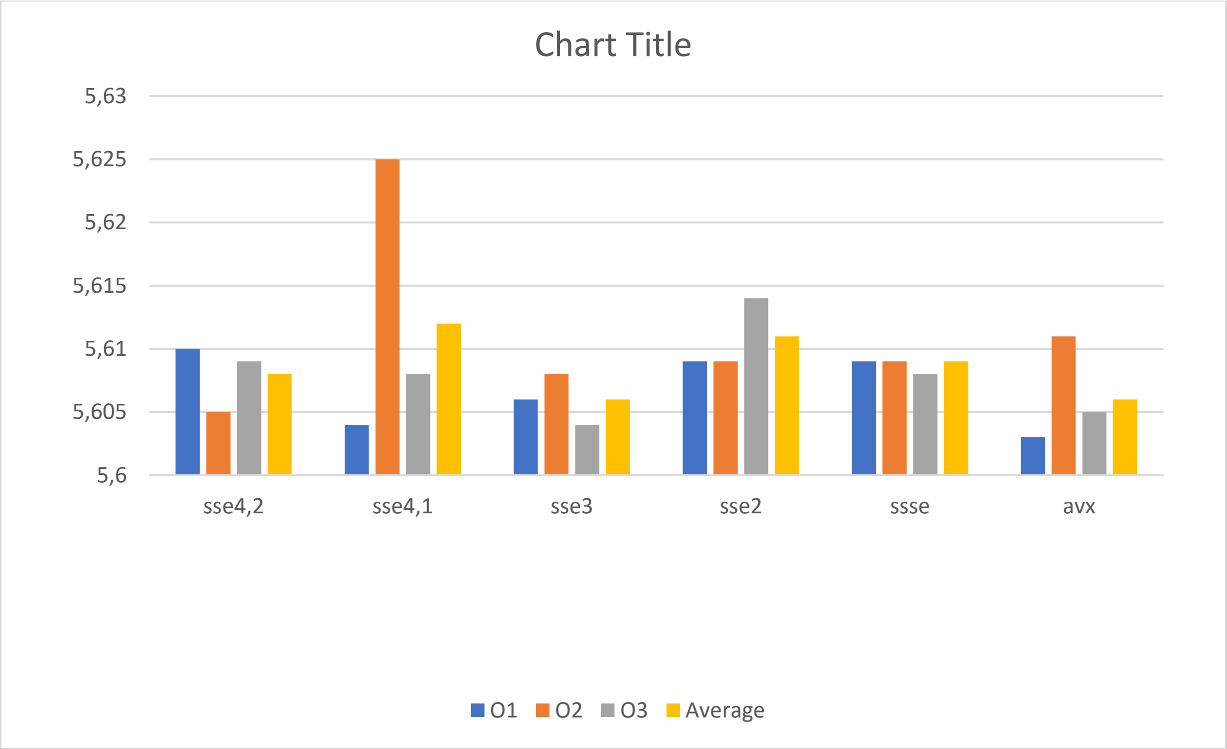


```
KNU:  :s1 [tb214 vec_samples]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.

Intel(R) C Intel(R) 64 Compiler for applications running on Intel(R) 64, Version 18.0.5.274 Buil
d 20180823

Compiler options: -std=c99 -qopt-report=4 -qopt-report-phase=vec -D NOALIAS -D ALIGNED -o MatVec
tor

Begin optimization report for: matvec(int, int, double (*)[*], double *__restrict__, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Multiply.c(37,5)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at src/Multiply.c(49,9)
      remark #15388: vectorization support: reference a[i][j] has aligned access   [ src/Multipl
y.c(50,21) ]
      remark #15388: vectorization support: reference x[j] has aligned access   [ src/Multiply.c
(50,31) ]
      remark #15305: vectorization support: vector length 2
      remark #15399: vectorization support: unroll factor set to 4
      remark #15309: vectorization support: normalized vectorization overhead 0.594
      remark #15300: LOOP WAS VECTORIZED
      remark #15448: unmasked aligned unit stride loads: 2
      remark #15475: --- begin vector cost summary ---
      remark #15476: scalar cost: 10
      remark #15477: vector cost: 4.000
      remark #15478: estimated potential speedup: 2.410
      remark #15488: --- end vector cost summary ---
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
   <Remainder loop for vectorization>
      remark #15388: vectorization support: reference a[i][j] has aligned access   [ src/Multipl
y.c(50,21) ]
      remark #15388: vectorization support: reference x[j] has aligned access   [ src/Multiply.c
(50,31) ]
      remark #15335: remainder loop was not vectorized: vectorization possible but seems ineffic
ient. Use vector always directive or -vec-threshold0 to override
      remark #15305: vectorization support: vector length 2
      remark #15309: vectorization support: normalized vectorization overhead 2.417
   LOOP END
LOOP END
===========================================================================
```



```
KNU:   :s1 [tb214 vec_samples]$ cat cs.cpp
#include<bits/stdc++.h>
using namespace std;
int main(){

    clock_t Start = clock();
    int N = 5e5;
    int nums[N];
    for (int i=0;i<N;++i)
        {
            nums[i] = rand();
            //cout<<nums[i]<<endl;
        }
    for (int e =0; e<1000;++e)
    for (int i=0;i<N;++i)
    {
        int x = nums[rand()%N];
    }

    printf("Time taken: %.2fs\n", (double)(clock() - Start)/CLOCKS_PER_SEC);

}
```

	sse4.2	sse4.1	sse3	sse2	ssse	avx
O1	5.610	5.604	5.606	5.609	5.609	5.603
O2	5.605	5.625	5.608	5.609	5.609	5.611
O3	5.609	5.608	5.604	5.614	5.608	5.605
average	5.608	5.612	5.606	5.611	5.609	5.606



Висновок: Під час лабораторної роботи ми ознайомились з роботою на обчислювальному кластері та методами оптимізації виконання програм на С та С++.