**Отчет по лабораторной работе № 2**

Технологии векторного программирования SIMD

Студенты:

Потапова Алина,

Запорожская Елисавета

Группа:

3530202/90002

**Код программы**

#include <stdio.h>

#include <conio.h>

#include <iostream>

#include <xmmintrin.h>

#include <mmintrin.h>

#include <immintrin.h>

#include <dvec.h>

**int** main()

{

**char** qw1[8] = { 1, 1, 1, 1, 1, 1, 1, 1 };

**char** qw2[8] = { 2, 2, 2, 2, 2, 2, 2, 2 };

**float** c[4] = { 1, 2, 3, 4 };

**float** d[4] = { 5, 6, 7, 8 };

**double** f[2] = { 16, 4 };

**char** a128[16] = { 1, 18, 3, 19, 5, 21, 7, 23, 9, 25, 11, 27, 13, 29, 15, 31 };

**char** b128[16] = { 17, 2, 19, 4, 21, 6, 23, 8, 25, 10, 27, 12, 29, 14, 31, 16 };

\_asm

{

movq mm0, qw1

movq mm1, qw2

pcmpeqb mm0, mm1

movq qw1, mm0

}

printf("%s\n", "Summing elements of vectors qw1 + qw2 :");

**for** (**int** i = 0; i < 8; i++)

{

printf("%d ", qw1[i]);

}

printf("\n\n");

\_asm

{

movups xmm0, c

movups xmm1, d

addps xmm0, xmm1

movups c, xmm0

}

printf("%s\n", "Summing elements of vectors c + d :");

**for** (**int** i = 0; i < 4; i++)

{

printf("%f ", c[i]);

}

printf("\n");

\_asm

{

movups xmm1, f

sqrtpd xmm0, xmm1

movups f, xmm0

}

printf("\n%s %f %s %f\n", "Square root of ", f[0], "is", f[1]);

\_asm

{

movups xmm0, a128

movups xmm1, b128

pminub xmm0, xmm1

movups a128, xmm0

}

printf("\n%s\n", "Comparing elements :");

**for** (**int** i = 0; i < 16; i++)

{

printf("( %d , %d) ; ", a128[i], b128[i]);

}

printf("\n");

printf("\n%s\n", "Minimum elements :");

**for** (**int** i = 0; i < 16; i++)

{

printf("%d ", a128[i]);

}

//AVX

printf("\n%s\n", "AVX :");

\_\_m256 evens = \_mm256\_set\_ps(2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0);

\_\_m256 odds = \_mm256\_set\_ps(1.0, 3.0, 5.0, 7.0, 9.0, 11.0, 13.0, 15.0);

\_\_m256 result = \_mm256\_sub\_ps(evens, odds);

**float**\* av = (**float**\*)&result;

printf("%f %f %f %f %f %f %f %f\n",

av[0], av[1], av[2], av[3], av[4], av[5], av[6], av[7]);

printf("\n\n");

//dvec

F64vec4 A[2]{ {1,2,3,4},{3,4,3,4} }, B[2]{ {1,2,3,4},{3,4,3,4} }, C[2]{ {1,2,3,4},{3,4,3,4} };

**for** (**int** i = 0; i < 8; i++)

{

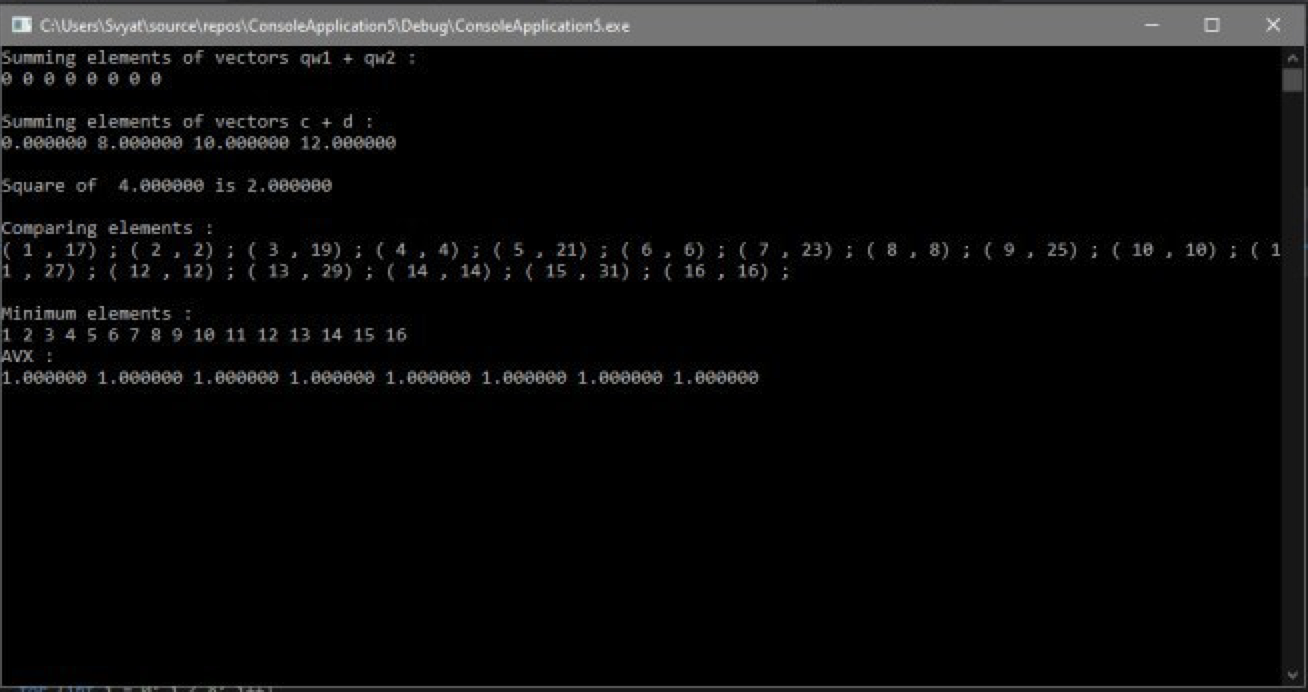
C[i] = A[i] + B[i];

}

**return** 0;

}

**Пример выполнения программы**

****