**Задание 5**

#include <stdlib.h>

#include <omp.h>

#include <iostream>

using namespace std;

const int n = 6;

const int m=8;

double mean(int d[][m])

{

int s = 0;

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

s += d[i][j];

return s/(n\*m);

}

int Min(int d[][m])

{

int min = d[0][0];

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

if (d[i][j] < min)

min = d[i][j];

return min;

}

int Max(int d[][m])

{

int max = d[0][0];

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

if (d[i][j] > max)

max = d[i][j];

return max;

}

int Kol(int d[][m])

{

int kol = 0;

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

if (d[i][j] % 3 == 0)

kol++;

return kol;

}

int main()

{

int d[6][8];

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

d[i][j] = rand() % 10;

#pragma omp parallel

{

#pragma omp sections

{

#pragma omp section

{

cout<<omp\_get\_thread\_num() << " Mean value = " << mean(d) << endl;

}

#pragma omp section

{

printf("\n%d Minimal value = %d Maximal value = %d\n",omp\_get\_thread\_num(), Min(d), Max(d));

}

#pragma omp section

{

printf("\n%d Kol elements = %d\n",omp\_get\_thread\_num(), Kol(d));

}

}

}

}