**#9**

#include <iostream>

#include <omp.h>

#include <cstdlib>

#include <ctime>

int main() {

int d[6][8];

srand(time(0));

for (int i = 0; i < 6; ++i)

for (int j = 0; j < 8; ++j)

d[i][j] = rand() % 100 + 1;

omp\_set\_num\_threads(4);

for (int i = 0; i < 6; ++i) {

for (int j = 0; j < 8; ++j) {

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

}

printf("\n");

}

int min, max;

int lmax, lmin;

min = d[0][0];

max = d[0][0];

#pragma omp parallel for shared(min,max) private(lmax,lmin)

for (int i = 0; i < 6; ++i) {

lmax = d[i][0];

lmin = d[i][0];

for (int j = 1; j < 8; ++j) {

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

lmin = d[i][j];

else if (d[i][j] > lmax)

lmax = d[i][j];

}

#pragma omp critical

{

if (lmax > max)

max = lmax;

else if (lmin < min)

min = lmin;

}

}

printf("min: %i\n", min);

printf("max: %i\n", max);

return 0;

}

**#10**

#include <omp.h>

#include <iostream>

int main() {

int a[30];

for (int i = 0; i < 30; ++i)

a[i] = rand() % 50 + 1;

for (int i = 0; i < 30; ++i)

printf(" %i", a[i]);

printf("\n");

int k = 0;

omp\_set\_num\_threads(4);

#pragma omp parallel for

for (int i = 0; i < 30; ++i) {

if (a[i] % 9 == 0) {

#pragma omp atomic

++k;

}

}

printf("kol-vo elementov kratnih 9: %i", k);

return 0;

}

**#11**

#include <omp.h>

#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

srand(time(0));

int a[10];

for (int i = 0; i < 10; ++i)

a[i] = rand() % 50 + 1;

for (int i = 0; i < 10; ++i)

printf(" %i", a[i]);

printf("\n");

int max = 0;

omp\_set\_num\_threads(8);

#pragma omp parallel for

for (int i = 0; i < 10; ++i) {

if (a[i] % 7 == 0) {

#pragma omp critical(zone)

{

if (a[i] > max)

max = a[i];

}

}

}

if (max == 0)

printf("net max elementa kratnogo 7");

else

printf("max elem : 7 is - %i", max);

return 0;

}