#include <omp.h>

#include <iostream>

const int n = 12;

void Print(int\* a)

{

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

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

printf("\n");

}

int main()

{

int c\_size = 3;

int a[n], b[n], c[n];

omp\_set\_num\_threads(3);

printf("It is first region\n");

#pragma omp parallel for schedule(static, c\_size)

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

{

a[i] = rand() % 10;

b[i] = rand() % 10;

printf("Kol-vo = %d. Thread = %d. A[%d] = %d. B[%d] = %d.\n",omp\_get\_num\_threads(), omp\_get\_thread\_num(), i, a[i], i, b[i]);

}

printf("\nIt is second region\n");

omp\_set\_num\_threads(4);

#pragma omp parallel for schedule(dynamic, c\_size)

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

{

c[i] = a[i] + b[i];

printf("Kol-vo = %d. Thread = %d. C[%d] = %d\n", omp\_get\_num\_threads(), omp\_get\_thread\_num(), i, c[i]);

}

Print(a);

Print(b);

Print(c);

return 0;

}