#include <stdio.h>

#include<omp.h>

int main()

{

int thread\_id,i=1,x=1,j=7,temp,lt;

int a=72,b,c=0;

int d=1, e=0;

int partial\_Sum, total\_Sum;

#pragma omp parallel private(thread\_id)

{

printf("Hello from private thread : %d\n",omp\_get\_thread\_num());

}

#pragma omp parallel for shared(a,b,c) private(temp)

for(i=1;i<3;i++)

{

temp = 2\*a;

b= temp+7;

c= c+b+temp;

}

printf("private var temp: %d\n",temp);

printf("shared var\n");

printf("A : %d\n",a);

printf("B : %d\n",b);

printf("C : %d\n",c);

#pragma omp parallel for firstprivate(j)

for(i=1; i<=3; i++)

{

if(i == 1 || i == 5)

j = j + 1;

d= d\*j;

}

printf("firstprivate j : %d\n",j);

#pragma omp parallel for lastprivate(x)

for(i=1; i<=3; i++)

{

x = 8\*i;

e = e+x+1;

}

printf("lastprivate x : %d\n",x);

lt = x;

return 0;

}