Lab4 - openMP

e.g. 1 compile and run

```
int printf(char *, ...);

main()
{
    int nt, tid;
    omp_set_num_threads(4);
    #pragma omp parallel
    {
        nt = omp_get_num_threads();
        tid = omp_get_thread_num();
        printf("%d %d₩n", nt, tid);
    }
}
```

(1) cc –fopenmp C-prog ***compile***

(2) #pragma omp parallel ***omp clause***

(3) omp_set_num_threads() ***set number of threads***

(4) omp_get_num_threads() ***get number of thread***

(5) omp_get_thread_num() ***get thread ID***

e.g. 2 private and shared variables

```
int printf(char *, ...);
main()
{
    int tid, sum=0;
    omp_set_num_threads(4);
    #pragma omp parallel private(tid) shared(sum)
    {
        tid = omp_get_thread_num();
        #pragma omp critical //atomic – this line may not be required
        sum += tid;
    }
    printf("%d₩n", sum);
}
```

e.g. 3 #omp for – openMP for loop

```
#define N 12

int printf(char *, ...);

main()
{
    int tid, A[N],i;
    omp_set_num_threads(4);

    #pragma omp parallel for
    for (i=0; i<N; i++)
        A[i] = N-i-1;

    #pragma omp parallel private(tid)
    {
        tid = omp_get_thread_num();
        #pragma omp for
        for (i=0; i<N; i++)
            printf("%d %d %d₩n", tid, i, A[i]);
    }
}
```

e.g. 4 if N is not divisible by #threads

```
#define N 8

int printf(char *, ...);

main()
{
    int tid, A[N], i;
    omp_set_num_threads(5);   // 4

    #pragma omp parallel private(tid)
    {
        tid = omp_get_thread_num();
        #pragma omp for
        for (i=0; i<N; i++)
            printf("%d %d₩n", tid, i);
    }
}
```

ex. add arrays

Design an openMP program to add two arrays into an array.

Your command to run the program is

add #threads #size_of_arrays

```
// initialization of A and B
for (i=0; i<N; i++) {
    A[i] = i;
    B[i] = N-i;
}

// openMP parallelizaion
for (i=0; i<N; i++)
     C[i] =   A[i] + B[i];
```

After done, submit your program.