Answer sheet for OpenMP exercise

1. openmp lets compiler to decide about making thread (implicit coding) and openmp is easier to migrate from single thread to multithread.
2. Ideal speed up must equal to number of logical core that we have **but** there are locking of a resource on `printf` function that work like writing on 1 resource.

```
// edited code
#pragma omp parallel for
for(int i=0; i<100000; i++){
    a[i] = 2*i+i;
    printf("a[%d], %d\n", i, a[i]);
}
```

3.
   - **Loop construct**:the loop will splitted into smaller block of code and free thread able to grab that parts by graping is defined by scheduler
   - **Sections**: a region of code that one/many thread(s) able to grab and run.
   - **Single**: only let one thread do and other will wait for it or **nowait**(if specify)
   - **Workshare**: a section of code which able to divided into smaller and many/one thread ablr to grab it.