Assume that len % k == 0

When len is small, the memcpy function in C library and single threading have similar throughput while multithreading has very low throughput. When len and k is large, the throughput of multithreading increases especially for multithreading with affinity. At some point, both multithreading with affinity and multithreading has a higher throughput than that of single threading and memcpy from C library. Note that when len and k is large, memcpy have a bottleneck and have a lower throughput compared to single and multithreading.

```
cylee@workbench:~/Desktop$ ./hw2 10240000 4
Vector size=10240000    threads num=4.
[C library: memcpy]start
[C library: memcpy]The throughput is 63.54 Gbps.
[Singlethreading]start
[Singlethreading]The throughput is 108.68 Gbps.
[Multithreading]start
[Multithreading]The throughput is 152.68 Gbps.
[Multithreading with affinity]start
Main thread runs on CPU 75
Set affinity mask to include CPUs (1, 3, 5, ...  2n+1)
[Multithreading with affinity]The throughput is 215.07 Gbps.
```

```
cylee@workbench:~/Desktop$ ./hw2 10240000 2
Vector size=10240000    threads num=2.
[C library: memcpy]start
[C library: memcpy]The throughput is 61.65 Gbps.
[Singlethreading]start
[Singlethreading]The throughput is 107.11 Gbps.
[Multithreading]start
[Multithreading]The throughput is 92.49 Gbps.
[Multithreading with affinity]start
Main thread runs on CPU 48
Set affinity mask to include CPUs (0, 2, 4, ...  2n)
[Multithreading with affinity]The throughput is 118.96 Gbps.
```

```
cylee@workbench:~/Desktop$ ./hw2 1024 4
Vector size=1024        threads num=4.
[C library: memcpy]start
[C library: memcpy]The throughput is 48.69 Gbps.
[Singlethreading]start
[Singlethreading]The throughput is 44.34 Gbps.
[Multithreading]start
[Multithreading]The throughput is 0.13 Gbps.
[Multithreading with affinity]start
Main thread runs on CPU 7
Set affinity mask to include CPUs (1, 3, 5, ...  2n+1)
[Multithreading with affinity]The throughput is 0.16 Gbps.
```