Q1:

```
Output from main_q1
----------------
Running main() from ./googletest-main/googletest/src/gtest_main.cc
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from testQ1
[ RUN      ] testQ1.test
Parallel
Sum Even: 757361650
Sum Odd: 742539102
Time: 0.00845482
Serial
Sum Even: 757361650
Sum Odd: 742539102
Time: 0.120326
[       OK ] testQ1.test (525 ms)
[----------] 1 test from testQ1 (525 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (525 ms total)
[  PASSED  ] 1 test.
```

Q2:

```
Output from main_q2
----------------
Running main() from ./googletest-main/googletest/src/gtest_main.cc
[==========] Running 7 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 7 tests from testQ2
[ RUN      ] testQ2.test1
[       OK ] testQ2.test1 (13 ms)
[ RUN      ] testQ2.test2
[       OK ] testQ2.test2 (4 ms)
[ RUN      ] testQ2.test3
[       OK ] testQ2.test3 (1 ms)
[ RUN      ] testQ2.test4
[       OK ] testQ2.test4 (5 ms)
[ RUN      ] testQ2.test5
[       OK ] testQ2.test5 (10 ms)
[ RUN      ] testQ2.serialSortTest
Serial Radix Sort: PASS
stl: 0.309157
serial radix: 0.0455599
[       OK ] testQ2.serialSortTest (523 ms)
[ RUN      ] testQ2.parallelSortTest
Parallel Radix Sort: PASS
stl: 0.306518
parallel radix: 0.0304899
[       OK ] testQ2.parallelSortTest (498 ms)
[----------] 7 tests from testQ2 (1057 ms total)

[----------] Global test environment tear-down
[==========] 7 tests from 1 test suite ran. (1057 ms total)
[  PASSED  ] 7 tests.
```

Q2_part6:

```
Output from main_q2_part6
-----------------
Running main() from ./googletest-main/googletest/src/gtest_main.cc
[==========] Running 8 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 8 tests from testQ2
[ RUN      ] testQ2.test1
[       OK ] testQ2.test1 (4 ms)
[ RUN      ] testQ2.test2
[       OK ] testQ2.test2 (3 ms)
[ RUN      ] testQ2.test3
[       OK ] testQ2.test3 (0 ms)
[ RUN      ] testQ2.test4
[       OK ] testQ2.test4 (3 ms)
[ RUN      ] testQ2.test5
[       OK ] testQ2.test5 (5 ms)
[ RUN      ] testQ2.serialSortTest
Serial Radix Sort: PASS
stl: 0.307474
serial radix: 0.045176
[       OK ] testQ2.serialSortTest (516 ms)
[ RUN      ] testQ2.parallelSortTest
Parallel Radix Sort: PASS
stl: 0.306065
parallel radix: 0.0264421
[       OK ] testQ2.parallelSortTest (498 ms)
[ RUN      ] testQ2.radixSortTest
Threads Blocks / Timing
         1       2       4       8       12      16      24      32      40      48
   1    0.042   0.037   0.037   0.048   0.073   0.099   0.192   0.301   0.347   0.384
   2    0.037   0.020   0.021   0.025   0.043   0.056   0.098   0.150   0.175   0.205
   4    0.049   0.027   0.017   0.018   0.030   0.033   0.059   0.090   0.096   0.113
   8    0.050   0.027   0.024   0.018   0.030   0.034   0.043   0.052   0.063   0.069
  12    0.071   0.036   0.026   0.019   0.023   0.027   0.034   0.044   0.048   0.053
  16    0.055   0.027   0.032   0.020   0.022   0.024   0.032   0.033   0.038   0.042
  24    0.060   0.061   0.046   0.032   0.033   0.030   0.035   0.040   0.048   0.049
  32    0.112   0.071   0.075   0.059   0.065   0.043   0.052   0.054   0.064   0.060
  40    0.209   0.160   0.103   0.069   0.062   0.084   0.048   0.076   0.057   0.067
  48    0.265   0.190   0.130   0.123   0.084   0.100   0.066   0.075   0.081   0.089
[       OK ] testQ2.radixSortTest (8324 ms)
[----------] 8 tests from testQ2 (9360 ms total)

[----------] Global test environment tear-down
[==========] 8 tests from 1 test suite ran. (9361 ms total)
[  PASSED  ] 8 tests.
```

Threads/Blocks	1	2	4	8	12	16	24	32	40	48
1	0.042	0.037	0.037	0.048	0.073	0.099	0.192	0.301	0.347	0.384
2	0.037	0.02	0.021	0.025	0.043	0.056	0.098	0.15	0.175	0.205
4	0.049	0.027	**0.017**	0.018	0.03	0.033	0.059	0.09	0.096	0.113
8	0.05	0.027	0.024	0.018	0.03	0.034	0.043	0.052	0.063	0.069

12	0.071	0.036	0.026	0.019	0.023	0.027	0.034	0.044	0.048	0.053
16	0.055	0.027	0.032	0.02	0.022	0.024	0.032	0.033	0.038	0.042
24	0.06	0.061	0.046	0.032	0.033	0.03	0.035	0.04	0.048	0.049
32	0.112	0.071	0.075	0.059	0.065	0.043	0.052	0.054	0.064	0.06
40	0.209	0.16	0.103	0.069	0.062	0.084	0.048	0.076	0.057	0.067
48	0.265	0.19	0.13	0.123	0.084	0.1	0.066	0.075	0.081	0.089

The fastest time recorded is 0.017 seconds for 4 threads and 4 blocks.

Other very low times are around this configuration, such as 4 threads with 8 blocks and 12 threads with 8 blocks.

Increasing Threads: Increasing the number of threads generally improves the performance up to a certain point. Beyond this point, the performance gains diminish, and sometimes the time increases due to overhead from thread management.

Increasing Blocks: Increasing the number of blocks generally improves performance up to a certain point. Beyond this, the performance gains are less pronounced or may even decrease.