Question 1: Comparing palyndrome.py with slow-pali.cpp

a) _

Output for measuring the time of slow-pali < t3.txt:

muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ time ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___

real 0m0.007s

user 0m0.004s

sys 0m0.003s

Output for measuring the time of slow-pali < t4.txt:

muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ time ./slow-pali < t4.txt
Longest palindrome: redder

real 0m3.935s

user 0m1.695s

sys 0m2.231s

Output for measuring the time of palindrome < t3.txt:

muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ time ./palindrome.py < t3.txt Longest palindrome: ___o.O.o___

real 0m0.025s

user 0m0.014s

sys 0m0.007s

Output for measuring the time of palindrome < t4.txt:

muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ time ./palindrome.py < t4.txt Longest palindrome: redder

real 0m0.194s

user 0m0.187s

sys 0m0.005s

b) The C++ program took 0.004 seconds to open t3.txt & 1.695 seconds to open t4.txt.

- The C++ program took 0.003 seconds waiting for IO to finish on t3.txt & 2.231 seconds waiting for IO to finish on t4.txt.
- The Python program took 0.014 seconds to open t3.txt & 0.187 seconds to open t4.txt.
- The Python program took 0.007 seconds waiting for IO to finish on t3.txt and 0.005 seconds waiting for IO to finish on t4.txt.

c) _

Output for running strace -c on palindrome.py on t3.txt

```
muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ strace -c ./palindrome.py < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 24.15    0.000364           1       252        38 newfstatat
 20.37    0.000307          19        16           getdents64
 12.87    0.000194           2        84        18 openat
  9.75    0.000147          29         5         3 execve
  6.77    0.000102           2        46           mmap
  6.44    0.000097           1        80           read
  5.97    0.000090           1        69           close
  2.99    0.000045           0        66           rt_sigaction
  2.65    0.000040           0        70         2 lseek
  1.92    0.000029           0        45        40 ioctl
  1.13    0.000017           2         8           mprotect
  1.13    0.000017           3         5           munmap
  1.06    0.000016           1        12           brk
  0.60    0.000009           1         9           pread64
  0.46    0.000007           1         4         3 readlink
  0.27    0.000004           2         2         2 access
  0.27    0.000004           2         2           getcwd
  0.27    0.000004           1         3           getrandom
  0.13    0.000002           2         1           write
  0.13    0.000002           0         4         2 arch_prctl
  0.13    0.000002           1         2           set_tid_address
  0.13    0.000002           1         2           set_robust_list
  0.13    0.000002           1         2           prlimit64
  0.07    0.000001           1         1           fcntl
  0.07    0.000001           1         1           getgid
  0.07    0.000001           0         2           futex
  0.07    0.000001           0         2           rseq
  0.00    0.000000           0         3           dup
  0.00    0.000000           0         1           sysinfo
  0.00    0.000000           0         1           getuid
  0.00    0.000000           0         1           geteuid
  0.00    0.000000           0         1           getegid
------ ----------- ----------- --------- --------- ----------------
100.00    0.001507           1       802       108 total
```

Output for running strace -c on palindrome.py on t4.txt

```
muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ strace -c ./palindrome.py < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 29.82    0.000164           0       252        38 newfstatat
 20.00    0.000110           6        16           getdents64
 14.91    0.000082           0        85           read
  8.55    0.000047           0        84        18 openat
  6.91    0.000038           0        70         2 lseek
  6.00    0.000033           0        69           close
  5.45    0.000030           0        45        40 ioctl
  4.00    0.000022           0        66           rt_sigaction
  1.64    0.000009           0        12           brk
  1.45    0.000008           0        46           mmap
  1.27    0.000007           2         3           dup
  0.00    0.000000           0         1           write
  0.00    0.000000           0         8           mprotect
  0.00    0.000000           0         5           munmap
  0.00    0.000000           0         9           pread64
  0.00    0.000000           0         2         2 access
  0.00    0.000000           0         5         3 execve
  0.00    0.000000           0         1           fcntl
  0.00    0.000000           0         2           getcwd
  0.00    0.000000           0         4         3 readlink
  0.00    0.000000           0         1           sysinfo
  0.00    0.000000           0         1           getuid
  0.00    0.000000           0         1           getgid
  0.00    0.000000           0         1           geteuid
  0.00    0.000000           0         1           getegid
  0.00    0.000000           0         4         2 arch_prctl
  0.00    0.000000           0         2           futex
  0.00    0.000000           0         2           set_tid_address
  0.00    0.000000           0         2           set_robust_list
  0.00    0.000000           0         2           prlimit64
  0.00    0.000000           0         3           getrandom
  0.00    0.000000           0         2           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.000550           0       807       108 total
```

Output for running strace -c on slow-pali.cpp on t3.txt

```
muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ strace -c ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0        43           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0         5           close
  0.00    0.000000           0        23           mmap
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         6           newfstatat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000           0       108         2 total
```

Output for running strace -c on slow-pali.cpp on t4.txt

```
muhammad.rizwan2@linux02-ed:~/cpsc457/tutorial/palindrome$ strace -c ./slow-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00    8.815231           1   5767198           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0         5           close
  0.00    0.000000           0        23           mmap
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         6           newfstatat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    8.815231           1   5767263         2 total
```

d) Slow-pali.cpp took a longer time to open t4.txt than Palindrome.py

However, Palindrome.py took a longer time opening t3.txt than Slow-pali.cpp.

This may be because of the number of system calls both C++ and Python files make when opening each file. Slow-pali.cpp took 5,767,263 calls in t4.txt which is a lot more than Palindrome.py which may be the reason why C++ took longer with t4.txt than Python.

Similarly, Palindrome.py made 802 calls in t3.txt which is a lot more than slow-pali.cpp.

Therefore, the reason is due to the **number of system calls made** by the Python and C++ files when opening each text file.

Question 3: Comparing my fast-pali.cpp with slow-pali.cpp

a) **Output for measuring the time of fast-pali < t3.txt and t4.txt:**

```
muhammad.rizwan2@linux13-wb:~/ncpsc457/palindrome$ time ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___

real    0m0.005s
user    0m0.001s
sys     0m0.002s
muhammad.rizwan2@linux13-wb:~/ncpsc457/palindrome$ time ./fast-pali < t4.txt
Longest palindrome: redder

real    0m0.157s
user    0m0.076s
sys     0m0.036s
```

Output for running strace -c on fast-pali.cpp on t3.txt

```
muhammad.rizwan2@linux13-wb:~/ncpsc457/palindrome$ strace -c ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         6           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0         5           close
  0.00    0.000000           0        23           mmap
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         6           newfstatat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000           0        71         2 total
```

Output for running strace -c on fast-pali.cpp on t4.txt

```
muhammad.rizwan2@linux13-wb:~/ncpsc457/palindrome$ strace -c ./fast-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 69.12    0.003440          81        42           brk
 23.75    0.001182         107        11           read
  6.51    0.000324          32        10           munmap
  0.56    0.000028           0        32           mmap
  0.04    0.000002           2         1           write
  0.02    0.000001           0         6           newfstatat
  0.00    0.000000           0         5           close
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.004977          37       133         2 total
```

b) My fast-pali is faster than slow-pali given because the program now has a modified buffer than takes 1MB (1024*1024) of bytes per system call rather than one-by-one which dramatically reduced the number of system calls read makes when reading a file.

	fast-pali.cpp	**slow-pali.cpp**
	t3.txt	
Time (Real - User Time)	0.003s	0.003s
strace -c Total Calls	71 calls	108 calls
	t4.txt	
Time (Real - User Time)	0.081s	2.24s
strace -c Total Calls	133 calls	5767263 calls

c) My fast-pali is also faster than palindrome.py because after reducing the number of *read* system calls. When comparing the *time* and *strace -c* on palindrome.py and fast-pali.cpp:

	fast-pali.cpp	**palindrome.py**
	t3.txt	
Time (Real - User Time)	0.003s	0.011s
strace -c Total Calls	71 calls	802 calls
	t4.txt	
Time (Real - User Time)	0.081s	0.007s
strace -c Total Calls	133 calls	807 calls