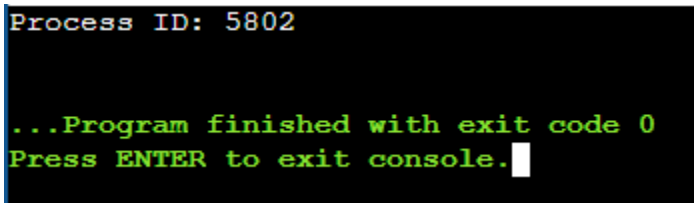
No E-Mail submissions will be accepted.
Submission formats and file naming:
File name : firstName_lastName_lab_1
File format: pdf or MS Word format
e.g. Jim_Carrey_lab_1.pdf

1. Consider the following C code (*main.c*) and answer/complete the following questions/tasks:

```c
#include <stdio.h>
#include <unistd.h>

int main()
{
    printf(" Preocess ID : %d \n", getpid());
    sleep(1);
    return 0;
}
```

A) Use the online C compiler https://www.onlinegdb.com/online_c_compiler and run your code. Attach a screenshot of your output.

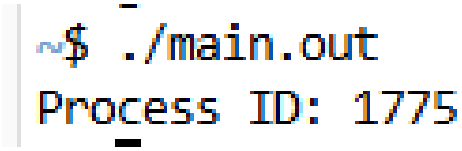


B) Use your Linux environment and run your code. Attach a screenshot of your output.

Hint:

Compilation

```
gcc main.c -o main.out
./main.out
```



C) In your code change the sleeping time from **1s** to **30s**, recompile your code and in the Linux terminal run the following commands (in the same terminal):

I) *./main.out &*
(Enter the command and then press the enter key two times)

```
~$ ./main.out &
[2] 1836
~$ Process ID: 1836
```

II) *pstree -p -T* (or *pstree -pT*)

and answer following questions:

C.1) Did you get the same process ID for *main.out* in both part I and II (Yes/No)? Attach a screenshot of your output.

```
~$ pstree -p -T
tini(1)──sh(6)──node(7)─┬─bash(1629)
                        ├─bash(1705)─┬─main.out(1826)
                        │            └─pstree(1846)
                        └─sshd(282)
[2]-  Done                    ./main.out
```

No

C.2) What does *"pstree -p -T"* mean?
Hint: Use *"man pstree"* command to answer this question.
pstree shows running processes as a tree. The tree is rooted at either pid or init if pid is omitted. If a user name is specified, all process trees rooted at processes owned by that user are shown.
pstree visually merges identical branches by putting them in square brackets and prefixing them with the repetition count, e.g.

C.3) What does *./main.out &* mean?
./main.out runs the program named main.out in the current directory
& tells it to run in the background

D) Make sure the sleeping time is 30s and then open two Linux terminals, in the first terminal run *./main.out* and in the second terminal run *pstree -pT*. Are you still getting the same process ID for *./main.out* in both terminals (Yes/No)? Attach a screenshot of your output.

```
~$ ./main.out
Process ID: 2032
```

```
~$ pstree -pT
tini(1)──sh(6)──node(7)─┬─bash(1629)
                        ├─bash(1705)──main.out(1826)
                        ├─bash(1958)──pstree(2043)
                        └─sshd(282)
```

No

Use your Linux environment and answer the following questions.

2. Run lscpu command and complete the following table:

Architecture	x86_64
CPU op-mode(s)	32-bit, 64-bit
Address sizes	46 bits physical, 46 bits virtual
Byte Order	Little Endian

Attach a screenshot of your output.

```
~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  4
  On-line CPU(s) list:   0-3
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) CPU @ 2.80GHz
    CPU family:          6
    Model:               85
    Thread(s) per core:  2
    Core(s) per socket:  2
    Socket(s):           1
    Stepping:            7
    BogoMIPS:            5599.99
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
                          ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_kn
                         own_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c r
                         drand hypervisor lahf_lm abm 3dnowprefetch invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced fsgs
                         base tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm avx512f avx512dq rdseed adx smap clflush
                         opt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves arat avx512_vnni md_clear arch
                         _capabilities
Virtualization features:
  Hypervisor vendor:     KVM
  Virtualization type:   full
Caches (sum of all):
  L1d:                   64 KiB (2 instances)
  L1i:                   64 KiB (2 instances)
  L2:                    2 MiB (2 instances)
  L3:                    33 MiB (1 instance)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-3
Vulnerabilities:
  Gather data sampling:  Not affected
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
  Meltdown:              Not affected
  Mmio stale data:       Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
  Reg file data sampling: Not affected
  Retbleed:              Mitigation; Enhanced IBRS
  Spec rstack overflow:  Not affected
  Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl and seccomp
  Spectre v1:            Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:            Mitigation; Enhanced / Automatic IBRS; IBPB conditional; RSB filling; PBRSB-eIBRS SW sequence; BH
                         I SW loop, KVM SW loop
  Srbds:                 Not affected
```

3. Run `lscpu | grep cache` command in your terminal and attach a screenshot of your output. What does `| grep cache` mean?

```
~$ lscpu | grep cache
L1d cache:                       64 KiB (2 instances)
L1i cache:                       64 KiB (2 instances)
L2 cache:                        2 MiB (2 instances)
L3 cache:                        33 MiB (1 instance)
```

`|` is a pipe operator to pass lscpu as an argument
grep cache filters lines that contain the word cache