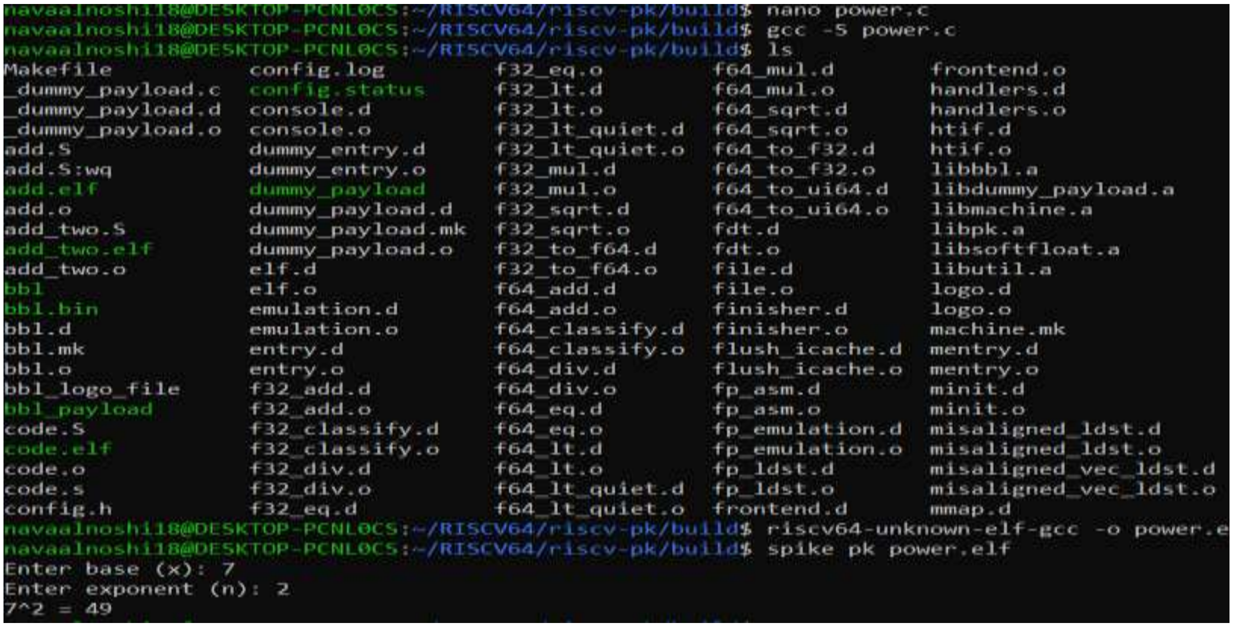
Summary- by Navaal Noshi

Task-4:



```
navaalnoshi18@DESKTOP-PCNL0CS:~/RISCV64/riscv-pk/build$ nano power.c
navaalnoshi18@DESKTOP-PCNL0CS:~/RISCV64/riscv-pk/build$ gcc -S power.c
navaalnoshi18@DESKTOP-PCNL0CS:~/RISCV64/riscv-pk/build$ ls
Makefile            config.log          f32_eq.o            f64_mul.d           frontend.o
_dummy_payload.c    config.status       f32_lt.d            f64_mul.o           handlers.d
_dummy_payload.d    console.d           f32_lt.o            f64_sqrt.d          handlers.o
_dummy_payload.o    console.o           f32_lt_quiet.d      f64_sqrt.o          htif.d
add.S               dummy_entry.d       f32_lt_quiet.o      f64_to_f32.d        htif.o
add.S:wq            dummy_entry.o       f32_mul.d           f64_to_f32.o        libbbl.a
add.elf             dummy_payload       f32_mul.o           f64_to_ui64.d       libdummy_payload.a
add.o               dummy_payload.d     f32_sqrt.d          f64_to_ui64.o       libmachine.a
add_two.S           dummy_payload.mk    f32_sqrt.o          fdt.d               libpk.a
add_two.elf         dummy_payload.o     f32_to_f64.d        fdt.o               libsoftfloat.a
add_two.o           elf.d               f32_to_f64.o        file.d              libutil.a
bbl                 elf.o               f64_add.d           file.o              logo.d
bbl.bin             emulation.d         f64_add.o           finisher.d          logo.o
bbl.d               emulation.o         f64_classify.d      finisher.o          machine.mk
bbl.mk              entry.d             f64_classify.o      flush_icache.d      mentry.d
bbl.o               entry.o             f64_div.d           flush_icache.o      mentry.o
bbl_logo_file       f32_add.d           f64_div.o           fp_asm.d            minit.d
bbl_payload         f32_add.o           f64_eq.d            fp_asm.o            minit.o
code.S              f32_classify.d      f64_eq.o            fp_emulation.d      misaligned_ldst.d
code.elf            f32_classify.o      f64_lt.d            fp_emulation.o      misaligned_ldst.o
code.o              f32_div.d           f64_lt.o            fp_ldst.d           misaligned_vec_ldst.d
code.s              f32_div.o           f64_lt_quiet.d      fp_ldst.o           misaligned_vec_ldst.o
config.h            f32_eq.d            f64_lt_quiet.o      frontend.d          mmap.d
navaalnoshi18@DESKTOP-PCNL0CS:~/RISCV64/riscv-pk/build$ riscv64-unknown-elf-gcc -o power.e
navaalnoshi18@DESKTOP-PCNL0CS:~/RISCV64/riscv-pk/build$ spike pk power.elf
Enter base (x): 7
Enter exponent (n): 2
7^2 = 49
```

In this task, I implemented a power(x, n) function in C to calculate x^n using a simple loop that multiplies x a total of n times. After writing the function in a file named power.c , I used the RISC-V cross-compiler (riscv64-unknown-elf-gcc) to convert the C code into RISC-V assembly by running **the -S flag**, which generated the power.s file. This assembly file was then assembled using riscv64-unknown-elf-as to produce an object file (power.o) and linked using **riscv64-unknown-elf-gcc** to create an executable ELF binary (power.elf). I executed this binary on the Spike RISC-V simulator with the pk proxy kernel. To analyze performance, I ran Spike with the -l flag to log all executed instructions to log.txt, but unfortunately it was not working with my spike version.