`This document describes the solution of this exercise!

1) Using readelf

readelf -h -W CrackMe |less // ELF header

Entry point address: 0x400650

Class: ELF64

readelf -l -W CrackMe |less // PHT = Program Header Table (Execution view)

.text section is on 2 segment, and it is a readable( R ) and executable (E)

02 .interp .note.ABI-tag .note.gnu.build-id .gnu.hash .dynsym .dynstr .gnu.version .gnu.version\_r .rela.dyn .rela.plt .init .plt .plt.got .text .fini .rodata .eh\_frame\_hdr .eh\_frame

LOAD 0x000000 0x0000000000400000 0x0000000000400000 0x000c34 0x000c34 R E 0x200000

readelf -S -W CrackMe |less // SHT = Section Header Table (Linking view)

REL = relocations

SYMTAB/DYNSIM = symbol tables

readelf -r -W CrackMe |less // Relocations

readelf -d -W CrackMe |less // .dynamic section

NEEDED = which libraries must be loaded before the program runs

Dynamic linker loads ( dynamic linker is provided in the program header .interp) the dynamic libraries and control is transferred to the \_\_start in the binary.