Question 1:
Performing the following tasks will help you become more familiar with the various tools for manipulating object files.

A. How many object files are contained in the versions of **`libc.a`** and **`libm.a`** on your system?

   Libc.a contains 1671 and libm.a contains 804

B. Does **`gcc -Og`** produce different executable code than **`gcc -Og -g`**?

   No, the differnce is debugging.

C. What shared libraries does the **`gcc`** driver on your system use?

   linux-vdso.so.1
   libm.so.6 => /lib64/libm.so.6
   libc.so.6 => /lib64/libc.so.6
   ld-linux-x86-64.so.2

Question 2:
This problem concerns the following **`m.o`** module:

```
// m.c
void swap();
int buf[2] = {1,2};
int main(){
        swap();
      return 0;
}
```

The following version of the **`swap.c`** function that counts the number of times it has been called:

```
extern int buf[];
int *bufp0 = &buf[0];
static int *bufp1;
static void incr(){
        static int count = 0;
        count++;
}
void swap(){
        int temp;
        incr();
        bufp1 = &buf[1];
        temp = *bufp0;
        *bufp0 = *bufp1;
        *bufp1 = temp;
}
```

For each symbol that is defined and referenced in **`swap.o`**, indicate if it will have a symbol table entry in the **`.symtab`** section in module **`swap.o`**. If so, indicate the module that defines the symbol (**`swap.o`** or **`m.o`**), the symbol type (**local**, **global**, or **extern**), and the section (**`.text`**, **`.data`**, or **`.bss`**) it occupies in that module.

Symbol	swap.o .symtab entry	Symbol type	Module where defined	Section
`buf`	Yes	extern	m.o	.data
`bufp0`	Yes	global	swap.o	.data
`bufp1`	Yes	local	swap.o	.bss
`swap`	Yes	global	swap.o	.text
`temp`	No	–	–	–
`incr`	Yes	local	swap.o	.text
`count`	Yes	local	swap.o	.bss

Question 3:

Consider the call to function **swap** in object file **m.o** (question 2).

```
b:     e8     00     00     00     00     callq e <main+0xe)  swap()
```

With the following relocation entry:

```
r.offset =  0xc
r.symbol = swap
r.type = R_X86_64_PC32
r.addend = -4
```

A. Suppose that the linker relocates **.text** in **m.o** to address `0x4004e0` and **swap** to address `0x4004f4`. Then what is the value of the relocated reference to **swap** in the **callq** instruction?

   The value will be 0x4004f4

B. Suppose that the linker relocates **.text** in **m.o** to address `0x4004d0` and **swap** to address `0x400500`. Then what is the value of the relocated reference to **swap** in the **callq** instruction?

   Value will be 0x400500