班号 2211104 学号： 2022211897 姓名：韩易峰

考虑如下所示的关系数据库模式，主码加了下划线，使用关系代数表达式表示查询。

Employee(person\_name, street, city)

Works(person\_name, company\_name, salary) ［５］ 。 然

Company(company\_name, city)

Manages(person\_name, manager\_name)

1. 找出“Baidu Corporation”公司中所有薪水超过100万的员工信息。

σ Works.company\_name='Baidu Corporation' and Works.salary > 1000000 (Works)

2. 找出所有担任经理职务的员工的名字。

π person\_name (Manages)

3. 找出所有被“Hans Write”管理的员工的名字。

π person\_name (Manages ⨝ σ manager\_name='Hans Write' (Manages))

4. 列出各个公司员工薪水的总数。

π company\_name, SUM(salary) (Works ⨝ Company)

5. 找出居住在威海市“Wenhua West Road”的员工所有在的所有公司的名字。

π person\_name, company\_name (Employee ⨝ Works ⨝ Company | city='威海市' and street='Wenhua West Road')

6. 找出与其经理居住在同一城市同一街道的所有员工的姓名。

π e1.person\_name (Employee × Manages × Employee as e1 | e1.manager\_name = Manages.manager\_name and e1.city = Employee.city and e1.street = Employee.street)

7. 找出不在“Baidu Corporation”工作的所有员工的姓名。

π person\_name (Employee - Works ⨝ π person\_name, company\_name (Works ⨝ σ company\_name='Baidu Corporation' (Company)))

8. 找出比“Sohu Corporation”所有员工收入都高的所有员工的姓名。

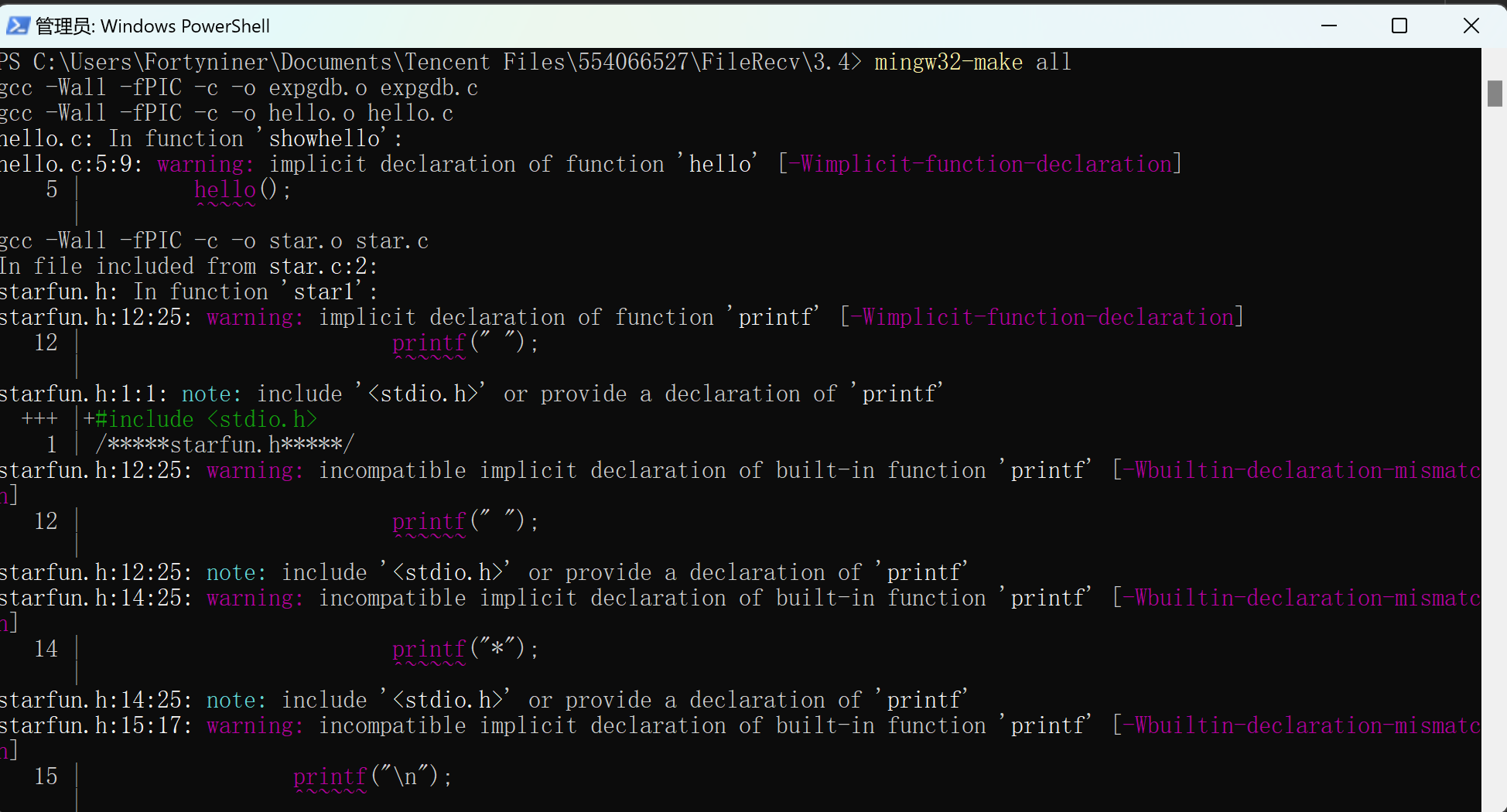
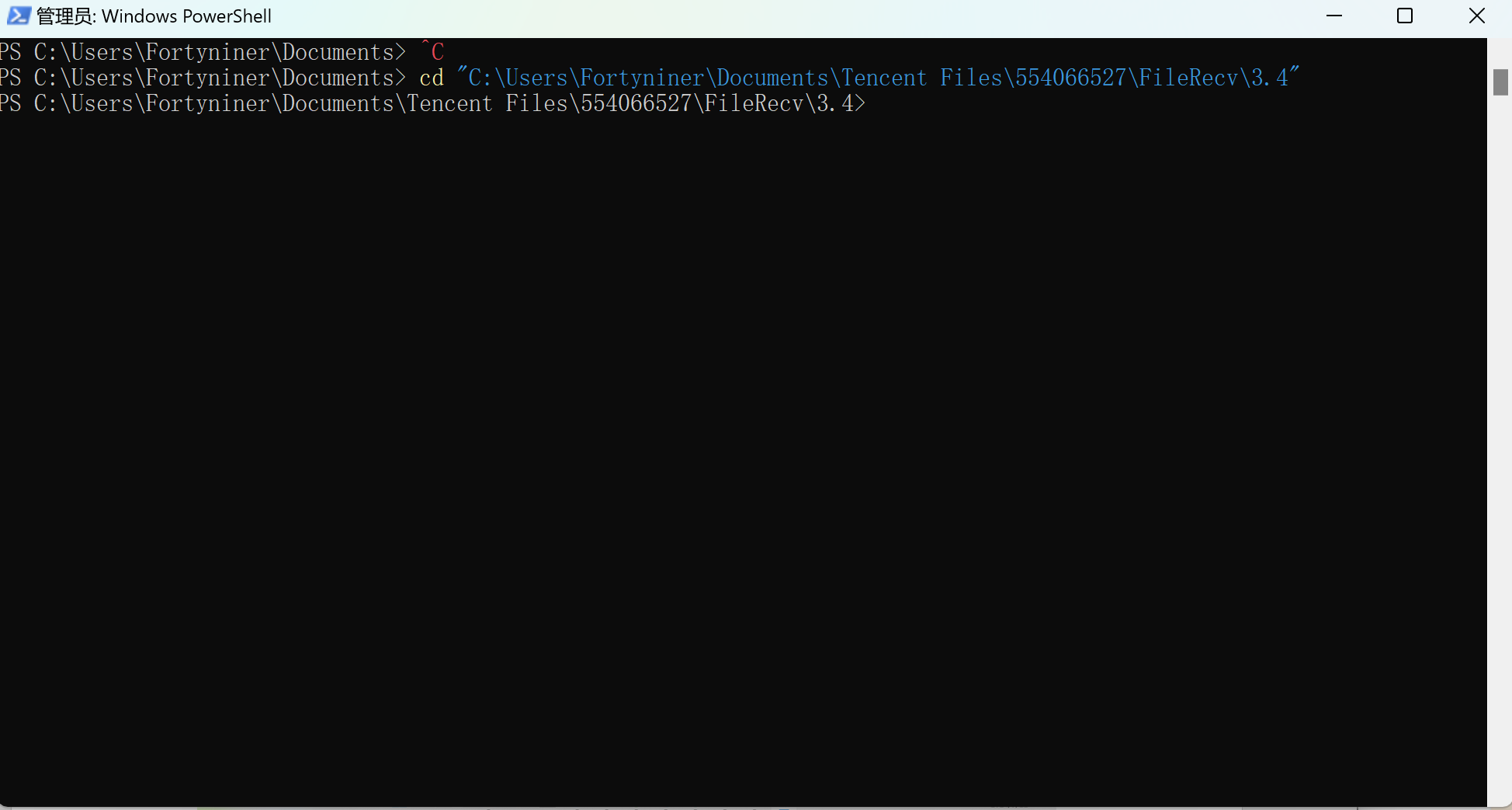
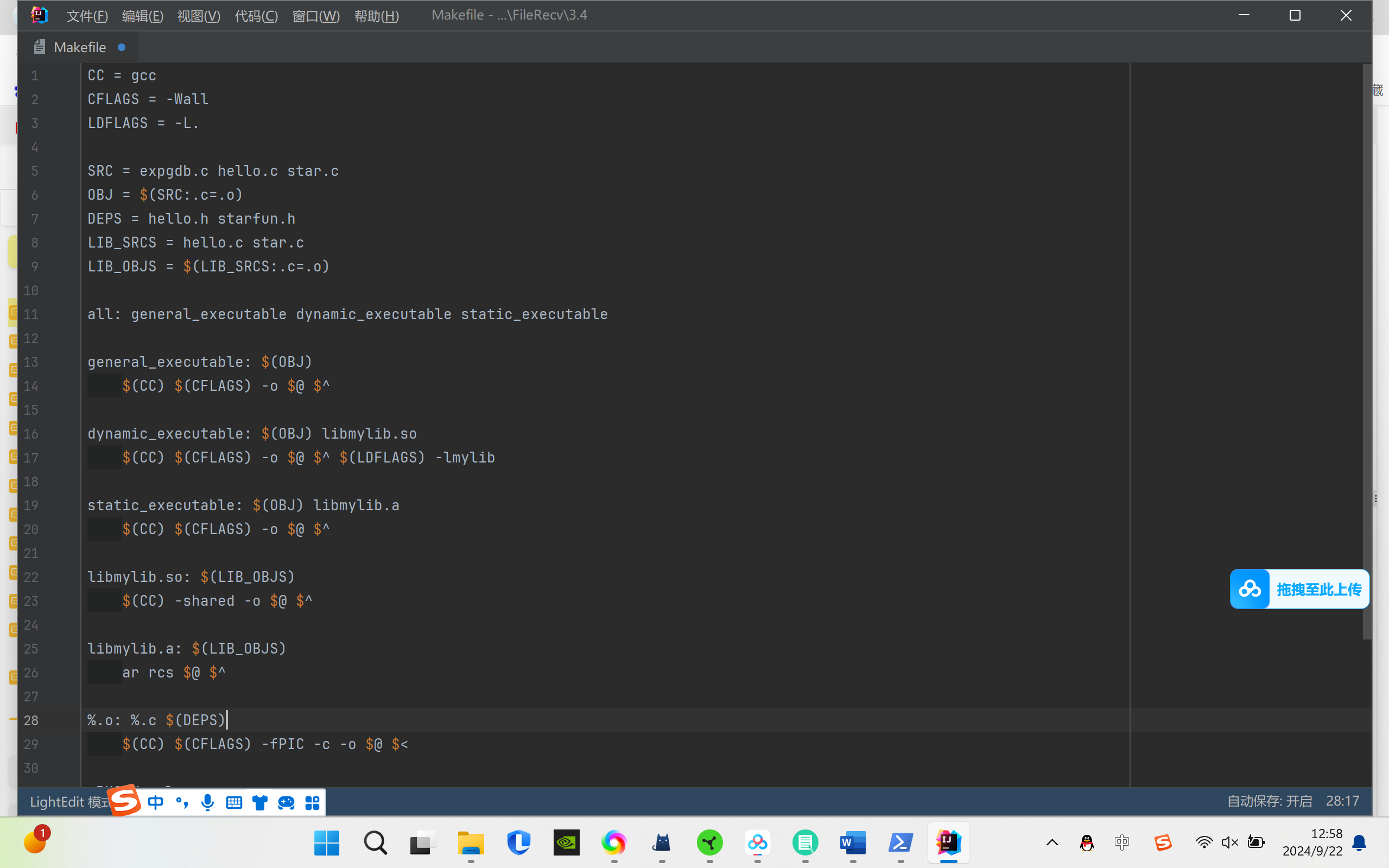
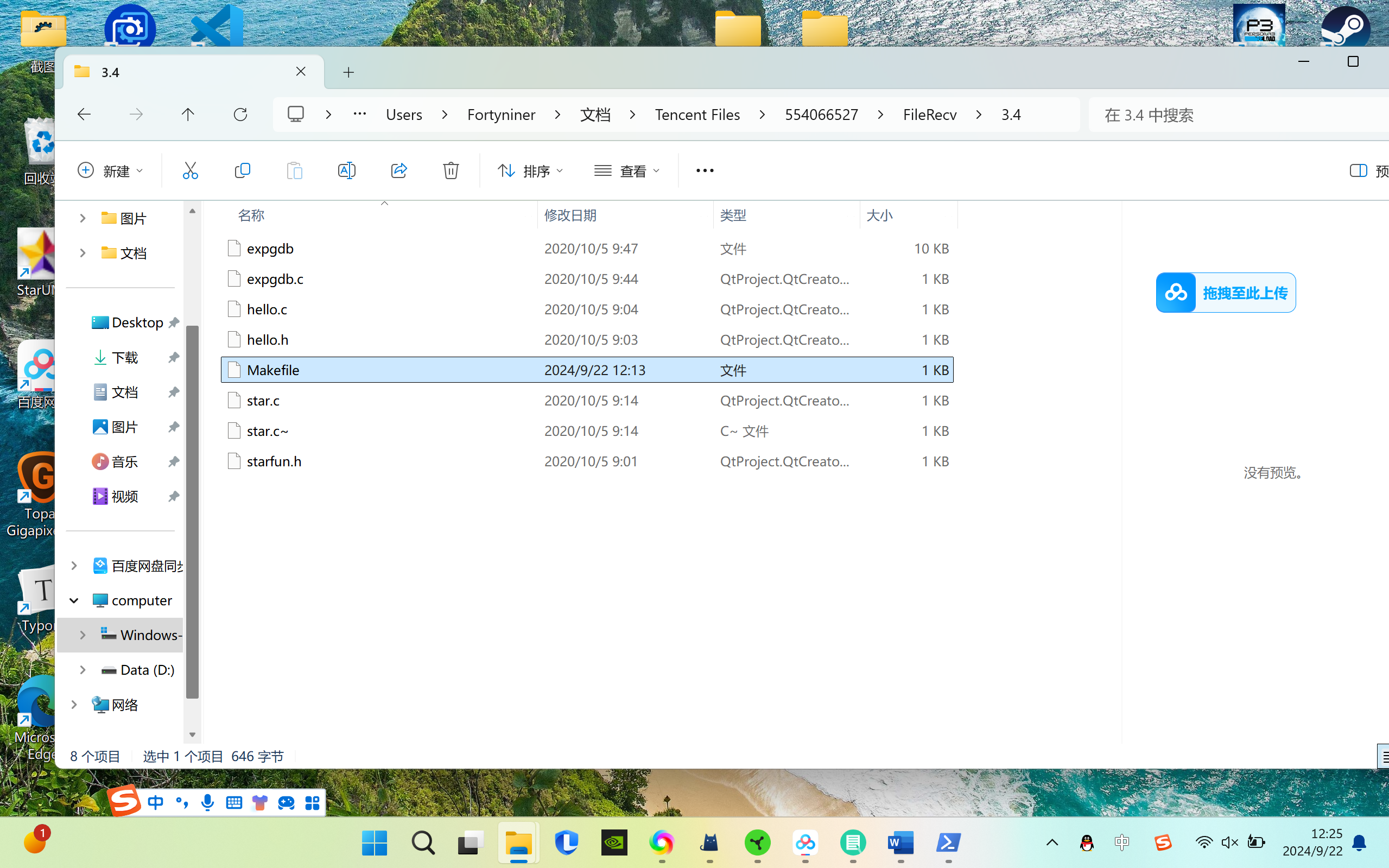
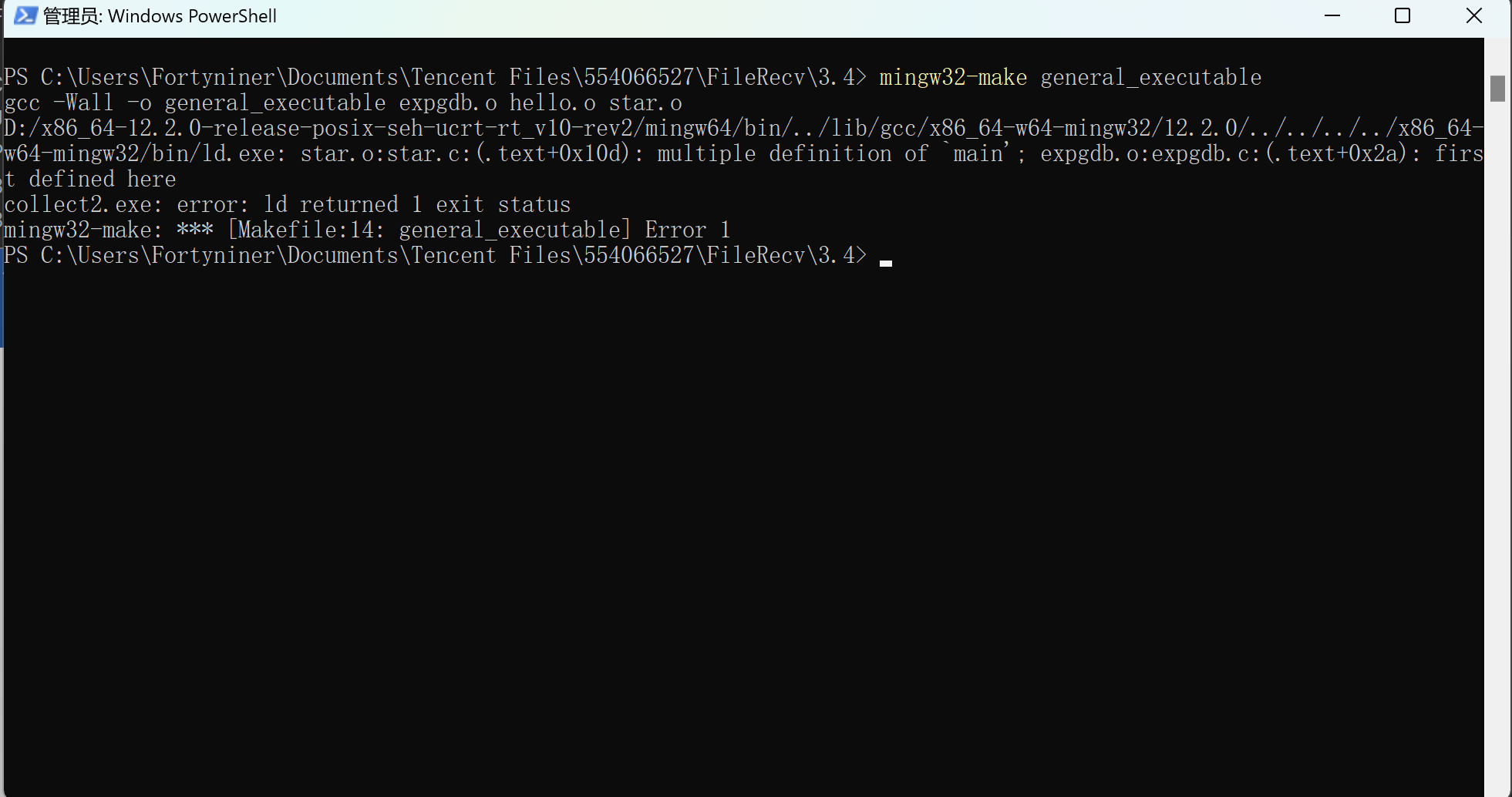
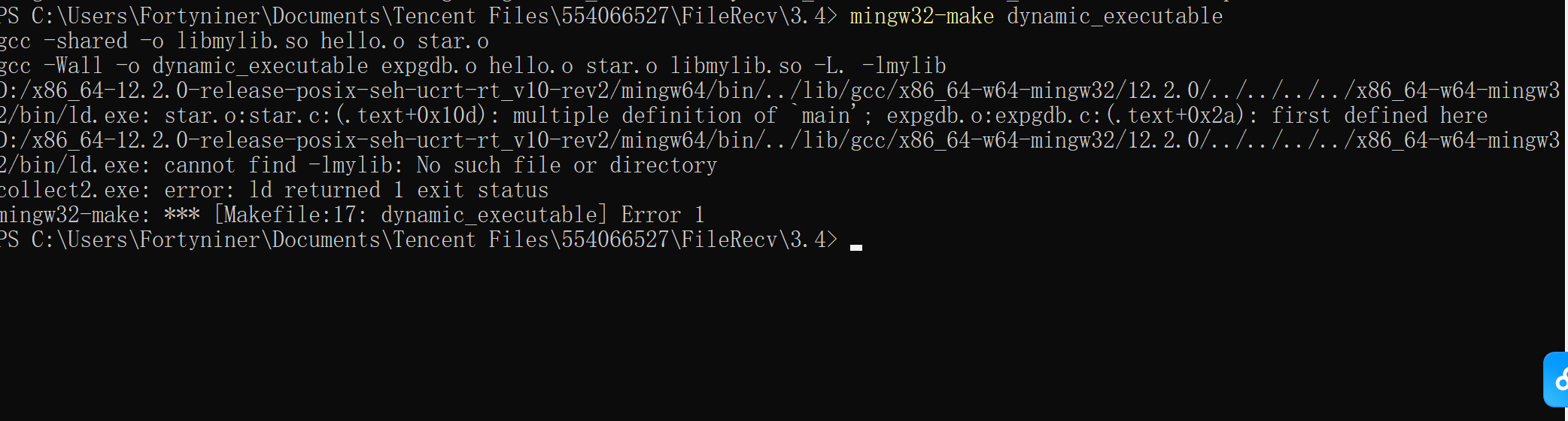
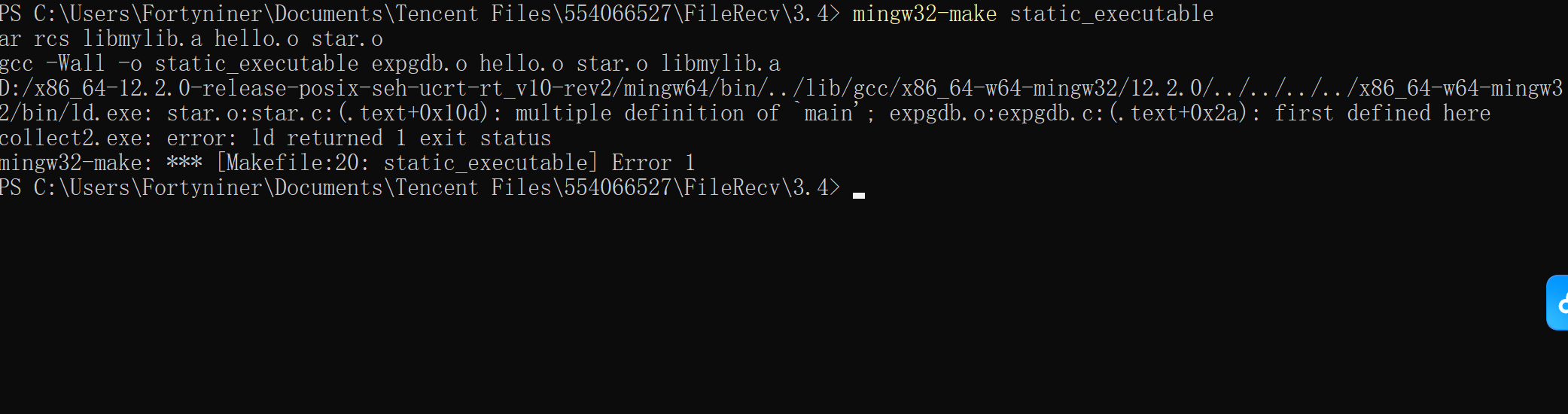
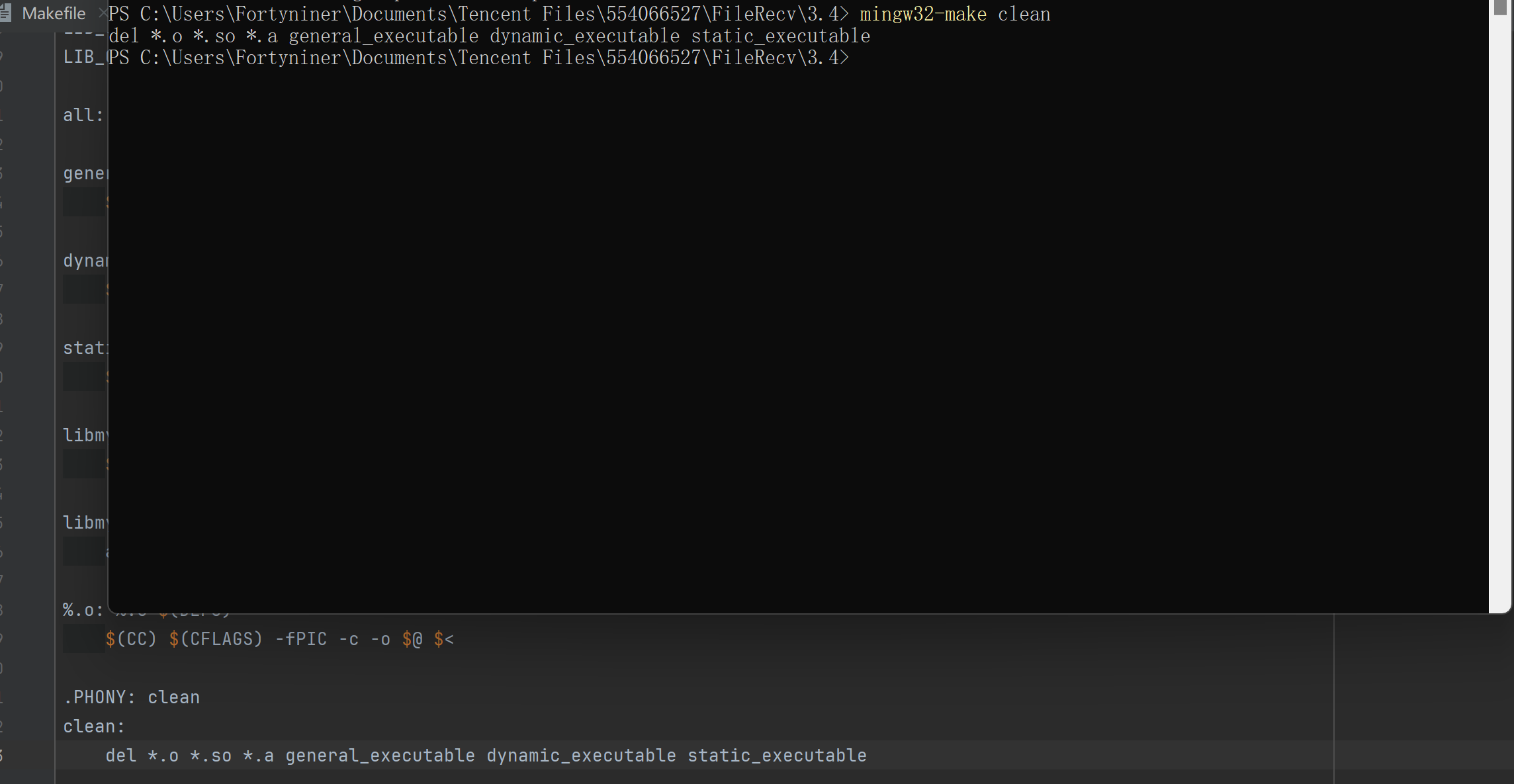
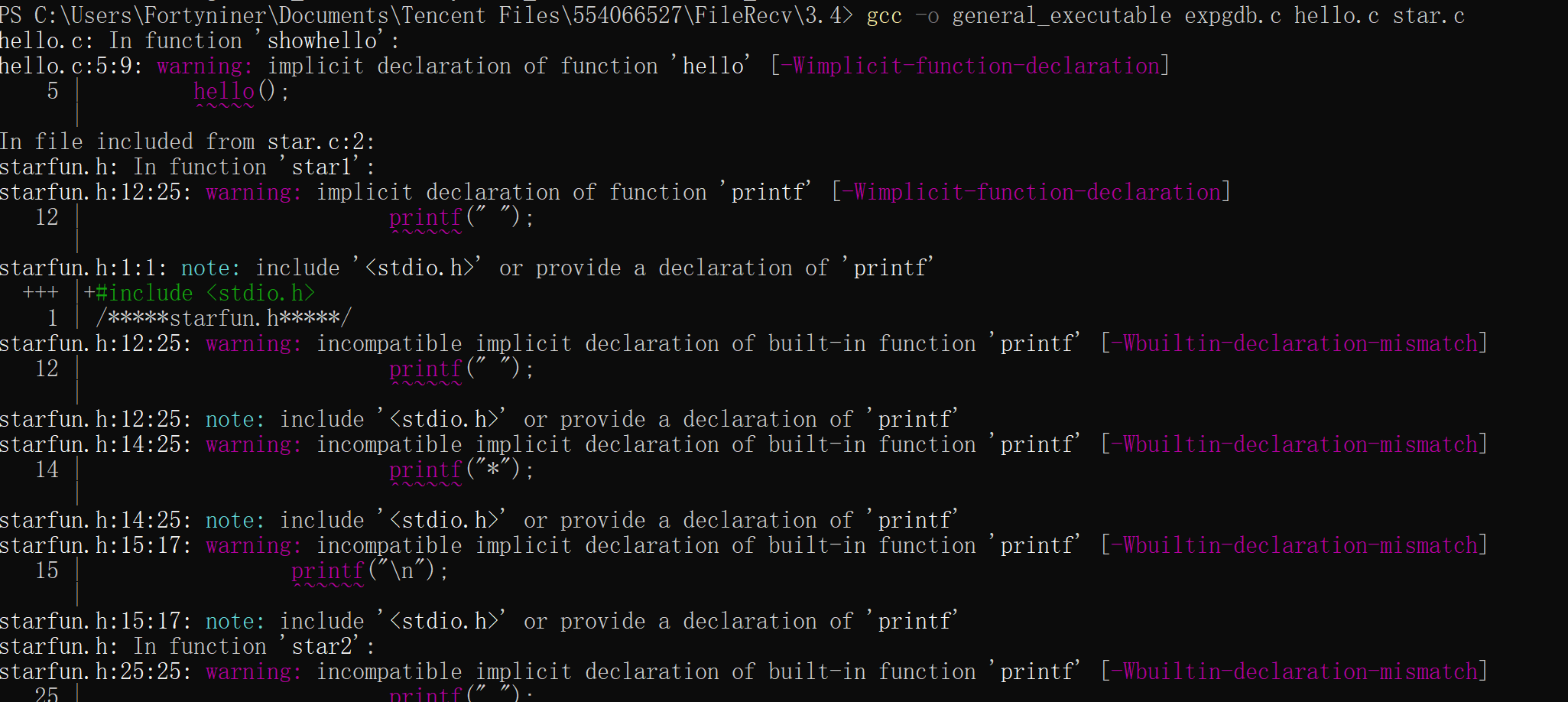
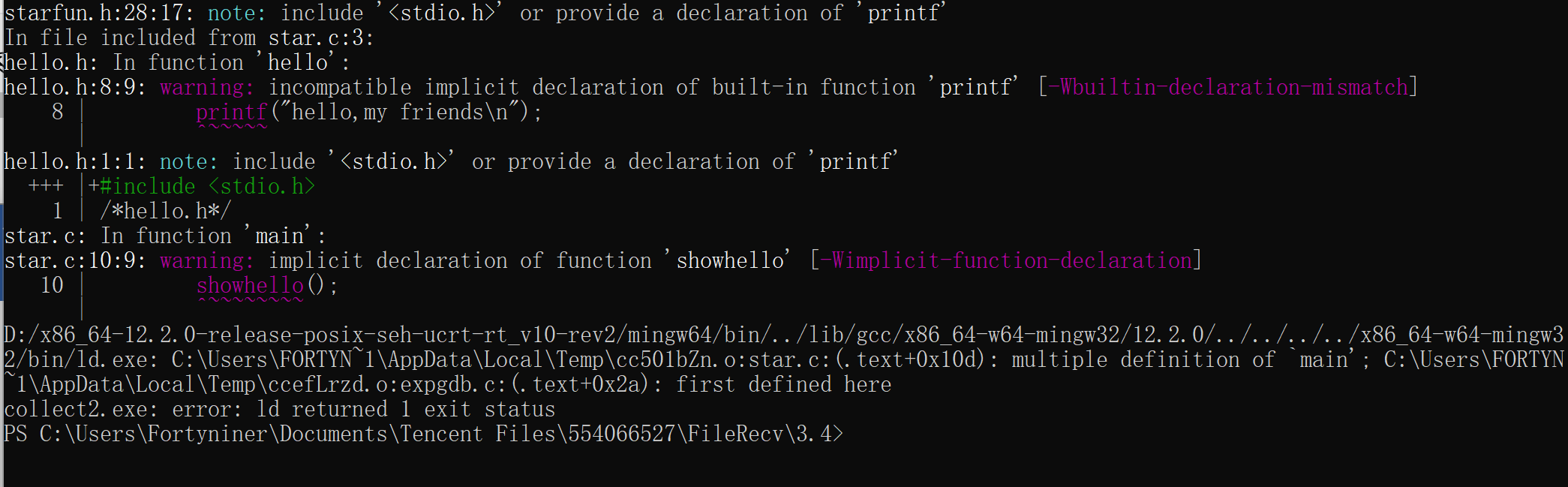
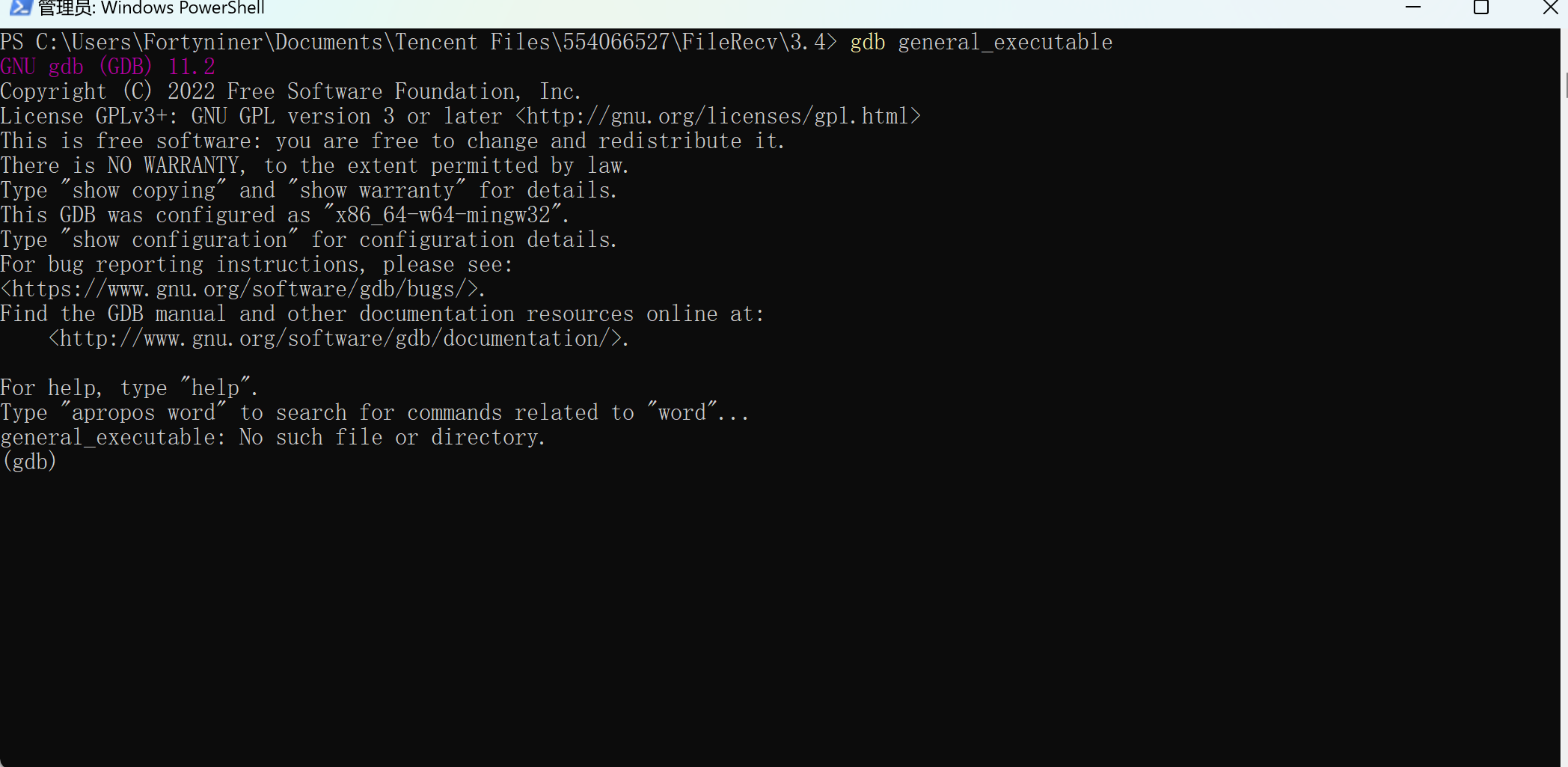
π person\_name (Works ⨝ (σ company\_name='Sohu Corporation' (Works)) - Works)

9. 找出“Sohu Corporation”公司中薪水最高的员工的名字。

π person\_name (σ salary = (max(Works.salary) ⨝ σ company\_name='Sohu Corporation' (Works)))

10. 找出“Sohu Corporation”公司所有员工薪水中的第二大的值。

π salary (σ salary < (max(Works.salary) ⨝ σ company\_name='Sohu Corporation' (Works)), Works)

请将上述Makefile内容保存为名为 Makefile 的文件，并确保文件名与您的源代码文件共处在同一目录下。然后，您可以在命令行中执行如下命令来编译生成各种类型的可执行文件：

* make all: 生成所有类型的可执行文件。
* make general\_executable: 生成一般可执行文件。
* make dynamic\_executable: 生成基于动态库的可执行文件。
* make static\_executable: 生成基于静态库的可执行文件。

此外，您可以使用make clean命令来清除生成的目标文件和可执行文件。

如果您在执行过程中遇到任何问题，或者需要进一步的帮助，请随时告诉我。我将尽力协助您解决相关问题。祝您顺利完成任务！

mingw32-make general\_executable

mingw32-make dynamic\_executable

mingw32-make clean

cd "C:\Users\Fortyniner\Documents\Tencent Files\554066527\FileRecv\3.4"

当您完成 3.4 部分的程序后，可以按照以下步骤来生成三种不同方式的可执行文件并进行调试：

**生成不使用库的可执行文件：**

1. 使用命令 gcc -o general\_executable expgdb.c hello.c star.c 编译生成生成可执行文件 general\_executable。
2. 运行 ./general\_executable 来执行程序。
3. 使用 gdb general\_executable 命令启动 gdb 调试器进行调试，设置断点、单步执行、查看变量值等。

**生成使用静态库的可执行文件：**

1. 生成静态库文件 libstar.a:
   * 使用命令 gcc -c star.c 编译生成 star.o
   * 使用命令 ar rcs libstar.a star.o 生成静态库文件 libstar.a
2. 使用命令 gcc -o static\_executable expgdb.c hello.c -L. -l:libstar.a 编译链接生成静态库的可执行文件 static\_executable。
3. 运行 ./static\_executable 来执行程序。
4. 使用 gdb static\_executable 命令启动 gdb 调试器进行调试。

**生成使用共享库的可执行文件：**

1. 生成共享库文件 libstar.so:
   * 使用命令 gcc -fPIC -shared -o libstar.so star.c 生成共享库文件 libstar.so。
2. 使用命令 gcc -o dynamic\_executable expgdb.c hello.c -L. -l:star 编译链接生成共享库的可执行文件 dynamic\_executable。
3. 运行 ./dynamic\_executable 来执行程序。
4. 使用 gdb dynamic\_executable 命令启动 gdb 调试器进行调试。

完成以上步骤后，您就可以分别运行三种不同方式生成的可执行文件，并使用 gdb 进行调试。调试完成后，您可以使用截图软件截取相关界面，并上传用于展示实际操作和结果。如果您需要更详细的帮助，请随时提出。祝您顺利完成任务！

PS C:\WINDOWS\system32> cd "C:\Users\Fortyniner\Documents\Tencent Files\554066527\FileRecv\3.4"

PS C:\Users\Fortyniner\Documents\Tencent Files\554066527\FileRecv\3.4>