Reverse Lab1 Report

Task 1

编写一个输出"helloworld"的程序，以下为gcc+llvm编译所得，可以看到，程序能够正确运行. 输入的指令：

```
llvm-mc -filetype=obj hello.s -o hello_gcc.o
gcc hello_gcc.o -o hello_gcc
```

```
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# gcc -S hello.c -o hello.s
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# llvm-mc -filetype=obj hello_gcc.s -o hello_gcc.o
llvm-mc: error: hello_gcc.s: No such file or directory
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# llvm-mc -filetype=obj hello.s -o hello_gcc.o
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# gcc hello_gcc.o -o hello_gcc
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# hello_gcc
hello_gcc: command not found
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# ./hello_gcc
Hello World
```

以下为clang+as的命令及运行截图： `clang -S hello.c -o hello_clang.s`

```
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# clang -S hello.c -o hello_clang.s
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/rev_lab1/program# as hello_clang.s -o hello_clang.o
hello_clang.s: Assembler messages:
hello_clang.s:36: Error: unknown pseudo-op: `.addrsig'
hello_clang.s:37: Error: unknown pseudo-op: `.addrsig_sym'
```

失败原因：

因为 clang 生成的汇编代码与 GNU as 的期望格式存在差异，即clang 默认生成的汇编代码可能包含一些 GNU as 无法识别或处理的指令或伪指令。

Task 2

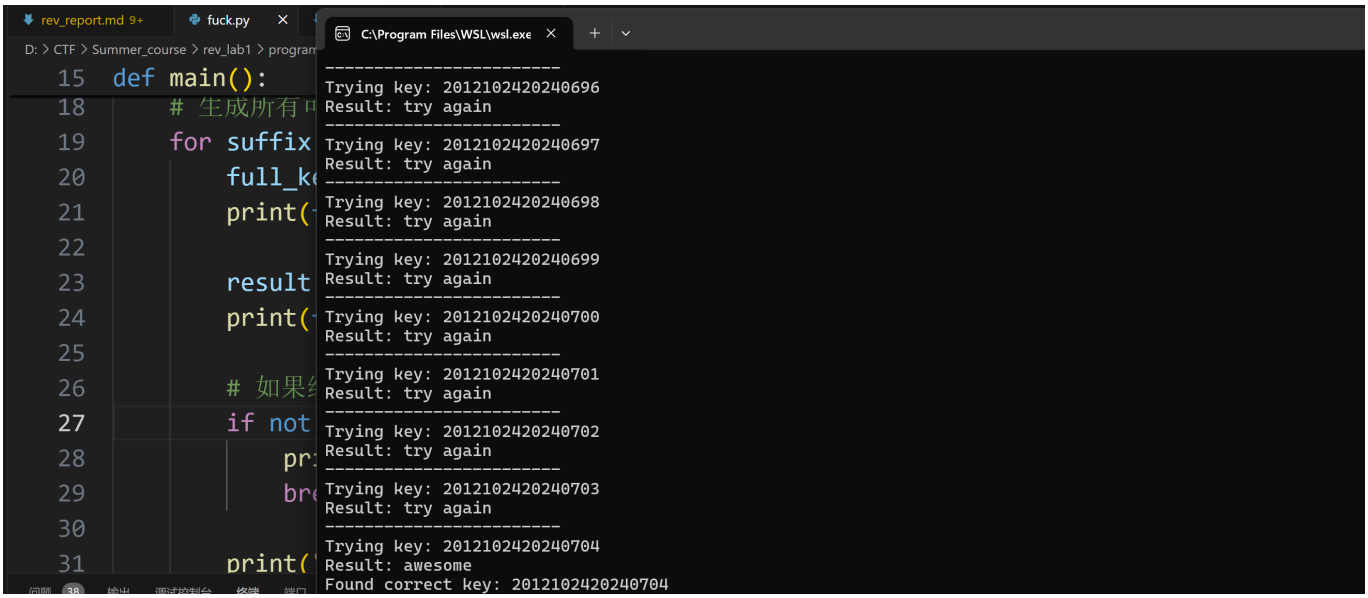
将程序拖入ida中查看，点开main函数之后进入whatthehell程序，看到里面有一个字符'A'以及一串由大小写组成的o的函数，再次点击之后仍显示一个"A",一直点下去，即可得到答案

```
; __unwind {
endbr64
push    rbp
mov     rbp, rsp
sub     rsp, 8
mov     [rbp+var_8], rdi
mov     rax, [rbp+var_8]
mov     byte ptr [rax], 41h ; 'A'
mov     rax, [rbp+var_8]
add     rax, 1
mov     rdi, rax
call    ooooooO
```

AAA{hope_u_have_fun~}

Task 3

本题文件为.bc文件，无法直接拖入ida中编译。所以需要进一步编译成elf文件，可以直接用clang编译，之后拖入ida查看。首先发现输入必需得是数字，然后看到程序对输入进行一个xcrc32操作，之后再和target里的数据进行比较。但本题分析程序内在逻辑较为复杂，故使用gdb调试法来寻找正确的输入。需要注意的是，程序中j+=2说明程序将两位数为一个整体来进行，所以输入必须是两位数（包括01，02等等！！！）设置断点，查看eax再与target中的值进行比对即可得到结果(其实只需要尝试二十几次)。。



经过不懈的尝试，试出答案为2012102420240704 最后四位由于尝试的时候忘了尝试0-9，所以甚至写了个脚本爆破()