﻿**CYB220 Lab 7 – Binary analysis**

**Due: Friday Nov 15, 11:59pm**

**Turn in: this report**

**Points: 30 pts**

**Task 1: in the assembly code for the main() function, mark/highlight**

**(with the corresponding color) the instruction for the following 3 statements in C++.**

**Note: you can execute the program with ./lab-password-v1**

﻿int main() {

int user\_input;

int pass1 = 345012;

int pass2 = 789456;

cout << "Do you know the passcode? enter the passcode: "<<endl;

cin >> user\_input;

if (user\_input == pass2){

cout << "Correct passcode!" <<endl;

}

else{

cout << "wrong passcode!\n" ;

cout << "Hint: the passcode has been hard coded in the program!\n";

cout << "Exit the program..." <<endl;

exit(0);

}

game();

return 0;

}

0000000000000cb3 <main>:

cb3: 55 push %rbp

cb4: 48 89 e5 mov %rsp,%rbp

cb7: 48 83 ec 20 sub $0x20,%rsp

cbb: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax

cc2: 00 00

cc4: 48 89 45 f8 mov %rax,-0x8(%rbp)

cc8: 31 c0 xor %eax,%eax

cca: c7 45 f0 b4 43 05 00 movl $0x543b4,-0x10(%rbp)

cd1: c7 45 f4 d0 0b 0c 00 movl $0xc0bd0,-0xc(%rbp)

cd8: 48 8d 35 91 02 00 00 lea 0x291(%rip),%rsi # f70 <\_ZStL19piecewise\_construct+0xc8>

cdf: 48 8d 3d 5a 13 20 00 lea 0x20135a(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

ce6: e8 75 fc ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

ceb: 48 89 c2 mov %rax,%rdx

cee: 48 8b 05 db 12 20 00 mov 0x2012db(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

cf5: 48 89 c6 mov %rax,%rsi

cf8: 48 89 d7 mov %rdx,%rdi

cfb: e8 70 fc ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

d00: 48 8d 45 ec lea -0x14(%rbp),%rax

d04: 48 89 c6 mov %rax,%rsi

d07: 48 8d 3d 52 14 20 00 lea 0x201452(%rip),%rdi # 202160 <\_ZSt3cin@@GLIBCXX\_3.4>

d0e: e8 0d fc ff ff callq 920 <\_ZNSirsERi@plt>

d13: 8b 45 ec mov -0x14(%rbp),%eax

d16: 3d d0 0b 0c 00 cmp $0xc0bd0,%eax

d1b: 75 43 jne d60 <main+0xad>

d1d: 48 8d 35 7b 02 00 00 lea 0x27b(%rip),%rsi # f9f <\_ZStL19piecewise\_construct+0xf7>

d24: 48 8d 3d 15 13 20 00 lea 0x201315(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

d2b: e8 30 fc ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

d30: 48 89 c2 mov %rax,%rdx

d33: 48 8b 05 96 12 20 00 mov 0x201296(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

d3a: 48 89 c6 mov %rax,%rsi

d3d: 48 89 d7 mov %rdx,%rdi

d40: e8 2b fc ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

d45: e8 c7 fd ff ff callq b11 <\_Z4gamev>

d4a: b8 00 00 00 00 mov $0x0,%eax

d4f: 48 8b 4d f8 mov -0x8(%rbp),%rcx

d53: 64 48 33 0c 25 28 00 xor %fs:0x28,%rcx

d5a: 00 00

d5c: 74 5f je dbd <main+0x10a>

d5e: eb 58 jmp db8 <main+0x105>

d60: 48 8d 35 4a 02 00 00 lea 0x24a(%rip),%rsi # fb1 <\_ZStL19piecewise\_construct+0x109>

d67: 48 8d 3d d2 12 20 00 lea 0x2012d2(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

d6e: e8 ed fb ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

d73: 48 8d 35 4e 02 00 00 lea 0x24e(%rip),%rsi # fc8 <\_ZStL19piecewise\_construct+0x120>

d7a: 48 8d 3d bf 12 20 00 lea 0x2012bf(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

d81: e8 da fb ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

d86: 48 8d 35 73 02 00 00 lea 0x273(%rip),%rsi # 1000 <\_ZStL19piecewise\_construct+0x158>

d8d: 48 8d 3d ac 12 20 00 lea 0x2012ac(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

d94: e8 c7 fb ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

d99: 48 89 c2 mov %rax,%rdx

d9c: 48 8b 05 2d 12 20 00 mov 0x20122d(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

da3: 48 89 c6 mov %rax,%rsi

da6: 48 89 d7 mov %rdx,%rdi

da9: e8 c2 fb ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

dae: bf 00 00 00 00 mov $0x0,%edi

db3: e8 d8 fb ff ff callq 990 <exit@plt>

db8: e8 c3 fb ff ff callq 980 <\_\_stack\_chk\_fail@plt>

dbd: c9 leaveq

dbe: c3 retq

**﻿ Task 2: in the assembly code of the game() function, mark/highlight the instruction for the following 5 statements in C++. Write some comments after the instructions to explain more.**

void game()

{

cout << "Let's play a simple game - guess a number" <<endl;

cout << "Guess a number (1-100)" <<endl;

cout << "You have 6 chances!" <<endl;

int user\_number;

int target\_number = get\_number();

for(int i=5; i>=0; i--){

cout << "What is your guess? " <<endl;

cin >> user\_number;

if (target\_number == user\_number){

cout << "Good! That's the target number!" <<endl;

break;

}

else if (target\_number > user\_number){

cout << "Too small!\n";

}

else{

cout << "Too big!" <<endl;

}

}

cout << "You lose the game..." <<endl;

}

**//This is the game function’s assembly code!**

0000000000000b11 <\_Z4gamev>:

b11: 55 push %rbp

b12: 48 89 e5 mov %rsp,%rbp

b15: 48 83 ec 20 sub $0x20,%rsp

b19: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax

b20: 00 00

b22: 48 89 45 f8 mov %rax,-0x8(%rbp)

b26: 31 c0 xor %eax,%eax

b28: 48 8d 35 81 03 00 00 lea 0x381(%rip),%rsi # eb0 <\_ZStL19piecewise\_construct+0x8>

b2f: 48 8d 3d 0a 15 20 00 lea 0x20150a(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

b36: e8 25 fe ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

b3b: 48 89 c2 mov %rax,%rdx

b3e: 48 8b 05 8b 14 20 00 mov 0x20148b(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

b45: 48 89 c6 mov %rax,%rsi

b48: 48 89 d7 mov %rdx,%rdi

b4b: e8 20 fe ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

b50: 48 8d 35 83 03 00 00 lea 0x383(%rip),%rsi # eda <\_ZStL19piecewise\_construct+0x32>

b57: 48 8d 3d e2 14 20 00 lea 0x2014e2(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

b5e: e8 fd fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

b63: 48 89 c2 mov %rax,%rdx

b66: 48 8b 05 63 14 20 00 mov 0x201463(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

b6d: 48 89 c6 mov %rax,%rsi

b70: 48 89 d7 mov %rdx,%rdi

b73: e8 f8 fd ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

b78: 48 8d 35 72 03 00 00 lea 0x372(%rip),%rsi # ef1 <\_ZStL19piecewise\_construct+0x49>

b7f: 48 8d 3d ba 14 20 00 lea 0x2014ba(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

b86: e8 d5 fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

b8b: 48 89 c2 mov %rax,%rdx

b8e: 48 8b 05 3b 14 20 00 mov 0x20143b(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

b95: 48 89 c6 mov %rax,%rsi

b98: 48 89 d7 mov %rdx,%rdi

b9b: e8 d0 fd ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

ba0: e8 25 ff ff ff callq aca <\_Z10get\_numberv>

ba5: 89 45 f4 mov %eax,-0xc(%rbp)

ba8: c7 45 f0 05 00 00 00 movl $0x5,-0x10(%rbp)

baf: 83 7d f0 00 cmpl $0x0,-0x10(%rbp)

bb3: 0f 88 bb 00 00 00 js c74 <\_Z4gamev+0x163>

bb9: 48 8d 35 45 03 00 00 lea 0x345(%rip),%rsi # f05 <\_ZStL19piecewise\_construct+0x5d>

bc0: 48 8d 3d 79 14 20 00 lea 0x201479(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

bc7: e8 94 fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

bcc: 48 89 c2 mov %rax,%rdx

bcf: 48 8b 05 fa 13 20 00 mov 0x2013fa(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

bd6: 48 89 c6 mov %rax,%rsi

bd9: 48 89 d7 mov %rdx,%rdi

bdc: e8 8f fd ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

be1: 48 8d 45 ec lea -0x14(%rbp),%rax

be5: 48 89 c6 mov %rax,%rsi

be8: 48 8d 3d 71 15 20 00 lea 0x201571(%rip),%rdi # 202160 <\_ZSt3cin@@GLIBCXX\_3.4>

bef: e8 2c fd ff ff callq 920 <\_ZNSirsERi@plt>

bf4: 8b 45 ec mov -0x14(%rbp),%eax

bf7: 39 45 f4 cmp %eax,-0xc(%rbp)

bfa: 75 2a jne c26 <\_Z4gamev+0x115>

bfc: 48 8d 35 1d 03 00 00 lea 0x31d(%rip),%rsi # f20 <\_ZStL19piecewise\_construct+0x78>

c03: 48 8d 3d 36 14 20 00 lea 0x201436(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

c0a: e8 51 fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

c0f: 48 89 c2 mov %rax,%rdx

c12: 48 8b 05 b7 13 20 00 mov 0x2013b7(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

c19: 48 89 c6 mov %rax,%rsi

c1c: 48 89 d7 mov %rdx,%rdi

c1f: e8 4c fd ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

c24: eb 4e jmp c74 <\_Z4gamev+0x163>

c26: 8b 45 ec mov -0x14(%rbp),%eax

c29: 39 45 f4 cmp %eax,-0xc(%rbp)

c2c: 7e 15 jle c43 <\_Z4gamev+0x132>

c2e: 48 8d 35 0b 03 00 00 lea 0x30b(%rip),%rsi # f40 <\_ZStL19piecewise\_construct+0x98>

c35: 48 8d 3d 04 14 20 00 lea 0x201404(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

c3c: e8 1f fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

c41: eb 28 jmp c6b <\_Z4gamev+0x15a>

c43: 48 8d 35 02 03 00 00 lea 0x302(%rip),%rsi # f4c <\_ZStL19piecewise\_construct+0xa4>

c4a: 48 8d 3d ef 13 20 00 lea 0x2013ef(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

c51: e8 0a fd ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

c56: 48 89 c2 mov %rax,%rdx

c59: 48 8b 05 70 13 20 00 mov 0x201370(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

c60: 48 89 c6 mov %rax,%rsi

c63: 48 89 d7 mov %rdx,%rdi

c66: e8 05 fd ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

c6b: 83 6d f0 01 subl $0x1,-0x10(%rbp)

c6f: e9 3b ff ff ff jmpq baf <\_Z4gamev+0x9e>

c74: 48 8d 35 da 02 00 00 lea 0x2da(%rip),%rsi # f55 <\_ZStL19piecewise\_construct+0xad>

c7b: 48 8d 3d be 13 20 00 lea 0x2013be(%rip),%rdi # 202040 <\_ZSt4cout@@GLIBCXX\_3.4>

c82: e8 d9 fc ff ff callq 960 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>

c87: 48 89 c2 mov %rax,%rdx

c8a: 48 8b 05 3f 13 20 00 mov 0x20133f(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>

c91: 48 89 c6 mov %rax,%rsi

c94: 48 89 d7 mov %rdx,%rdi

c97: e8 d4 fc ff ff callq 970 <\_ZNSolsEPFRSoS\_E@plt>

c9c: 90 nop

c9d: 48 8b 45 f8 mov -0x8(%rbp),%rax

ca1: 64 48 33 04 25 28 00 xor %fs:0x28,%rax

ca8: 00 00

caa: 74 05 je cb1 <\_Z4gamev+0x1a0>

cac: e8 cf fc ff ff callq 980 <\_\_stack\_chk\_fail@plt>

cb1: c9 leaveq

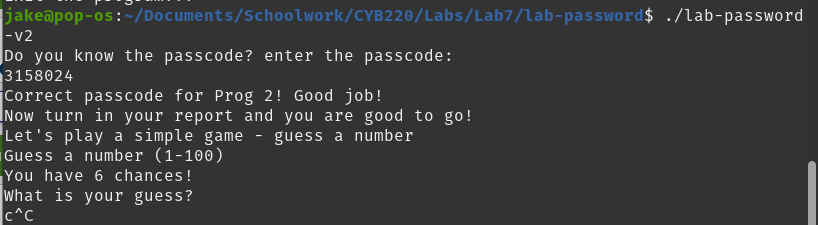
cb2: c3 retq

**Task3: Now move to the password-v2. There is no given source code, so the only file you can play with is the binary executable file.**

Goal: find out the passcode (in decimal) for this program.

Note: execute the program ./lab-password-v2, you only need to try the passcode part, no game (guess number) playing.

Hint: For this v2, I only changed the passcode with some arithmetic operations. I didn’t change other parts of the code. So if you examine the assembly code, you should be able to identify those arithmetic operations applied (The code structure is almost identical to v1). Once you figure out the correct passcode, take a screenshot of the messages you got (after typing the correct passcode) and paste it here. Briefly mention how you figured it out (screenshot on the related instructions may be helpful).

To find the passcode, I first used objdump and piped the output to a text file. I then found the main function in the assembly, and deleted everything else. After that, I used grep to search through the main function for all instances of cmp. After doing that, I found that the only instance of cmp compared %eax and -0xc(%rbp). I now knew that the passcode was at -0xc(%rbp). By tracing back through the code, I found its initial value of 0xc0bd0. I then found the shll and addl instructions, which shifted the code 2 bits to the left, then added 0xc8 to it. After going through all the steps and converting to decimal, I got 3158024.