

## Part 1

```
# problem 1
listing = currentProgram.getListing()
main = getGlobalFunctions("main")[0]
addr = main.getBody()
instructions = listing.getCodeUnits(addr, True)

instructions_dict = {}
for i in instructions:
    instr = str(i).split(' ')[0]
    if instr in instructions_dict.keys():
        instructions_dict[instr] += 1
    else:
        instructions_dict.update({instr:1})

for k, v in instructions_dict.items():
    print(k, v)
```

```
# Problem 2
outfile = open('pcode.txt', 'w')

main = getGlobalFunctions("main")[0]
main_body = main.getBody()
listing = currentProgram.getListing()
instructions = listing.getInstructions(main_body, True)
while instructions.hasNext():
    opcode = instructions.next()
    pcode = opcode.getPCode()
    for entry in pcode:
        outfile.writelines(str("  {}".format(entry)) + '\n')

outfile.close()
```

```

# Problem 3
listing = currentProgram.getListing()
main = getGlobalFunctions("main")[0] #
addr = main.getBody()
instructions = listing.getCodeUnits(addr, True) #

branches = 0
for i in instructions: #
    if(str(i)[0] == 'b'):
        branches += 1

print("The total amount of branch instructions is: " + str(branches))

```

lab8.py> Running...

('stmdb', 1)

('add', 8)

('sub', 2)

('b', 4)

('bhi', 1)

('mvn', 3)

('cmp', 4)

('bl', 6)

('blt', 1)

('ldmia', 1)

('cpy', 5)

('beq', 2)

('str', 7)

('mov', 3)

('ldrb', 2)

('ldr', 18)

The total amount of branch instructions is: 14

lab8.py> Finished!

Pcode is at the bottom

## Part 2

The first thing that this program does is takes two inputs. If you do not have two inputs then you will get an error message. Then the program checks to make sure that both arguments are at least seven characters. Then the program checks to see if the lengths of both inputs are the same. After that, the program checks to make sure that the length of the inputs is even. Then the program checks to see if the first half of the inputs are the same character. Finally, it checks to see if the remaining characters are different

```
[austinsbrown@inspiron5567 lab8]$ ./half-twins AAAAAAAA AAAAHHHH
Abby & Gabby: yaayy!! nice job! :D
[austinsbrown@inspiron5567 lab8]$
```

## Pcode for Part 1

```
(register, 0x80, 4) INT_SUB (register, 0x54, 4) , (const, 0x4, 4)
--- STORE (const, 0x1a1, 8) , (register, 0x80, 4) , (register, 0x58, 4)
(register, 0x80, 4) INT_SUB (register, 0x80, 4) , (const, 0x4, 4)
--- STORE (const, 0x1a1, 8) , (register, 0x80, 4) , (register, 0x4c, 4)
(register, 0x80, 4) INT_SUB (register, 0x80, 4) , (const, 0x4, 4)
(register, 0x54, 4) INT_ADD (register, 0x80, 4) , (const, 0x4, 4)
(register, 0x66, 1) INT_CARRY (register, 0x54, 4) , (const, 0x4, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x54, 4) , (const, 0x4, 4)
(register, 0x4c, 4) INT_ADD (register, 0x54, 4) , (const, 0x4, 4)
(register, 0x64, 1) INT_SLESS (register, 0x4c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x4c, 4) , (const, 0x0, 4)
(register, 0x66, 1) INT_LESSEQUAL (const, 0x18, 4) , (register, 0x54, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x54, 4) , (const, 0x18, 4)
(register, 0x54, 4) INT_SUB (register, 0x54, 4) , (const, 0x18, 4)
(register, 0x64, 1) INT_SLESS (register, 0x54, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x54, 4) , (const, 0x0, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffffe8, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x20, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffffe4, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x24, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (const, 0x8580, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffff4, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x2c, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffffe8, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x66, 1) INT_LESSEQUAL (const, 0x2, 4) , (register, 0x2c, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x2c, 4) , (const, 0x2, 4)
```

```

(unique, 0x6960, 4) INT_SUB (register, 0x2c, 4) , (const, 0x2, 4)
(register, 0x64, 1) INT_SLESS (unique, 0x6960, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (unique, 0x6960, 4) , (const, 0x0, 4)
(register, 0x62, 1) COPY (register, 0x66, 1)
(register, 0x61, 1) COPY (register, 0x65, 1)
(register, 0x60, 1) COPY (register, 0x64, 1)
(register, 0x63, 1) COPY (register, 0x67, 1)
--- CBRANCH (ram, 0x84a8, 4) , (register, 0x61, 1)
(register, 0x20, 4) LOAD (const, 0x1a1, 8) , (const, 0x8584, 4)
(register, 0x58, 4) COPY (const, 0x84a0, 4)
--- CALL (ram, 0x8300, 4)
(register, 0x2c, 4) INT_NEGATE (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
--- BRANCH (ram, 0x8574, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffffe4, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x66, 1) INT_CARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x2c, 4) INT_ADD (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xdf0, 4) INT_ADD (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xdf0, 4)
(register, 0x20, 4) COPY (register, 0x2c, 4)
(register, 0x58, 4) COPY (const, 0x84bc, 4)
--- CALL (ram, 0x8324, 4)
(register, 0x2c, 4) COPY (register, 0x20, 4)
(register, 0x66, 1) INT_LESSEQUAL (const, 0x9, 4) , (register, 0x2c, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x2c, 4) , (const, 0x9, 4)
(unique, 0x6960, 4) INT_SUB (register, 0x2c, 4) , (const, 0x9, 4)
(register, 0x64, 1) INT_SLESS (unique, 0x6960, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (unique, 0x6960, 4) , (const, 0x0, 4)
(register, 0x62, 1) COPY (register, 0x66, 1)
(register, 0x61, 1) COPY (register, 0x65, 1)
(register, 0x60, 1) COPY (register, 0x64, 1)
(register, 0x63, 1) COPY (register, 0x67, 1)
(unique, 0x3b0, 1) BOOL_NEGATE (register, 0x61, 1)
(unique, 0x3d0, 1) BOOL_AND (register, 0x62, 1) , (unique, 0x3b0, 1)
--- CBRANCH (ram, 0x84d8, 4) , (unique, 0x3d0, 1)
(register, 0x20, 4) LOAD (const, 0x1a1, 8) , (const, 0x8588, 4)
(register, 0x58, 4) COPY (const, 0x84d0, 4)
--- CALL (ram, 0x8300, 4)
(register, 0x2c, 4) INT_NEGATE (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)

```

```

(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
--- BRANCH (ram, 0x8574, 4)
(register, 0x2c, 4) COPY (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x2c, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffe4, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x66, 1) INT_CARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x2c, 4) INT_ADD (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xdf0, 4) INT_ADD (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xdf0, 4)
(register, 0x20, 4) COPY (register, 0x2c, 4)
(register, 0x58, 4) COPY (const, 0x84f4, 4)
--- CALL (ram, 0x8324, 4)
(register, 0x2c, 4) COPY (register, 0x20, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff8, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x2c, 4)
(register, 0x2c, 4) COPY (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x2c, 4)
--- BRANCH (ram, 0x8558, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff4, 4)
(register, 0x28, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x68, 1) COPY (register, 0x62, 1)
(register, 0x66, 1) INT_CARRY (register, 0x28, 4) , (register, 0x2c, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x28, 4) , (register, 0x2c, 4)
(register, 0x2c, 4) INT_ADD (register, 0x28, 4) , (register, 0x2c, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xdf0, 4) INT_ADD (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0x6b70, 1) LOAD (const, 0x1a1, 8) , (unique, 0xdf0, 4)
(register, 0x2c, 4) INT_ZEXT (unique, 0x6b70, 1)

```

```

(register, 0x66, 1) INT_CARRY (register, 0x2c, 4) , (const, 0x2, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x2c, 4) , (const, 0x2, 4)
(register, 0x28, 4) INT_ADD (register, 0x2c, 4) , (const, 0x2, 4)
(register, 0x64, 1) INT_SLESS (register, 0x28, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x28, 4) , (const, 0x0, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xfffffe4, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x66, 1) INT_CARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x2c, 4) INT_ADD (register, 0x2c, 4) , (const, 0x4, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xdf0, 4) INT_ADD (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x24, 4) LOAD (const, 0x1a1, 8) , (unique, 0xdf0, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x68, 1) COPY (register, 0x62, 1)
(register, 0x66, 1) INT_CARRY (register, 0x24, 4) , (register, 0x2c, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x24, 4) , (register, 0x2c, 4)
(register, 0x2c, 4) INT_ADD (register, 0x24, 4) , (register, 0x2c, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xdf0, 4) INT_ADD (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0x6b70, 1) LOAD (const, 0x1a1, 8) , (unique, 0xdf0, 4)
(register, 0x2c, 4) INT_ZEXT (unique, 0x6b70, 1)
(register, 0x68, 1) COPY (register, 0x62, 1)
(register, 0x66, 1) INT_LESSEQUAL (register, 0x2c, 4) , (register, 0x28, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x28, 4) , (register, 0x2c, 4)
(unique, 0x6980, 4) INT_SUB (register, 0x28, 4) , (register, 0x2c, 4)
(register, 0x64, 1) INT_SLESS (unique, 0x6980, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (unique, 0x6980, 4) , (const, 0x0, 4)
(register, 0x62, 1) COPY (register, 0x66, 1)
(register, 0x61, 1) COPY (register, 0x65, 1)
(register, 0x60, 1) COPY (register, 0x64, 1)
(register, 0x63, 1) COPY (register, 0x67, 1)
--- CBRANCH (ram, 0x854c, 4) , (register, 0x61, 1)
(register, 0x20, 4) LOAD (const, 0x1a1, 8) , (const, 0x858c, 4)
(register, 0x58, 4) COPY (const, 0x8544, 4)
--- CALL (ram, 0x8300, 4)
(register, 0x2c, 4) INT_NEGATE (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
--- BRANCH (ram, 0x8574, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)

```

```

(register, 0x66, 1) INT_CARRY (register, 0x2c, 4) , (const, 0x1, 4)
(register, 0x67, 1) INT_SCARRY (register, 0x2c, 4) , (const, 0x1, 4)
(register, 0x2c, 4) INT_ADD (register, 0x2c, 4) , (const, 0x1, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
--- STORE (const, 0x1a1, 8) , (unique, 0xe10, 4) , (register, 0x2c, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff0, 4)
(register, 0x28, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(unique, 0xe10, 4) INT_ADD (register, 0x4c, 4) , (const, 0xffffffff8, 4)
(register, 0x2c, 4) LOAD (const, 0x1a1, 8) , (unique, 0xe10, 4)
(register, 0x68, 1) COPY (register, 0x62, 1)
(register, 0x66, 1) INT_LESSEQUAL (register, 0x2c, 4) , (register, 0x28, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x28, 4) , (register, 0x2c, 4)
(unique, 0x6980, 4) INT_SUB (register, 0x28, 4) , (register, 0x2c, 4)
(register, 0x64, 1) INT_SLESS (unique, 0x6980, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (unique, 0x6980, 4) , (const, 0x0, 4)
(register, 0x62, 1) COPY (register, 0x66, 1)
(register, 0x61, 1) COPY (register, 0x65, 1)
(register, 0x60, 1) COPY (register, 0x64, 1)
(register, 0x63, 1) COPY (register, 0x67, 1)
(unique, 0x440, 1) INT_NOTEQUAL (register, 0x60, 1) , (register, 0x63, 1)
--- CBRANCH (ram, 0x8508, 4) , (unique, 0x440, 1)
(register, 0x20, 4) LOAD (const, 0x1a1, 8) , (const, 0x8590, 4)
(register, 0x58, 4) COPY (const, 0x8570, 4)
--- CALL (ram, 0x8300, 4)
(register, 0x2c, 4) COPY (const, 0x0, 4)
(register, 0x64, 1) INT_SLESS (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x2c, 4) , (const, 0x0, 4)
(register, 0x66, 1) COPY (register, 0x68, 1)
(register, 0x67, 1) COPY (register, 0x63, 1)
(register, 0x20, 4) COPY (register, 0x2c, 4)
(register, 0x66, 1) INT_LESSEQUAL (const, 0x4, 4) , (register, 0x4c, 4)
(register, 0x67, 1) INT_SBORROW (register, 0x4c, 4) , (const, 0x4, 4)
(register, 0x54, 4) INT_SUB (register, 0x4c, 4) , (const, 0x4, 4)
(register, 0x64, 1) INT_SLESS (register, 0x54, 4) , (const, 0x0, 4)
(register, 0x65, 1) INT_EQUAL (register, 0x54, 4) , (const, 0x0, 4)
(register, 0x80, 4) COPY (register, 0x54, 4)
(register, 0x4c, 4) LOAD (const, 0x1a1, 8) , (register, 0x80, 4)
(register, 0x80, 4) INT_ADD (register, 0x80, 4) , (const, 0x4, 4)
(register, 0x5c, 4) LOAD (const, 0x1a1, 8) , (register, 0x80, 4)
(register, 0x80, 4) INT_ADD (register, 0x80, 4) , (const, 0x4, 4)
(register, 0x54, 4) COPY (register, 0x80, 4)
(unique, 0x0, 4) INT_AND (register, 0x5c, 4) , (const, 0x1, 4)
(register, 0xb0, 1) INT_NOTEQUAL (unique, 0x0, 4) , (const, 0x0, 4)
(register, 0x69, 1) COPY (register, 0xb0, 1)
(register, 0x5c, 4) INT_AND (register, 0x5c, 4) , (const, 0xfffffffffe, 4)

```

--- RETURN (register, 0x5c, 4)