# 3827 OH

Eumin Hong (eh2890)

Columbia University

March 29, 2022

# Overview

1. Announcements
   - Upcoming Exams and Homework
   - Midterm
   - Feedback
2. Homework 5 Material
   - Overview and Relevant Lectures
   - General Part 3 Structure
3. Homework 6 Material
   - Relevant Lectures

# Announcements

# Announcements: Upcoming Exams and Homework

- HW5 due 4/1 (Ed post #252)
- HW6 due 4/4 (Ed post #256)

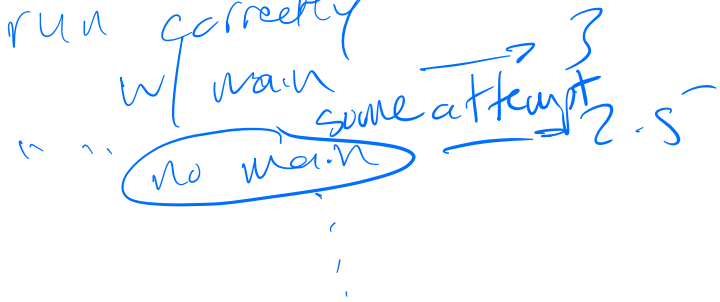
# Announcements: Midterm

- Do not throw away your exam if you plan to request a regrade
- Form: `https://forms.gle/8S8WYGD2B8LSsRmt9`
- I cannot comment on how a certain question was scored since I may not have scored it
- I do not know how the different exams are relatively curved

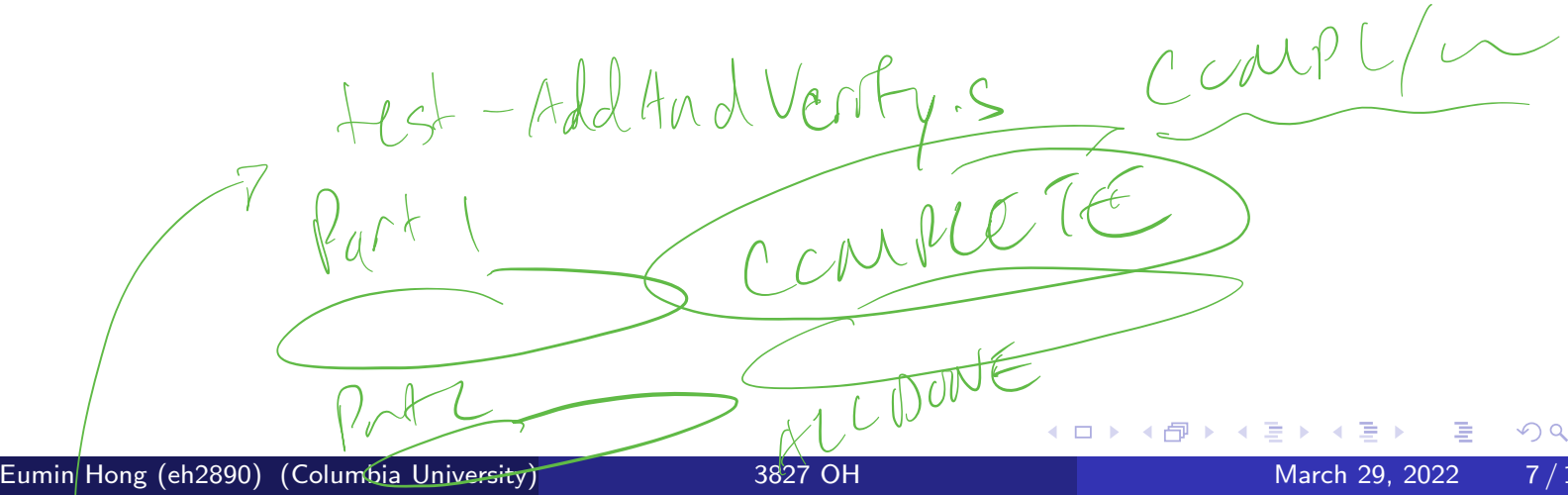
#274

# Announcements: Feedback

- Form: `https://forms.gle/cnUmKVNYN7WvRbHA6`



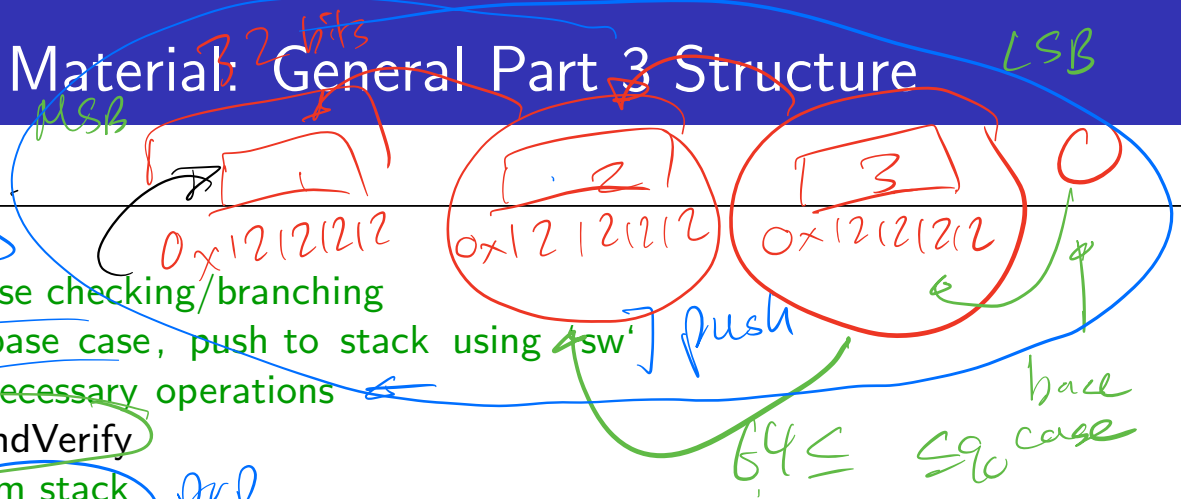

# Homework 5 Material



# Homework 5 Material: Overview and Relevant Lectures

- All Part *n* references are for Part 2.*n* in the "Coding Details" part of HW5 (i.e. the programming parts)
- Introduction to MIPS programming (Lecture 07, Slides 35-81)
- Conditional logic in MIPS (Lecture 07, Slides 82-95)
- Stack pointer and recursion (Lecture 07, Slides 96-111)
- MIPS calling conventions (Lecture 07, Slides 112-166)
- For Part 3 (AddAndVerify), use new testing program test−AddAndVerify−plus.s on CourseWorks
  - The original does not print any decrypted string even if you are correct
  - Also original says "ALL DONE" even if you are wrong
  - It is fairly clear when the new testing code outputs correct string
- For the main message (Part 4), the last four characters can be ignored

# Homework 5 Material: General Part 3 Structure



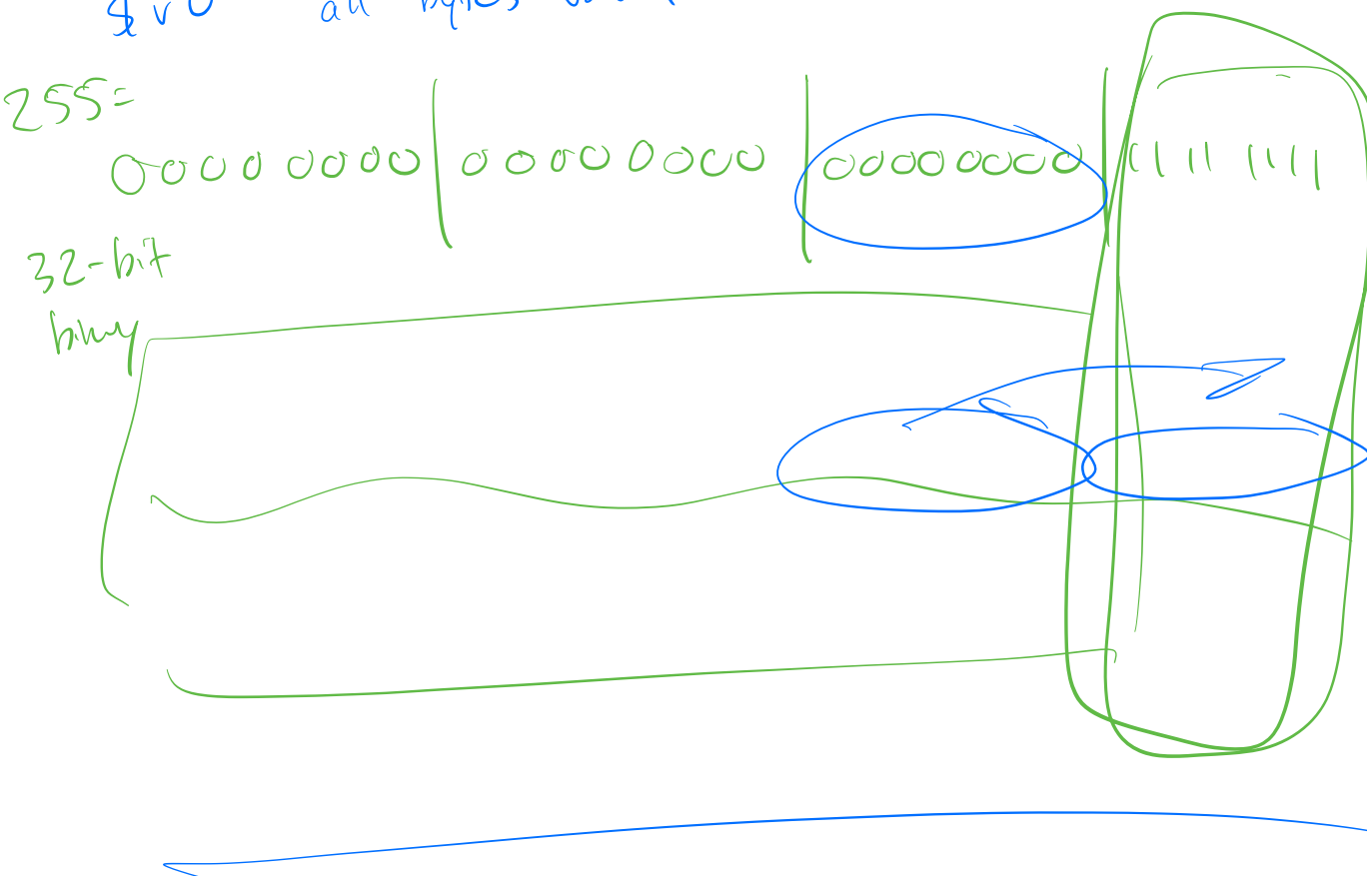

```
1   AddAndVerify:
2       # Base case checking/branching
3       # If not base case, push to stack using 'sw'
4       # Other necessary operations
5       jal AddAndVerify
6       # Pop from stack
7       # Check $v0 to see if suffix is valid for branching
8       # Other necessary operations
9       jal WordDecrypt
10      # Other necessary operations
11      jal IsCandidate
12      # Other necessary operations such as writing to destination address
13      jr $ra
```

- This is just a rough outline – you must fill in the rest of the code

# Is Candidate (part 2)

$a0 input word

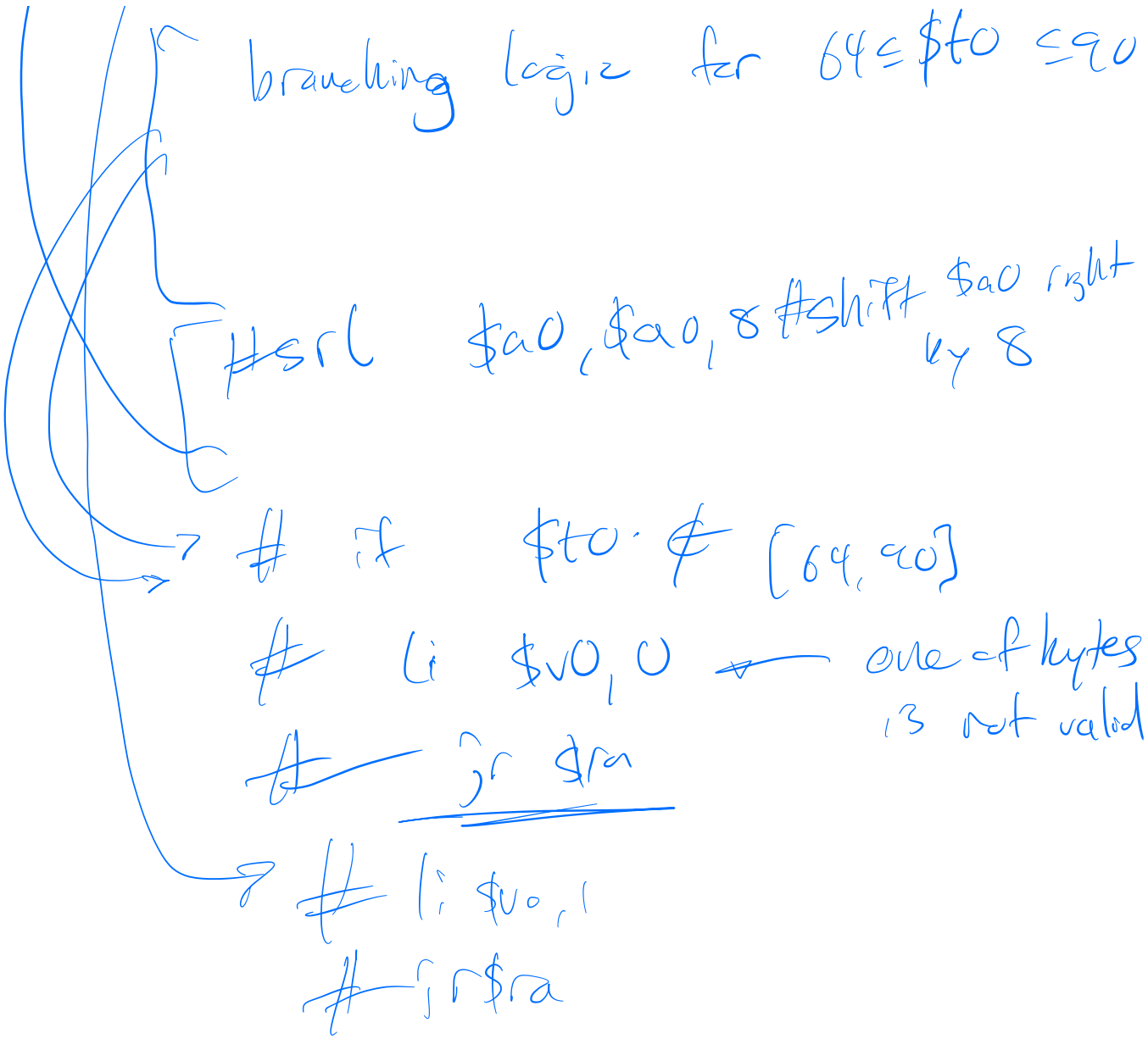
$v0 all bytes valid




Is Candidate:

iterator

```
# initialize mask, i
# loop:
#   check for i<4 or
#   and $t0, $a0, mask
```


branching logic for 64 ≤ $t0 ≤ 90
#srl $a0, $a0, 8 #shift $a0 right by 8
# if $t0 ∉ [64, 90]
# li $v0, 0 — one of bytes is not valid
# jr $ra
# li $v0, 1
# jr $ra

# Homework 6 Material

- Coincident selection (Lecture 10, Slides 41-58)
- Scaling memory using multiple chips (Lecture 10, Slides 59-85)