

3.1 - Week 3 - Applied - Theory

MIPS - Decisions/Arrays/Loops

bröther may i have some lööps



- To understand how to implement decisions and loops in MIPS.
- To test your knowledge of arrays and to understand how to implement them in MIPS.

Recap of Week 2

Let's spend some time recapping the concepts that we covered last week! It is important to remember those as they will become the basis for the next couple of weeks.

Question 1 *Submitted Aug 6th 2022 at 4:06:37 pm*

During a faithful translation, why must you reload the relevant values into registers for each line of python code?

Make the translation easier, thus reducing the chances of committing mistakes

Question 2 *Submitted Aug 6th 2022 at 4:01:52 pm*

Consider the following two lines of python code (ignore the fact that they aren't defined, just focus on these two lines)

```
y = x + 2  
z = 2 * x
```

Which of the following translations is faithful?

line 1
lw \$t3, x
addi \$t0, \$t3, 2
sw \$t0, y

line 2
add \$t0, \$t3, \$t3
sw \$t0, z

```
# line 1  
lw $t0, x  
addi $t0, $t0, 2  
sw $t0, y
```

•

```
# line 2  
lw $t0, x  
addi $t1, $0, 2  
mult $t0, $t1  
mflo $t0  
sw $t0, z
```

```
# line 1  
lw $t3, x  
addi $t0, $t3, 2  
sw $t0, y
```

○

```
# line 2  
addi $t1, $0, 2  
mult $t3, $t1  
mflo $t0  
sw $t0, z
```

```
# line 1  
lw $t0, x  
addi $t0, $t0, 2  
sw $t0, y
```

○

```
# line 2  
lw $t0, x  
add $t0, $t0, $t0  
sw $t0, z
```

Question 3 Submitted Aug 6th 2022 at 4:02:42 pm

When accessing an array, why do we add 1 to the index variable?

The first index is the length of array

Question 4 Submitted Aug 6th 2022 at 4:04:46 pm

When accessing an array, why do we multiply the index (+1) variable by 4?

4 bytes of memory per element

Question 5 Submitted Aug 6th 2022 at 4:04:03 pm

Now, lets use the knowledge from the last two questions and figure out how we are going to calculate the address of a particular element in an array.

How would you calculate the address of the i^{th} element in an array which starts at address x?

$x + 2 * i$

$4 * x + i + 1$

$4 * (i + 1) + x$

$(x + 1) + 4 * i$

Question 6 Submitted Aug 6th 2022 at 4:04:13 pm

What is the complexity of array access in MIPS?

$O(n^2)$

$O(1)$

$O(2^n)$

$O(n)$

⚠️ Advanced AI in MIPS! ⚠️

Just kidding! We are now going to cover decisions in MIPS. But if you really think about it



Anyway, lets figure out how we can define decisions in MIPS. We will need these for if-then-else conditions.

We'll cover the following topics -

- Labels and jumps
- Set less than statements
- Branch statements

Decisions, Decisions, Decisions!

Let's use our knowledge from the previous discussion to answer the following questions about the code below:

```
.data
n: .word 0
m: .word 0

.text
addi $v0 , $0 , 5
syscall
sw $v0 , n

addi $v0 , $0 , 5
syscall
sw $v0 , m

lw $t0 , n
lw $t1 , m
slt $t0 , $t1 , $t0
beq $t0 , $0 , one

lw $a0 , n
j two

one:lw $a0 , m

two:addi $v0 , $0 , 1
syscall

addi $v0 , $0 , 10
syscall
```

Question 1 Submitted Aug 9th 2022 at 8:31:13 am

What does this MIPS code do?

The MIPS code computes in \$a0 the maximum of n and m and prints it.

Question 2 Submitted Aug 9th 2022 at 8:20:25 am

Could line 22 be instead one: add \$a0, \$t1, \$0?

True

False

Question 3 Submitted Aug 9th 2022 at 8:20:57 am

Could line 19 be instead add \$a0, \$t0, \$0?

True

False

Question 4 Submitted Aug 9th 2022 at 8:21:04 am

Could line 22 be instead one: sw \$t1, \$a0?

True

False

Question 5 Submitted Aug 9th 2022 at 8:42:58 am

Comment the code and give labels one and two meaningful names.

```
bigger_than  
smaller_than
```

Loop-de-loop!

Okay! Now that we have a handle on decisions, we can turn our attention to Loops.

Loops are excellent in programming and essential to almost every programming language, and python is no exception.





We need to learn how loops function and hence, translate them from Python into MIPS.

To learn how to do that, we need to understand the nature of how loops work and how (in the image above) Barney keeps ending up back in the bar after Moe throws him out. We will cover the following topics:

- Types of loops
- How to convert every loop into a while loop
- Finally, how to use conditions and jumps to translate every loop

Test Your Knowledge on Loops

Alright!

Hopefully that all made sense, and now we can put it into practice and actually code up some loops! Here are a few questions to help you get started.

Question 1 Submitted Aug 6th 2022 at 4:08:05 pm

Consider the following Python code:

```
""" A very simple while loop, without much meaning """
n = int(input( "Enter integer: " ))

while n > 1 :
    print (n)
    n = n//2
```

What does this python code snippet do?

- Halves the number each run and prints it
- Halves the number and prints the quotient each time until the quotient becomes less than 1
- Halves the number and prints the quotient each time until the quotient becomes less than or equal to 1
- print the number, then changes it to the quotient when divided by 2 until the number itself becomes less than or equal to 1

Question 2 Submitted Aug 9th 2022 at 9:04:30 am

For the questions below, you will be given snippets of MIPS code, which is a faithful translation of the python code above. However, there is a mistake in each of the blocks. What you need to do is find the mistake, correct the code block and then submit the answer. Ready to go?

i Please note - If you think there is no error, just copy and paste the same code as the question. Also, if you press ` `` it will start a new code block and you must paste your answers in that block.

Question 3

```
.data

prompt : .asciiz "Enter integer: "

newline : .asciiz "\n"

n: .word 0
```

No response

Question 4 Submitted Aug 9th 2022 at 9:10:11 am

Code block 2 - Print prompt

```
.text
# Print "Enter integer: "
lw $a0 , prompt
addi $v0 , $0 , 5
syscall
```

```
.text
# Print "Enter integer: "
la $a0 , prompt
addi $v0 , $0 , 4
syscall
```

Question 5 Submitted Aug 9th 2022 at 9:09:51 am

Code block 3 - Read an integer input

```
# Read integer and store into n
addi $v0 , $0 , 4
syscall
sw $v0 , n
```

```
# Read integer and store into n
addi $v0 , $0 , 5
syscall
sw $v0 , n
```

Question 6 Submitted Aug 9th 2022 at 9:11:11 am

Code block 4 - Starting of the while loop

```
while_gt_1 : # while n > 1
```

```
lw $t0 , n
addi $t1 , $0 , 1
slt $t2 , $t0 , $t1 # is n > 1?
bne $t2 , $0 , end_while_gt_1 # if not , goto end of while loop
```

```
while_gt_1 :# while n > 1
lw $t0 , n
addi $t1 , $0 , 1
slt $t2 , $t1 , $t0 # is n > 1?
beq $t2 , $0 , end_while_gt_1 # if not , goto end of while loop
```

Question 7 Submitted Aug 9th 2022 at 9:11:33 am

Code Block 5 - Print the integer

```
# body of while : print integer n
lw $a0 , n
add $v0 , $0 , 1
syscall
```

```
# body of while : print integer n
lw $a0 , n
addi $v0 , $0 , 1
syscall
```

Question 8 Submitted Aug 9th 2022 at 9:12:31 am

Code Block 6 - Cut in half, store quotient

```
# body of while : n = n//2
lw $t0 , n
addi $t1, $0, 2
div $t1, $t0
mfhi $t0
sw $t0 , n
```

```
# body of while : n = n//2
lw $t0 , n
addi $t1, $0, 2
div $t0, $t1
mflo $t0
sw $t0 , n
```

Question 9 Submitted Aug 9th 2022 at 9:12:43 am

Code Block 7 - Print a newline character

```
# Print newline
la $a0 , newline
addi $v0 , $0 , 4
syscall
```

```
# Print newline
la $a0 , newline
addi $v0 , $0 , 4
syscall
```

Question 10 Submitted Aug 9th 2022 at 9:13:13 am

Code Block 8 - Jump back to loop and exit

```
# Go back to while loop
j while_gt_1
```

```
end_while_gt_1 : # End while
```

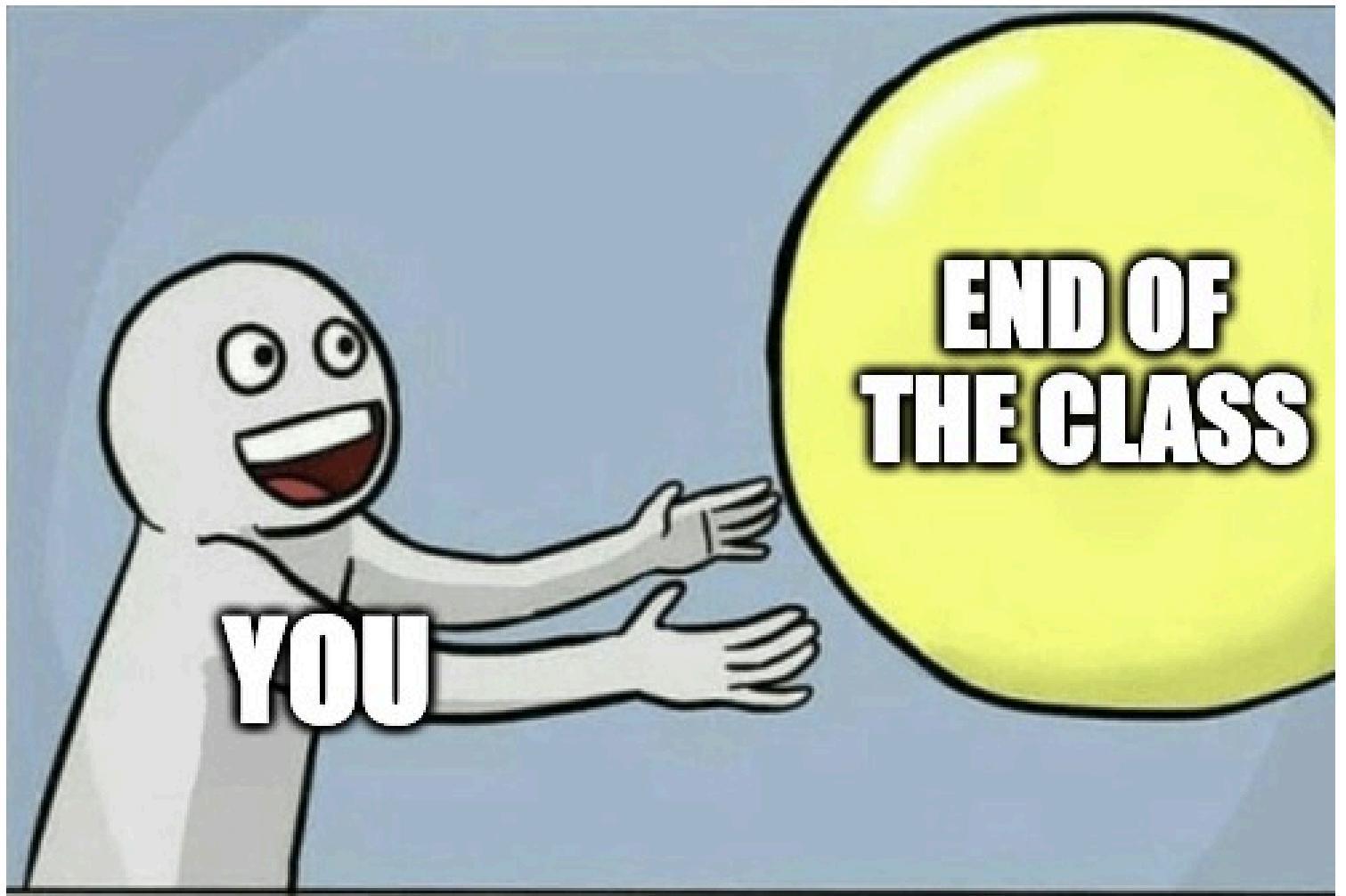
```
# Exit the program
addi $v0 , $0 , 9
syscall
```

```
# Go back to while loop
j while_gt_1
```

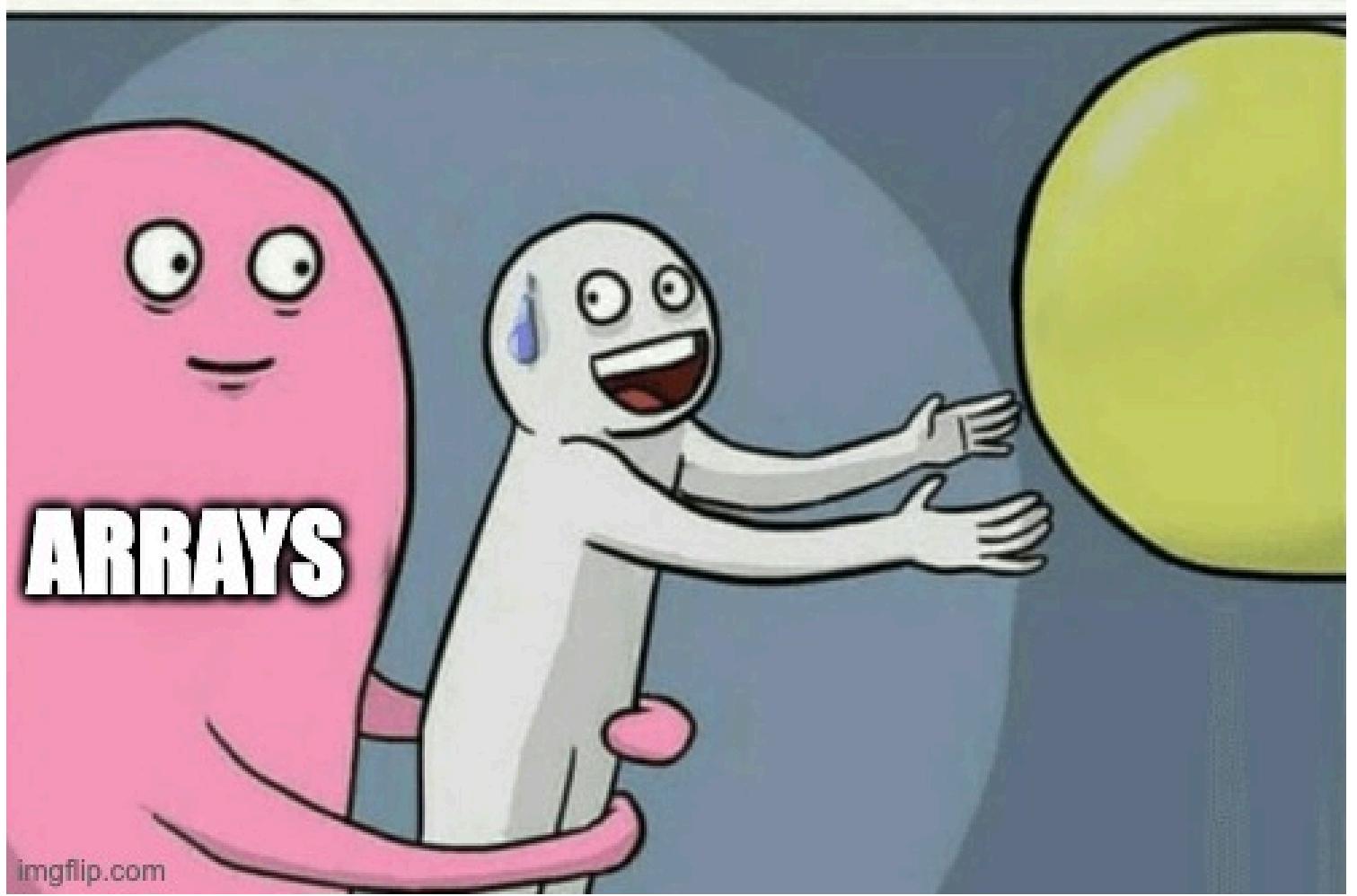
```
endwhile_gt_1 : # End while
```

```
# Exit the program
addi $v0 , $0 , 10
syscall
```

Creating and working with Arrays



YOU



ARRAYS

We can now proceed to our work with arrays.

Arrays in MIPS are constructed using addresses and we must learn how to read and store information to a particular address.

There are a few topics that we will now cover:

- Sizing of a register in MIPS
- Short-hand multiplication and division in MIPS (`sll` and `sra`)
- The two different ways to create arrays in MIPS
 - The static way
 - The dynamic way: addressing each element
- How to read and store an element at a particular position in the array

A Quiz on MIPS Arrays

Let us go through some questions about arrays on MIPS and test our knowledge!

Question 1 Submitted Aug 6th 2022 at 4:08:36 pm

What is the size of one register in MIPS? (PS - more than one answer could be correct)

4 bits

8 bits

32 bits

4 bytes

Question 2 Submitted Aug 9th 2022 at 9:17:20 am

What will be the value of `$t1` at the end of the following statements:

```
addi $t0, $0, 10  
sll $t1, $t0, 2
```

40

Question 3 Submitted Aug 9th 2022 at 9:14:59 am

What is the system call code for allocating memory?

9

Question 4 Submitted Aug 6th 2022 at 4:09:24 pm

How many bytes of memory would we need for an array that contains 4 elements?

20

Question 5 Submitted Aug 6th 2022 at 4:09:37 pm

What is the purpose of using () in MIPS?

- They look C()()L
- They help with separating different elements of the line of code
- They help read and store from particular addresses
- They are a part of the syntax of sw and lw

Question 6 Submitted Aug 6th 2022 at 4:12:22 pm

We have to translate line 2 from the following python code into MIPS:

```
length = int(input())
array = [None]*length
```

lw \$t0, length

addi \$t0, \$t0, 1

sll \$a0, \$t0, 2

addi \$v0, \$0, 9

syscall

sw \$v0, array

Question 7 Submitted Aug 9th 2022 at 9:35:54 am

Now, we need to translate line 6 from the following python code into MIPS:

```
z = int(input("Enter length : "))

y = [ 0 ] * z

for i in range(z):
    y[i] = int(input("Enter integer : "))
```

Write down the statements required to do so. Please also try and get into the habit of commenting

your code.

```
.data
int: .word 0
prompt: .asciiz "Enter integer : "
newline: .asciiz "\n"

.text
la $a0, prompt
addi $v0, $0, 4
syscall

la $a0, newline
addi $v0, $0, 4
syscall

addi $v0 , $0 , 5
syscall
sw $v0 , int
```

```
#load y
lw $t1, y

#load the ith element
lw $t0, i
addi $t0, $t0, 2
sll $t0, $t0, $t1
add $t0, $t0, $t1

#take int input
addi $v0, $0, 5

#store
sw $v0, int
```

RECAP - Week 3 & Potential Interview Questions

1. Explain the process of creating an array in MIPS
2. How would you translate a for loop from Python to MIPS (for i in range(n))
3. Why do we need to add 1 to the index of any array while accessing the elements?
4. What arithmetic operation is performed when you shift right 2 bits in binary?
5. Convert 11011001 from binary to hexadecimal