FIT2085-S1-2021
Week 3 Workshop Malaysia

Learning Objectives

- Coding simple MIPS programs involving decisions and iterations.
- Learning how to use Arrays in MIPS.

If-elif-else in MIPS

Consider the following Python code:

```
secret = 42
guess = int(input("Enter guess: "))
diff = secret - guess

if diff < 0:
    print("Too high")
elif diff == 0:
    print("Good guess!")
else:
    print("Too low")
```

Faithfully translate the above code into a properly commented MIPS program using the file provided.

For loops in MIPS

Consider the following Python code:

```
previous = -1
current = 1

size = int(input("Input size: "))

for i in range(0, size, 1):
    current = current + previous
    previous = current - previous
    print(current, end=' ')
print()
```

We have done the work of translating the for loop into a while loop below:

```
previous = -1
current = 1

size = int(input("Input size: "))

i = 0
while i < size:
    current = current + previous
    previous = current - previous
    print(current, end=' ')
    i += 1
print()
```

Faithfully translate the above code (with the while loop) into a properly commented MIPS program using the file provided.

Arrays in MIPS

Consider the following Python code:

```
secret = 42
guess = int(input("Enter guess: "))
diff = secret - guess

if diff < 0:
    print("Too high")
elif diff == 0:
    print("Good guess!")
else:
    print("Too low")
```

Note how the complexity of creating a subarray of size k from an array of size n with this code is O(k) (so it is independent of n).

Faithfully translate the above code (with the while loop) into a properly commented MIPS program using the file provided.

If you are struggling to understand the memory and addressing aspects of this question, consider using MARS's GUI and debugger locally on your computer.