Exercise 2-28

```mips
sll $a2, $a2, 2
sll $a3, $a3, 2
add $v0, $zero, $zero
add $t0, $zero, $zero
outer: add $t4, $a0, $t0
lw $t4, 0($t4)
add $t1, $zero, $zero
inner: add $t3, $a1, $t1
lw $t3, 0($t3)
bne $t3, $t4, skip
addi $v0, $v0, 1
skip
addi $t1, $t1, 4
bne $t1, $a3, inner
addi $t0, $t0, 4
bne $t0, $a2, outer
```

- (10 pts) Add comments to the MIPS code above. This code processes two arrays and produces an important value in register $v0. Assume that each array consists of 2500 words indexed 0 through 2499, that the base addresses of the arrays are stored in $a0 and $a1 respectively, and their sizes (2500) and stored in $a2 and $a3, respectively. In your comments, call the arrays Array1 and Array2.

- (10 pts) In one sentence, what does this code compute and store in $v0? Be very specific (make sure that your answer doesn’t have more than one interpretation of what it means)
(5 pts) Exercise 2-31

Suppose you are given the code for the following function:

```c
int function1(int a, int b);
```

Write MIPS code to call `function1(3, 7)` and then store the result in `$s0`.

(5 pts) Exercise 2-32

Now you have this definition for `function1`:

```c
int function1(int a, int b) {
    return (a - b);
}
```

Write MIPS code to define `function1`.
Write MIPS code to define the following function:

```c
int cat(int a, int b) {
    if (a < b)
        return a;
    else
        return b;
}
```
(5 pts) Exercise 2-36

- Write the MIPS code to define the following function
  int function2(int g, int h)
  { return g + function1(g, h); }
(You will need to store something on the stack – why?)
Exercise 2-37

- Write the MIPS code to define the following function
  int function3(int a, int b)
  { return function6(a) + function7(b); }
(You will need to store something on the stack – why?)
(15 pts) Exercise 2-38

- Write the MIPS code to define the following function
  int lemur(int a, int b)
  { return panda(a) + b; }
• Write the MIPS code to \textbf{define} the following function
  
  \begin{verbatim}
  int alpaca(int x, int y, int z)
  {
    int temp = ferret(y, z);
    return hedgehog(temp + x);
  }
  \end{verbatim}