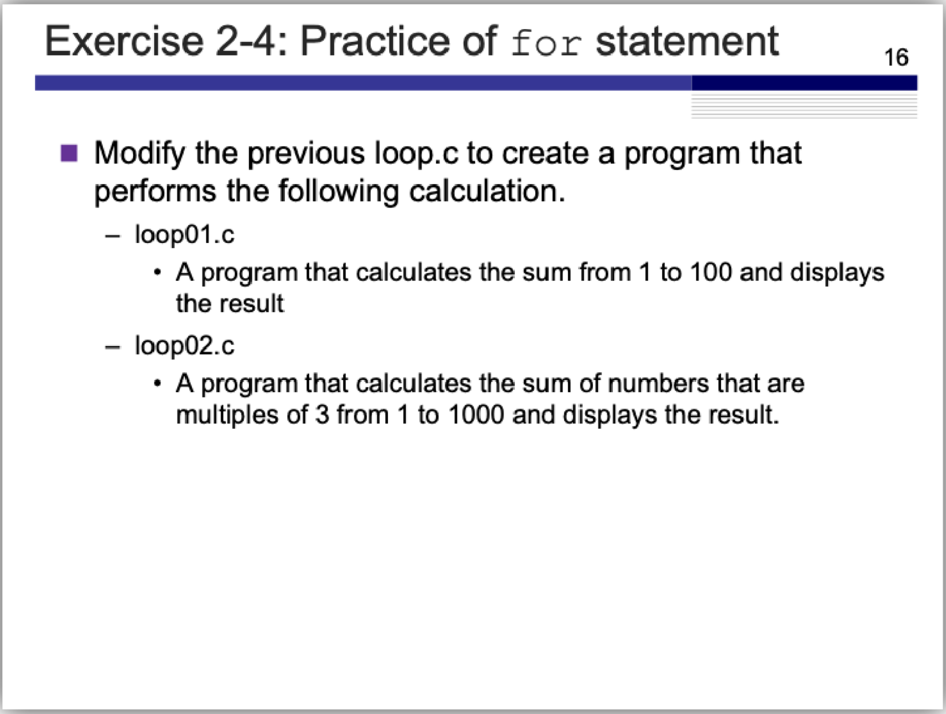
Exercise 2-4: Practice of `for` statement



Exercise 2-4: Practice of `for` statement

16

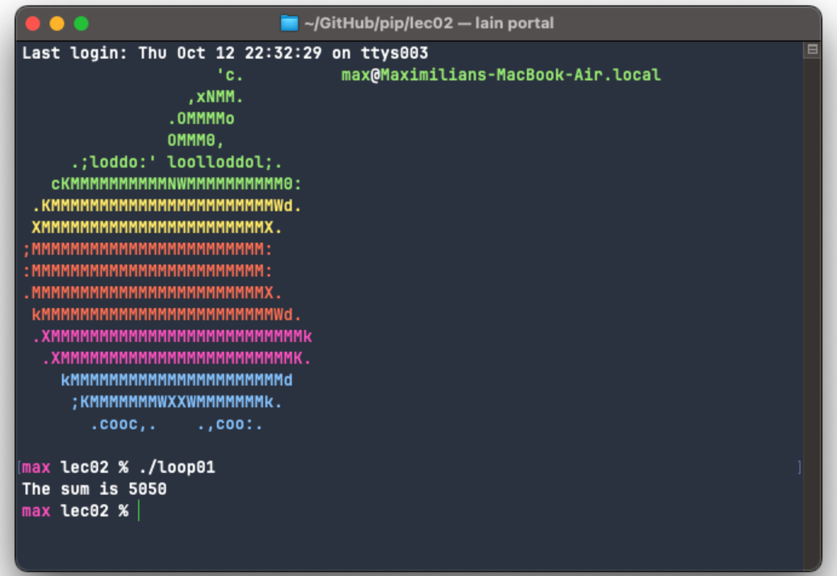
- Modify the previous loop.c to create a program that performs the following calculation.
  - loop01.c
    - A program that calculates the sum from 1 to 100 and displays the result
  - loop02.c
    - A program that calculates the sum of numbers that are multiples of 3 from 1 to 1000 and displays the result.

loop01.c

Exercise 2-4 demonstrates the simple applications of the `for` statement. First, we are tasked to calculate the sum of whole numbers from 1 to 100 and display the result. Although this seems easy enough at first glance, the logic might be a little harder.

```c
#include <stdio.h>

int main()
{
    /**** variable declaration ****/
    int i,sum = 0;

    /**** processing contents ****/
    for(i=0;i<100;i++) {
        sum += i+1;
    }

    printf("The sum is %d\n", sum);

    return 0;

}
```

If we only do `sum += i`, the program will start adding from the initial value of `i`, which is 0. Since we want to do $1 + 2 + ... + 100$, we need to start from 1. To do this, we offset the addition into `sum` by doing `sum += 1`, so that it starts by adding 1 into `sum` when `i=0`, adding 2 into `sum` when `i=1`, and so on until it adds 100 into `sum` when `i=99`. Compiling and running the program yields:



```
Last login: Thu Oct 12 22:32:29 on ttys003
                    'c.          max@Maximilians-MacBook-Air.local
                 ,xNMM.
               .OMMMMo
               OMMM0,
     .;loddo:' loolloddol;.
   cKMMMMMMMMMMNWMMMMMMMMMM0:
 .KMMMMMMMMMMMMMMMMMMMMMMMWd.
 XMMMMMMMMMMMMMMMMMMMMMMMX.
;MMMMMMMMMMMMMMMMMMMMMMMM:
:MMMMMMMMMMMMMMMMMMMMMMMM:
.MMMMMMMMMMMMMMMMMMMMMMMMX.
 kMMMMMMMMMMMMMMMMMMMMMMMMWd.
 .XMMMMMMMMMMMMMMMMMMMMMMMMMMk
  .XMMMMMMMMMMMMMMMMMMMMMMMMK.
    kMMMMMMMMMMMMMMMMMMMMMMd
     ;KMMMMMMMWXXWMMMMMMMk.
       .cooc,.    .,coo:.

max lec02 % ./loop01
The sum is 5050
max lec02 %
```

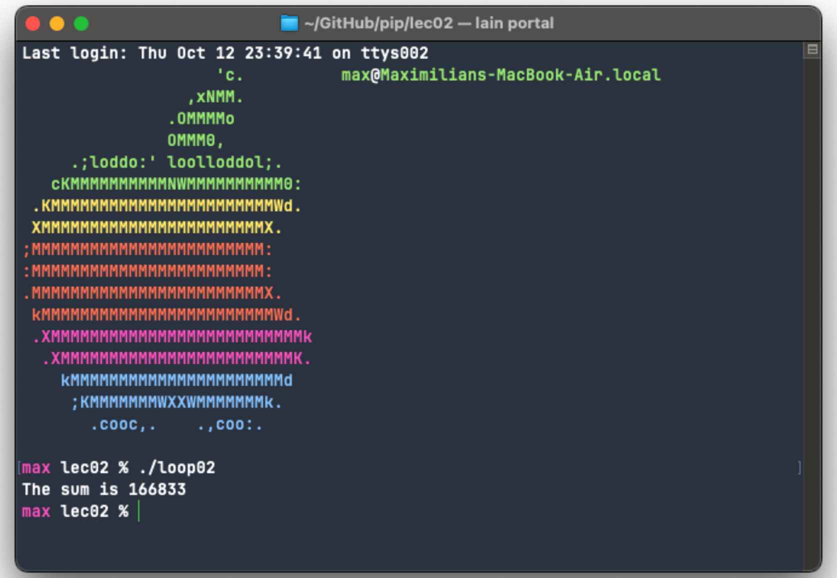
loop02.c

Next we are tasked to calculate the sum of numbers that are multiples of 3 from 1 to 1000 and display the result. There are many ways to do this, for example, the most intuitive way after reading the problem statement is probably by iterating through all the numbers from 1 to 1000, check if it's divisible by 3, and if it is, add it to the sum. However, this method requires us to run the `if` condition a thousand times. While C is incredibly fast, it is bad programming practice to rely on many logical processing when avoidable.

Another way to do this is actually simpler. We can use the `for` loop to work for us.

```c
#include <stdio.h>

int main()
{
    /**** variable declaration ****/
    int i,sum = 0;

    /**** processing contents ****/
    for(i=3;i<=1000;i+=3) {
        sum += i;
    }

    printf("The sum is %d\n", sum);

    return 0;
}
```

Instead of starting at 1 intuitively, since we know that 1 is not a multiple of 3, we can start at 3 and add 3 to `i` after each iteration. Compiling and running the code yields:



```
Last login: Thu Oct 12 23:39:41 on ttys002
                    'c.          max@Maximilians-MacBook-Air.local
                 ,xNMM.
               .OMMMMo
               OMMM0,
     .;loddo:' loolloddol;.
   cKMMMMMMMMMMNWMMMMMMMMMM0:
 .KMMMMMMMMMMMMMMMMMMMMMMMWd.
 XMMMMMMMMMMMMMMMMMMMMMMMX.
;MMMMMMMMMMMMMMMMMMMMMMMM:
:MMMMMMMMMMMMMMMMMMMMMMMM:
.MMMMMMMMMMMMMMMMMMMMMMMMX.
 kMMMMMMMMMMMMMMMMMMMMMMMMWd.
 .XMMMMMMMMMMMMMMMMMMMMMMMMMMk
  .XMMMMMMMMMMMMMMMMMMMMMMMMK.
    kMMMMMMMMMMMMMMMMMMMMMMd
     ;KMMMMMMMWXXWMMMMMMMk.
       .cooc,.    .,coo:.

max lec02 % ./loop02
The sum is 166833
max lec02 %
```