Lab 8: Learning from the Past

郑悟强 PB22051082

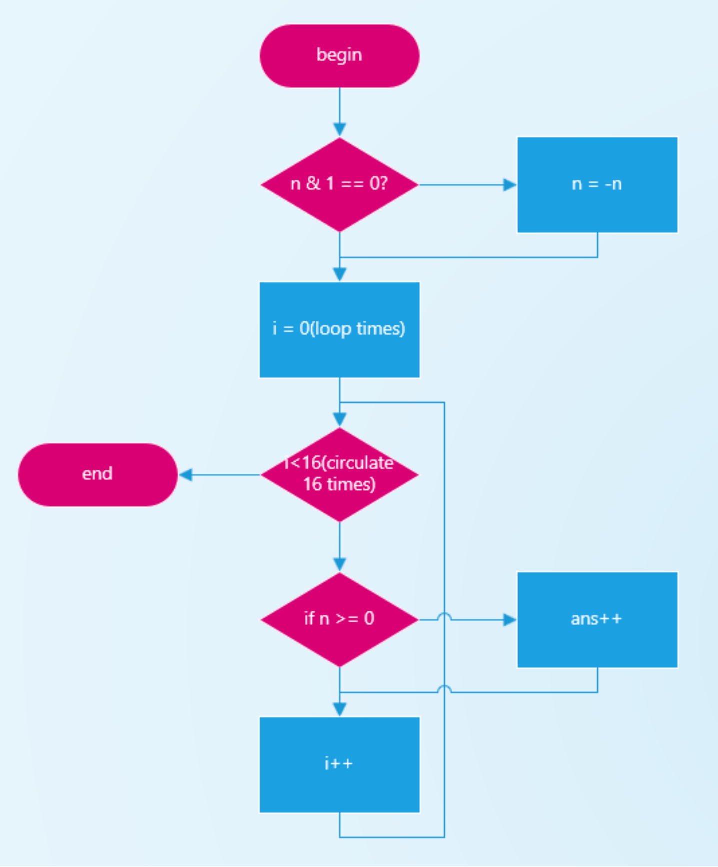
2023.1.11

1 Purpose

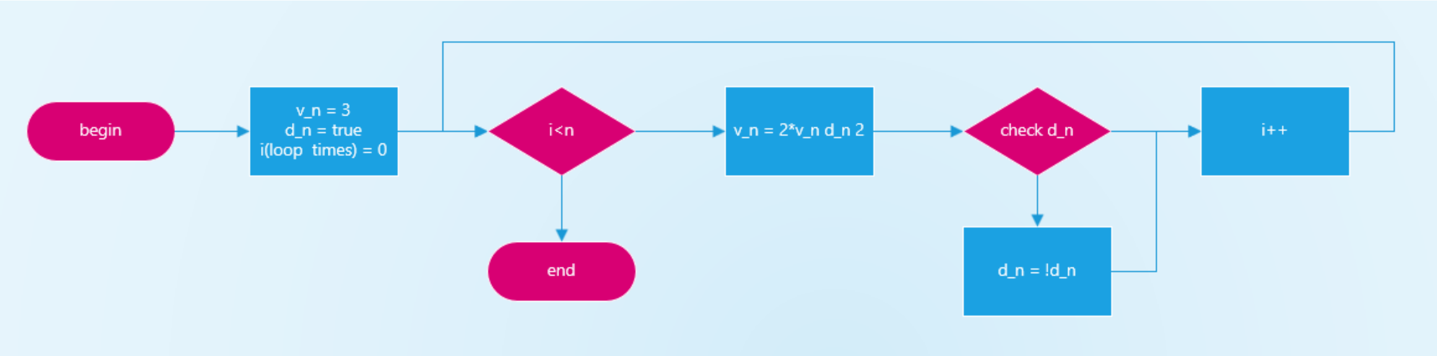
After completing the previous seven labs, this final lab should be relatively straightforward for you. In this lab, you are required to implement all the code from previous labs using a high-level programming language (e.g., C/C++). Note that the algorithm should remain consistent with the methods used in the earlier labs (e.g., modulo operations cannot be replaced with % as in the second lab, and strcmp() cannot be used in lab 3).

2 Principles

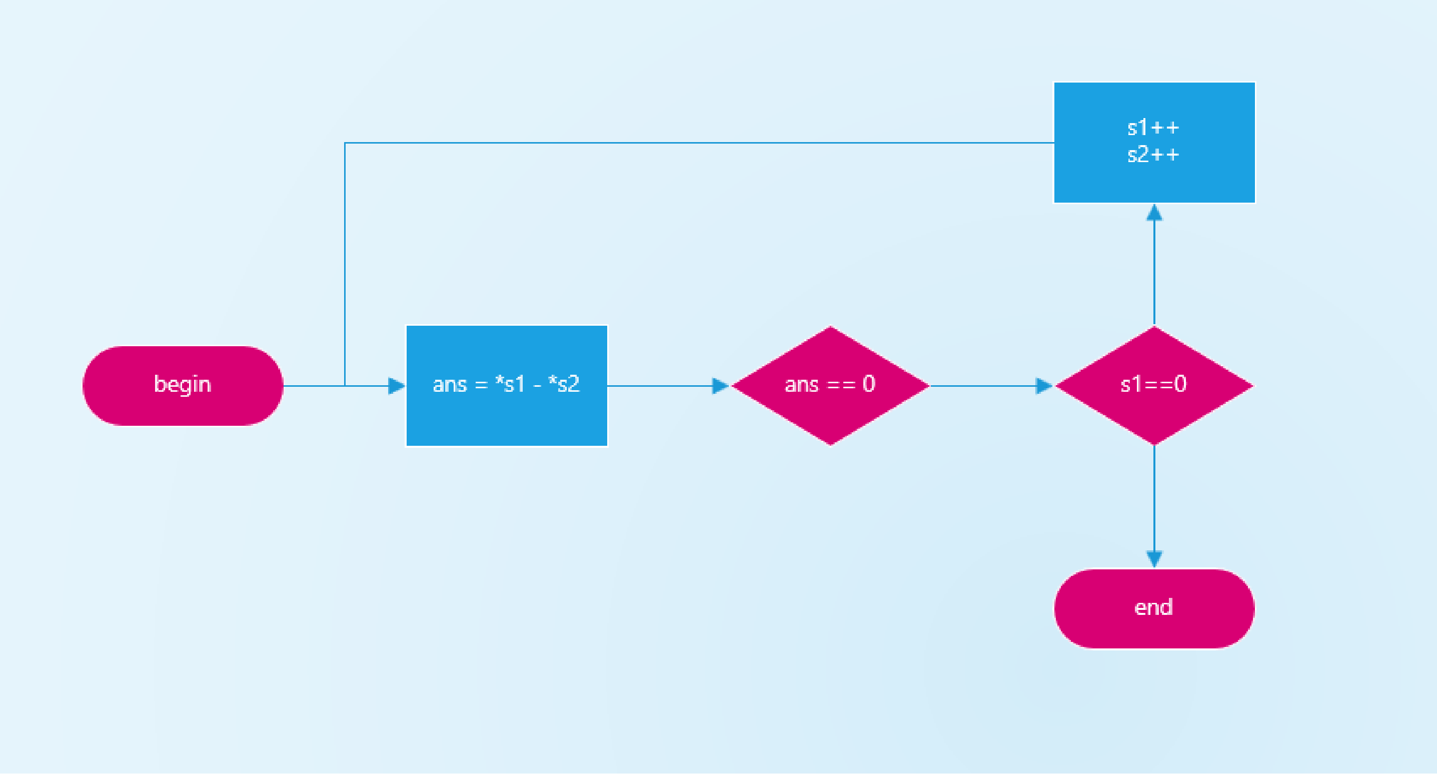
2.1 Lab 1



2.2 Lab 2



2.3 Lab 3



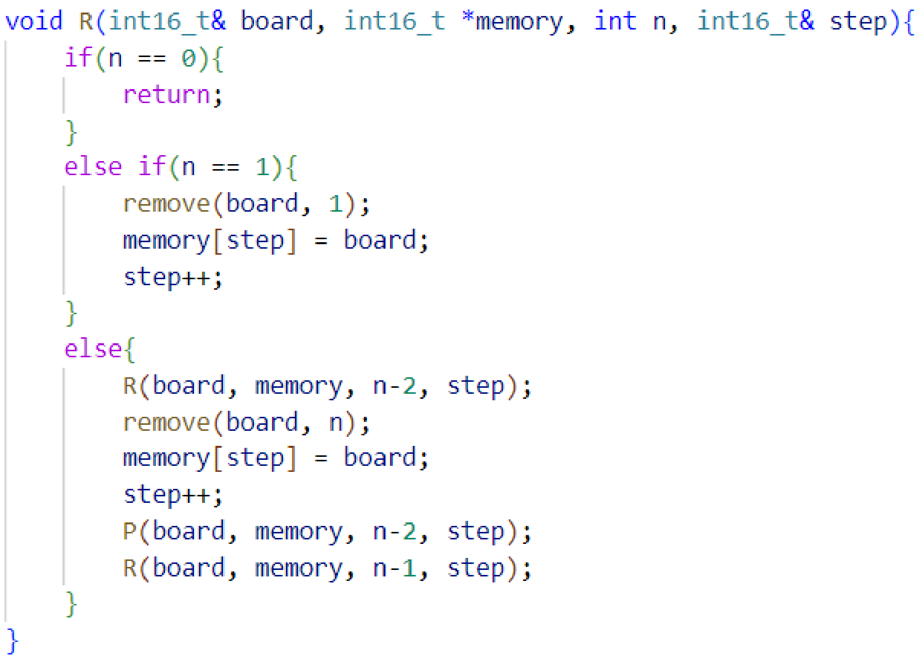
2.4 Lab 4

Recursive Expression:

$$R(n) = \begin{cases} \text{nothing to do} & , n = 0 \\ \text{remove the 1st ring} & , n = 1 \\ R(n-2) + \text{remove the nth ring} + P(n-2) + R(n-1) & , n \geq 2 \end{cases}$$

$$P(n) = \begin{cases} \text{nothing to do} & , n = 0 \\ \text{put the 1st ring} & , n = 1 \\ P(n-1) + R(n-2) + \text{put the nth ring} + P(n-2) & , n \geq 2 \end{cases}$$

Just follow this recursive expression.



```
void R(int16_t& board, int16_t *memory, int n, int16_t& step){
    if(n == 0){
        return;
    }
    else if(n == 1){
        remove(board, 1);
        memory[step] = board;
        step++;
    }
    else{
        R(board, memory, n-2, step);
        remove(board, n);
        memory[step] = board;
        step++;
        P(board, memory, n-2, step);
        R(board, memory, n-1, step);
    }
}
```

P is similar to R.

3 Procedure

3.1 How to check whether to shift d_n in lab 2

Let `int16_t temp = v_n;` and keep executing `temp = temp - 8` until `temp <= 0`.
If `temp == 0`, shift d_n;
else `temp = v_n - 8`, then change 8 to 10 and repeat the above operation.

3.2 How to change the nth bit in lab 4

Remove (to set the nth bit to 1):

1. Set `mask = 1`.
2. Let `mask * 2 n times`, resulting in `mask` being the nth bit equal to 1, with all other bits equal to 0.
3. `board = board | mask`.

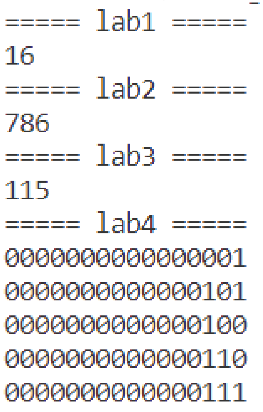
Put (to set the nth bit to 0):

1. Set `mask = 1`.
2. Let `mask * 2 n times` (mask's nth bit is 1, and all other bits are 0).
3. `mask = ∼mask` (mask's nth bit is 0, and all other bits are 1).
4. `board = board & mask`.

4 Results

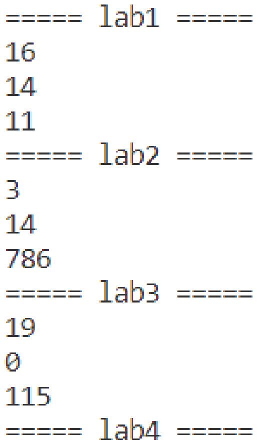
The last digit of my student ID is 2.

1.test.txt (delivered by TA)



```
===== lab1 =====
16
===== lab2 =====
786
===== lab3 =====
115
===== lab4 =====
0000000000000001
0000000000000101
0000000000000100
0000000000000110
0000000000000111
```

2.test_multi.txt (delivered by TA)



```
===== lab1 =====
16
14
11
===== lab2 =====
3
14
786
===== lab3 =====
19
0
115
===== lab4 =====
```

The output of lab 4 is too long and takes up too much space, but it turns out to be correct.