國立清華大學資訊工程學系

11110 CS 410000 計算機結構

Homework 2

Deadline: 2021/10/19 (23:59)

There are two parts in this homework.

PART I. (Load, Store, Add, Sub)

Please load the data 4($gp) as A, 8($gp) as B, and 12($gp) as C, and do the following calculations.

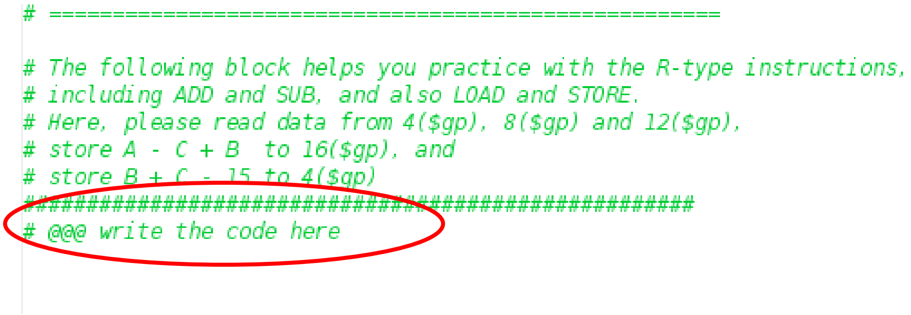
D = A - C + B, store D to 16($gp)

E = B + C - 15, store E to 4($gp)

global pointer(gp): A system defined pointer which is used to refer to constant.

Hint

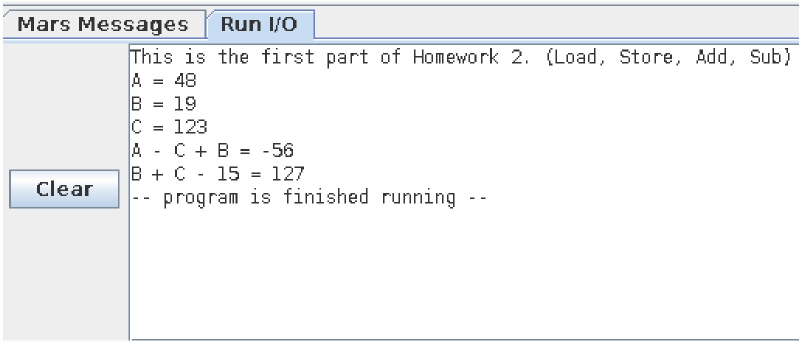
We will give a template called arch_hw2_p1_template.asm, just open it using Mars4_5.jar, write your code within the ####### block in the file (i.e., line 36~41), but DO NOT modify the code elsewhere. Please refer to the following figure.



```
# ====================================================

# The following block helps you practice with the R-type instructions,
# including ADD and SUB, and also LOAD and STORE.
# Here, please read data from 4($gp), 8($gp) and 12($gp),
# store A - C + B  to 16($gp), and
# store B + C - 15 to 4($gp)
####################################################
# @@@ write the code here
```

After you write your code, save it. Next, press “F3” to assemble the code. (Make sure there is no error!) Next, press “F5” to run.

The “Run I/O” screen should show the result like the following figure.



```
This is the first part of Homework 2. (Load, Store, Add, Sub)
A = 48
B = 19
C = 123
A - C + B = -56
B + C - 15 = 127
-- program is finished running --
```

PART II. (Branch Loop, System call, Arithmetic Operations)

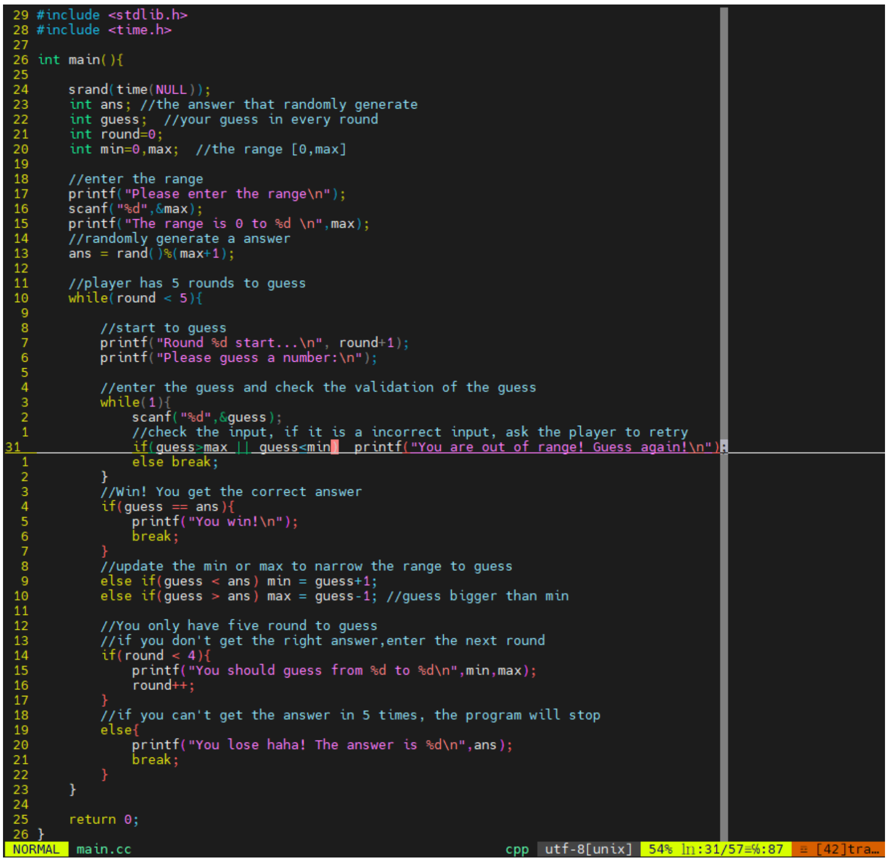
Please convert the following C-like code to MIPS assembly code. Write a new assembly file for this part.

Description: Please design a “Number guessing game” by MIPS.

This is a game to guess the correct number. The program will randomly generate a number, then the player should find the answer in 5 rounds. Every time the player guesses, the range will be narrowed according to the number the player guesses.

You should consider the situation that the player makes a guess that is out of the range. Please ask the player to retry (and it won’t cost any round).

Here is a screenshot of the C code, you can find the completed C code in the attached files (**arch_hw2_p2.c**).



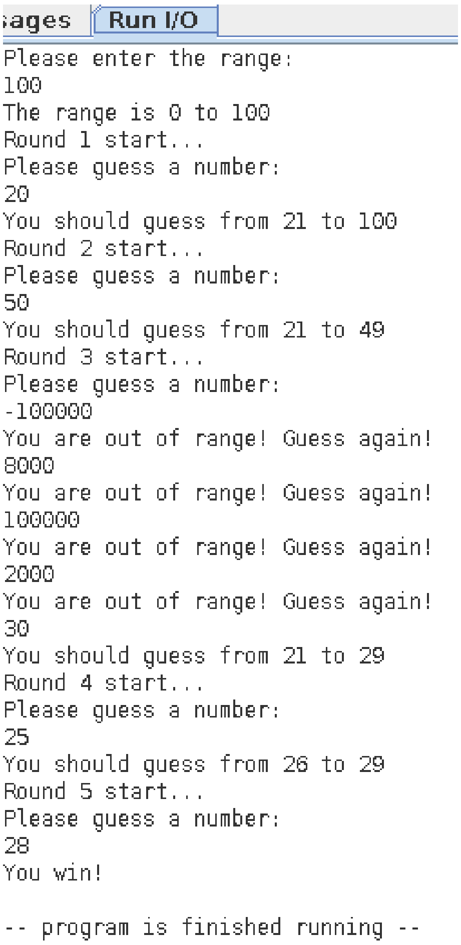
```c
#include <stdlib.h>
#include <time.h>

int main(){

    srand(time(NULL));
    int ans; //the answer that randomly generate
    int guess;  //your guess in every round
    int round=0;
    int min=0,max;  //the range [0,max]

    //enter the range
    printf("Please enter the range\n");
    scanf("%d",&max);
    printf("The range is 0 to %d \n",max);
    //randomly generate a answer
    ans = rand()%(max+1);

    //player has 5 rounds to guess
    while(round < 5){

        //start to guess
        printf("Round %d start...\n", round+1);
        printf("Please guess a number:\n");

        //enter the guess and check the validation of the guess
        while(1){
            scanf("%d",&guess);
            //check the input, if it is a incorrect input, ask the player to retry
            if(guess>max || guess<min)  printf("You are out of range! Guess again!\n");
            else break;
        }
        //Win! You get the correct answer
        if(guess == ans){
            printf("You win!\n");
            break;
        }
        //update the min or max to narrow the range to guess
        else if(guess < ans) min = guess+1;
        else if(guess > ans) max = guess-1; //guess bigger than min

        //You only have five round to guess
        //if you don't get the right answer,enter the next round
        if(round < 4){
            printf("You should guess from %d to %d\n",min,max);
            round++;
        }
        //if you can't get the answer in 5 times, the program will stop
        else{
            printf("You lose haha! The answer is %d\n",ans);
            break;
        }
    }

    return 0;
}
```

Hint

a. We will give the C code called arch_hw2_p2.c for reference. We will also give a MIPS template called arch_hw2_p2_template.asm. We strongly recommend you do this part by yourself, or you can refer to the template if you need some help.

b. TA will use other numbers to test if your program is correct

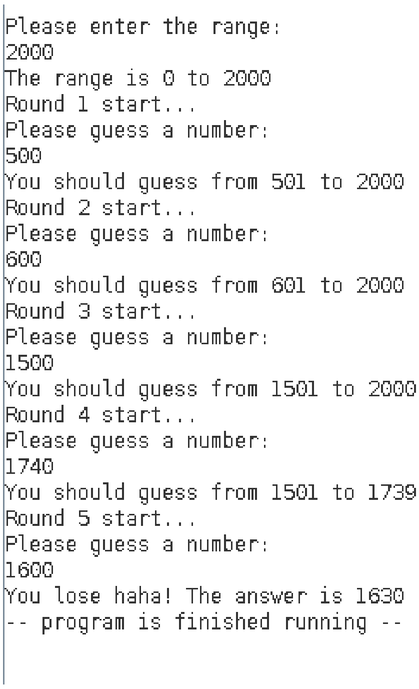
Example



```
Please enter the range:
100
The range is 0 to 100
Round 1 start...
Please guess a number:
20
You should guess from 21 to 100
Round 2 start...
Please guess a number:
50
You should guess from 21 to 49
Round 3 start...
Please guess a number:
-100000
You are out of range! Guess again!
8000
You are out of range! Guess again!
100000
You are out of range! Guess again!
2000
You are out of range! Guess again!
30
You should guess from 21 to 29
Round 4 start...
Please guess a number:
25
You should guess from 26 to 29
Round 5 start...
Please guess a number:
28
You win!

-- program is finished running --
```

First, you should enter the upper bound of range. Here I entered 100, so the range would be 0 to 100. The program will generate an answer, then you can start to guess. If you can guess the correct number in 5 rounds, you will win the game.

If you enter a number that out of range, the program will ask you to try again and again until you make a valid guess.



```
Please enter the range:
2000
The range is 0 to 2000
Round 1 start...
Please guess a number:
500
You should guess from 501 to 2000
Round 2 start...
Please guess a number:
600
You should guess from 601 to 2000
Round 3 start...
Please guess a number:
1500
You should guess from 1501 to 2000
Round 4 start...
Please guess a number:
1740
You should guess from 1501 to 1739
Round 5 start...
Please guess a number:
1600
You lose haha! The answer is 1630
-- program is finished running --
```

In this example, I didn’t get the correct answer, so I lost the game.

Submission (Two assembly programs)

Please name your assembly program with your student ID; for example, arch_hw2_p1_102062801.asm & arch_hw2_p2_102062801.asm, and upload these 2 files onto eeclass. (https://eeclass.nthu.edu.tw/course/9209)

Grading Criteria

Correctness: 80%

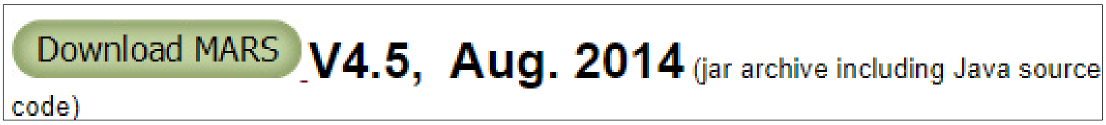
Comments in your code: 10%

Output format: 10%

嚴格禁止抄襲，抄襲者與被抄襲者一律 0 分！

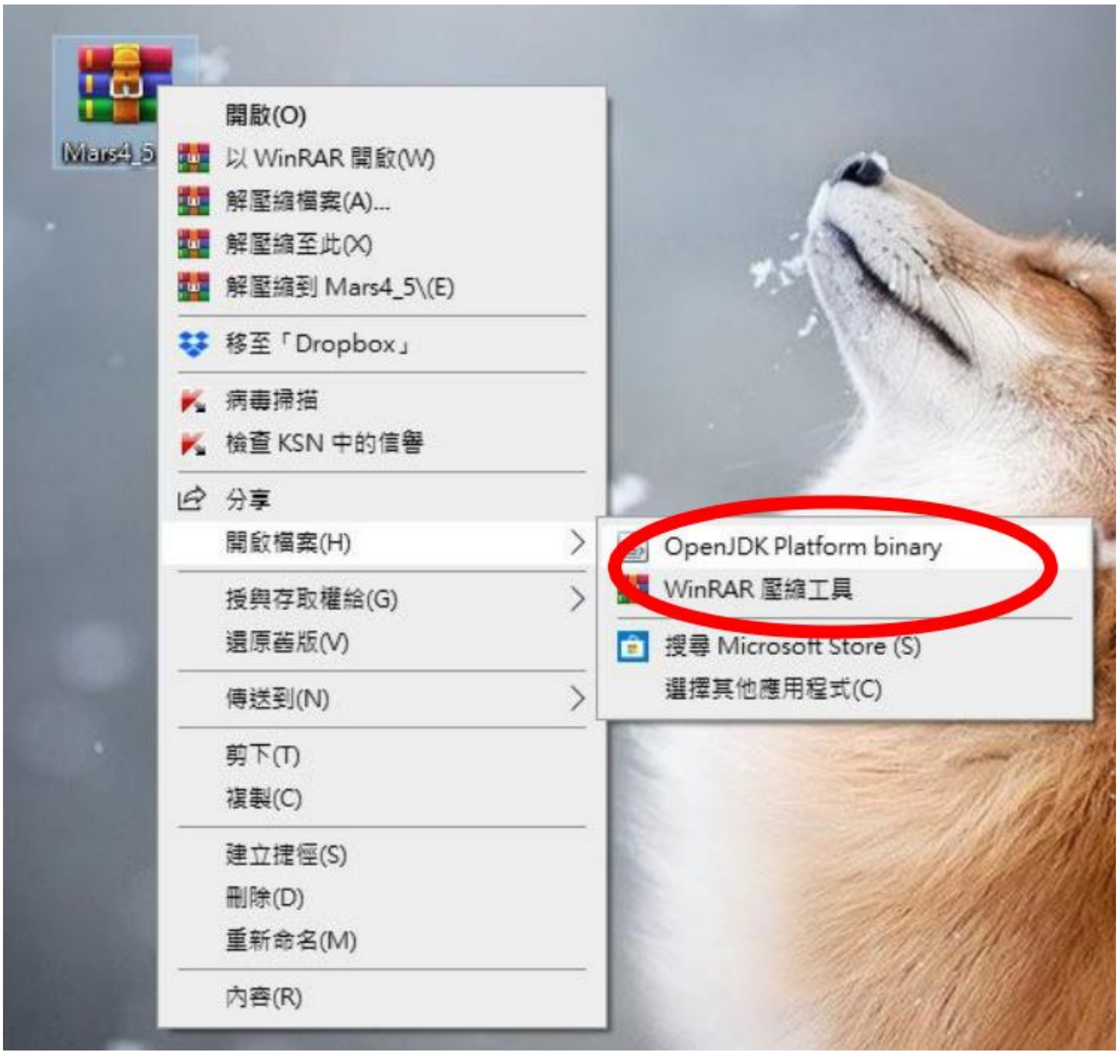
MARS (MIPS Assembler and Runtime Simulator)

1. MARS can assemble and simulate the execution of MIPS assembly language programs. Please refer to the following URL to download Mars4_5.jar: http://courses.missouristate.edu/kenvollmar/mars/download.htm

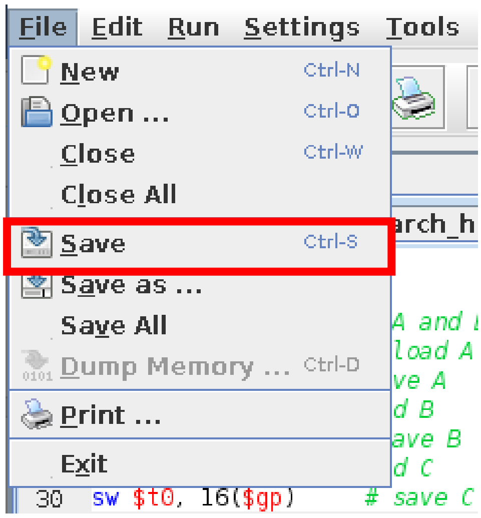


Download MARS V4.5, Aug. 2014 (jar archive including Java source code)

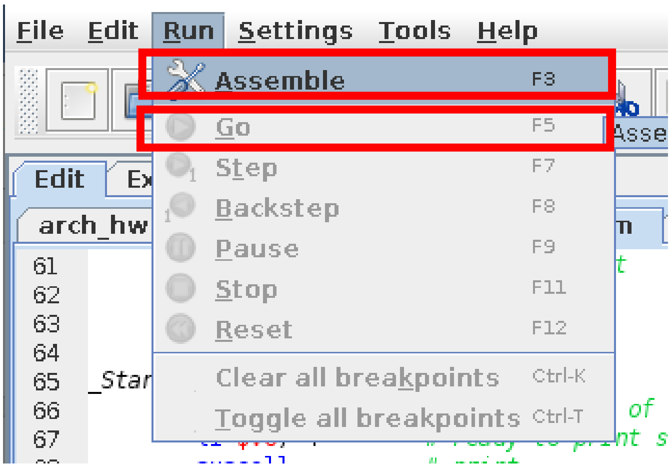
2. MARS is developed with Java language, and it requires JRE (Java Runtime Environment) installed on your computer. Please refer to the following URL to download JRE: https://www.oracle.com/java/technologies/downloads/
3. After you download the MARS, it is a ".jar" file. Please DO NOT decompress it. You can open the MARS by following method.



4. Usage of MARS: P. S. Save your file, Assemble and Go



```
sw $t0, 16($gp)       # save C
```



After saving the file, you can use Assemble button / F3 to assemble your code and use Go button / F5 to execute it.

The result will be shown in the Run I/O window.

5. If you want to execute Mars on CAD server, please use the option -X to run the graphic application.

   ssh -X ic56

   java -jar Mars4_5.jar

Appendix

1. This is a reference for the usage of syscall, you can learn how to do printf, scanf and generate random number in MIPS from this. https://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html

2. Two references for finding the functionality of MIPS instruction.
   a. http://alumni.cs.ucr.edu/~vladimir/cs161/mips.html
   b. https://blog.xuite.net/tzeng015/twblog/113272086-MIPS+%E6%8C%87%E4%BB%A4%E9%9B%86