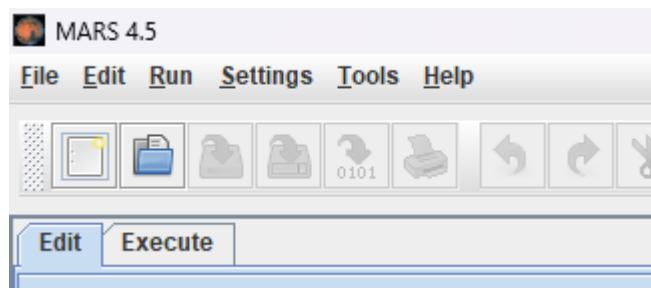
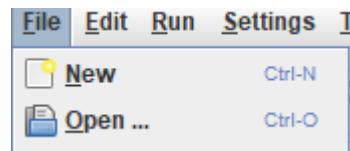


如何运行

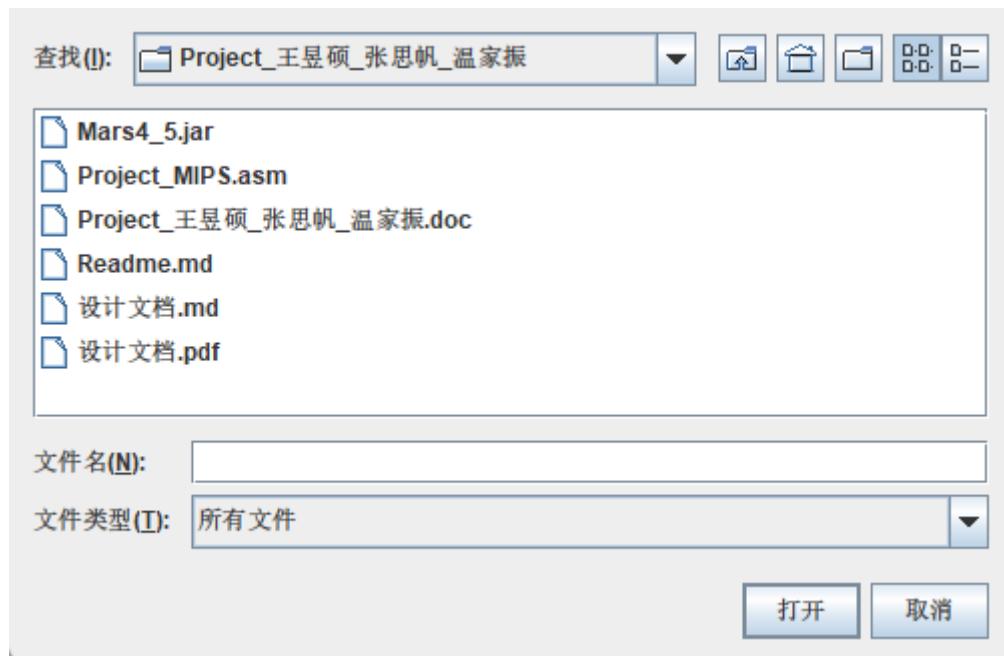
运行 Mars4_5.jar



点击左上角File，再点击Open



选择Project_MIPS.asm，再点击打开



打开后即可见软件代码

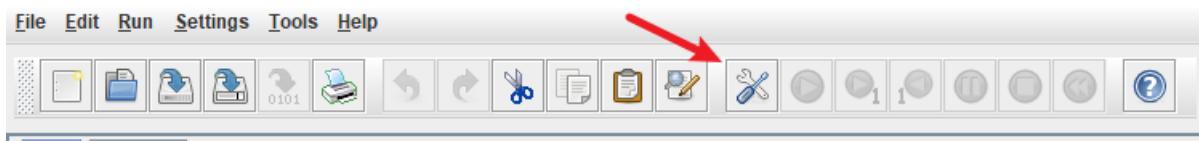
The screenshot shows the QEMU interface with the assembly code for Project_MIPS.sasm. The code includes functions for input conversion and calculation. The Registers window shows various CPU registers with their addresses and values. The Memory dump window shows the memory starting at address 0x10000000.

```

1 .data
2     input:    .space 100
3
4     Curb:      .ascii "==>The binary result is:\0"          # 转换输出
5     Curb:      .ascii "==>The hexadecimal result is:\0"       # 转换输出
6     Ansbt:    .ascii "==>The binary result of calculation is:\0"
7     Ansbt:    .ascii "==>The hexadecimal result of calculation is:\0"
8     MsgInputExpressionWrong: .ascii "syntax error!\n"
9     MsgWelcome: .ascii "Please select a function==\n 1. Conversion\n 2. Calculation\n 3. Exit\n Enter a number(1-3):\0"
10    MsgConversion: .ascii "Please enter the floating point number to be converted:\n"
11    MsgCalculation: .ascii "Please enter an expression, e.g. 1+1:\n"
12    MsgErrorInput: .ascii "Wrong input, please try again!"
13    Newline:   .ascii "\n"
14    output:    .ascii "syntax error!\n"
15
16
17 .text
18
19 main:    #开始执行
20     #欢迎界面(1.转换, 2.计算, 3.退出)
21     li    $a0,  MsgWelcome
22     li    $v0,  4
23     syscall
24     li    $v0,  5

```

再点击左上角该按钮 (Assemble the current file and clear breakpoints)

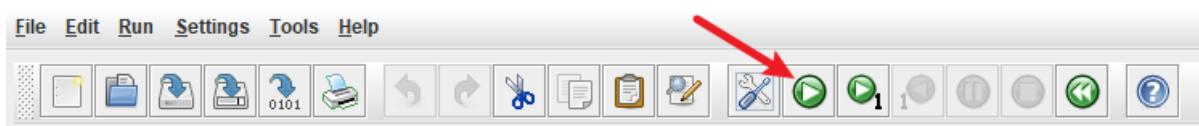


会显示如下界面

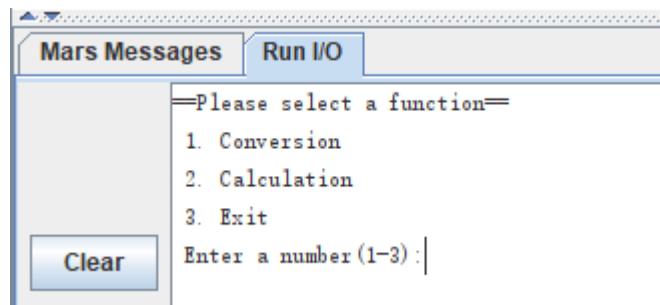
The screenshot shows the QEMU interface after assembly. The assembly code is now in the text segment, and the memory dump shows the assembled binary code. The registers window shows the CPU state after execution.

Blk#	Address	Code	Basic	Source
0x00400000	0x3c010001	lui \$1.0x00001001	21: la \$a0, MsgWelcome	
0x00400004	0x34240103	ori \$4, \$1, 0x00000013		
0x00400008	0x24020004	addi \$2, \$0, 0x00000004	22: li \$v0, 4	
0x0040000c	0x0000000e	syscall	23: syscall	
0x00400010	0x24020005	addi \$2, \$0, 0x00000005	24: li \$v0, 5	
0x00400014	0x0000000e	syscall	25: syscall	调用系统\$0=0读取输入的整数值并存入\$v0
0x00400018	0x24080005	addi \$2, \$0, 0x00000001	27: li \$t0, 1	
0x0040001c	0x10480005	beq \$t2, \$t0, 0x00000005	28: beq \$v0, \$t0, BeginConversion	=跳转到转换
0x00400020	0x24080002	addi \$2, \$0, 0x00000002	29: li \$t0, 2	
0x00400024	0x10480008	beq \$t2, \$t0, 0x00000008	30: beq \$v0, \$t0, BeginCalculation	=跳转到计算
0x00400028	0x24080003	addiu \$2, \$0, 0x00000003	31: li \$t0, 3	
0x0040002c	0x1048000b	beq \$t2, \$t0, 0x0000000b	32: beq \$v0, \$t0, SystemExit	=退出
0x00400030	0x144480068	hme \$2, \$0, 0x00000068	33: hme \$v0, \$t0, ErrorInput	=错误输入
0x00400034	0x3c110001	lui \$1.0x00001001	37: la \$a0, MsgConversion	
0x00400038	0x3424015f	ori \$4, \$1, 0x00000015f		
0x0040003c	0x24020004	addi \$2, \$0, 0x00000004	38: li \$v0, 4	

再点击左上角该按钮 (Run the current program)



控制台成功输出，软件开始运行。



再根据设计文档中的测试部分进行测试即可。