

Name: Phan Trần Thanh Huy

ID: ITCSIU22056

Lab 10

Exercise 1:

Code:

```

1 .data
2     array: .space 32
3     Name:  .ascii "Phan Tran Thanh Huy\UITCSIU22050\
4     comma: .ascii ", "
5     prompt: .ascii "Enter your array: \n"
6     prompt1: .ascii "The final array: "
7 .text
8 .globl main
9 main:
10     la $a0, Name
11     li $v0, 4
12     syscall
13
14     la $s0, array
15     li $t1, 8
16     li $t5, 3
17
18     la $a0, prompt
19     li $v0, 4
20     syscall
21
22 loop:
23     beq $t1, $zero, end
24     li $v0, 5
25     syscall
26     move $t2, $v0 #t2 = input
27
28     div $t3, $t2, $t5 #t3 = t2/3
29     mfhi $t4
30     beq $t4, $zero, divideByThree #if t2%3 == 0 --> divideByThree
31
32     move $a0, $t2
33     jal closestDivideByThree
34     move $t2, $v0
35     j next1
36
37 divideByThree:
38     move $t2, $t3
39 next1:
40     sw $t2, 0($s0)
41     addi $t1, $t1, -1 #size--
42     addi $s0, $s0, 4
43     j loop
44 end:
45     li $t1, 8
46     la $s0, array
47
48
49     la $a0, prompt1
50     li $v0, 4
51     syscall
52
53 print_loop:
54     beq $t1, $zero, end_print
55     lw $a0, 0($s0)
56     li $v0, 1
57     syscall
58
59     la $a0, comma
60     li $v0, 4
61     syscall
62
63     addi $s0, $s0, 4
64     addi $t1, $t1, -1
65     j print_loop
66 end_print:
67
68     li $v0, 10
69     syscall
70
71 closestDivideByThree:
72     subu $sp, $sp, 20
73     sw $t0, 0($sp)
74     sw $t1, 4($sp)
75     sw $t2, 8($sp)
76     sw $t3, 12($sp)
77     sw $t5, 16($sp)
78
79     move $t0, $a0
80     move $t1, $t0
81     move $t2, $t0
82
83 check_loop_1:
84     addi $t1, $t1, 1
85     div $t3, $t1, 3
86     mfhi $t4
87     beq $t4, $zero, satisfied_1
88     j check_loop_1
89 satisfied_1:
90     move $t5, $t1
91
92 check_loop_2:
93     addi $t2, $t2, -1
94     div $t3, $t2, 3
95     mfhi $t4
96     beq $t4, $zero, satisfied_2
97     j check_loop_2
98
99 satisfied_2:
100     move $t6, $t2
101
102     sub $t7, $t5, $t0
103     sub $t8, $t0, $t6
104
105     blt $t7, $t8, else
106     move $v0, $t6
107     j end_check
108 else:
109     move $v0, $t5
110 end_check:
111     lw $t0, 0($sp)
112     lw $t1, 4($sp)
113     lw $t2, 8($sp)
114     lw $t3, 12($sp)
115     lw $t5, 16($sp)
116     addi $sp, $sp, 20
117     jr $ra

```

Test case:

The screenshot shows the Immunity Debugger interface. The assembly window displays the instruction 'array: .space 32' at address 00400070. The CPU registers window shows the value of the 'array' register as 00000000. The 'Registers' window shows the value of the 'array' register as 00000000.

Exercise 2:

Code:

Test case:

FP Regs	Int Regs [16]	Data	Text
nt Regs [16]			
PC	= 400174	[0040000c] 34240054 ori \$4, \$1, 84 [array]	
EPC	= 0	[00400100] 3405000f ori \$5, \$0, 15 ; 78: 14 84	
Cause	= 0	[00400104] 0c10005e jal 0x00400178 [secondLargest], 79:	
BadVAddr	= 0	[00400108] 00024021 addu \$8, \$0, \$2 ; 80: move	
Status	= 3000fff10	[0040010c]	
HI	= 0	[00400110]	
LO	= 5	[00400114] Phan Tran Thanh Huy	
		[00400118] ITC5IU22056	
		[0040011c]	
R0 [r0]	= 0	[00400120]	
R1 [a1]	= 10010000	[00400124]	
R2 [v0]	= a	[00400128]	
R3 [v1]	= 0	[0040012c]	
R4 [a0]	= 10010021	[00400130]	
R5 [a3]	= f	[00400134]	
R6 [a2]	= 7fffffb0	[00400138]	
R7 [a3]	= 0	[0040013c]	
R8 [t0]	= c	[00400140]	
R9 [t1]	= f	[00400144]	
R10 [t2]	= f	[00400148]	
R11 [t3]	= f	[0040014c]	
R12 [t4]	= 10010084	[00400150]	
R13 [t5]	= 3	[00400154]	
R14 [t6]	= c	[00400158]	
R15 [t7]	= 1	[0040015c]	
R16 [s0]	= 1001008c	[00400160]	
R17 [s1]	= c	[00400164]	
R18 [s2]	= f	[00400168]	
R19 [s3]	= ffffffff	[0040016c]	
R20 [s4]	= 0	[00400170]	
R21 [s5]	= 0	[00400174]	
Memory and registers cleared			
SPIM Version 9.1.24 of August 1, 2023 (final) Copyright 1990-2023 by James Larus. All Rights Reserved. SPIM is distributed under a BSD license. See the file README for a full copyright notice. QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.			
.data			
Name: .asciiz "Phan Tran Thanh Huy\nITC5IU22056\n"			
comma: .asciiz ", "			
prompt2: .asciiz "\nThe second max element: "			
prompt3: .asciiz ", found in the index "			
Console			
Enter your array: \n"			
al array: "			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
The final array: 0, 3, 1, 3, 6, 2, 6, 9, 3, 9, 12, 4, 12, 15, 5,			
The second max element: 12, found in the index 11, 13,			
out			
#t3 = t2/3			
three #if t2X3 == 0 --> divideByThree			
move \$a0, \$t2			
jal closestDivideByThree			

Exercise 3:

Code:

```

1 .data
2 Name: .asciiz "phan tran thanh huy\NTCS1022056\vn"
3 array: .word 1, 2, 3, 3, 3, 1, 7, 8, 9, 10
4 tempArr: .space 40
5 duplicateArr: .space 50
6 prompt: .asciiz "Duplicated value: "
7 Duplicated: .asciiz ", repeated "
8 comma: .asciiz " times; "
9 prompt1: .asciiz "Unique values: "
10 comma: .asciiz ", "
11 givenArray: .asciiz "Given array: 1, 2, 3, 3, 3, 1, 7, 8, 9, 10\vn"
12 .text
13 .globl main
14 main:
15     la $a0, Name
16     li $v0, 4
17     syscall
18     la $a0, givenArray
19     li $v0, 4
20     syscall
21
22     la $s0, array
23     la $s1, tempArr
24     la $s4, duplicateArr
25
26     li $t2, 10 #lastIndex = 9
27     li $s3, -1 #i = 0
28
29     la $a0, prompt
30     li $v0, 4
31     syscall
32
33 loop:
34     addi $s3, $s3, 1 #i++
35     beq $s3, $t2, endLoop
36
37     sll $t3, $s3, 2
38     add $t4, $s0, $t3
39     lw $t1, 0($t4) # t1 = array[i]
40
41     li $t5, 0 #j = 0
42     li $t0, 0 #count = 0
43     loop2:
44         beq $t5, $t2, endSubloop
45         sll $t3, $t5, 2
46         add $t4, $s0, $t3
47         lw $t2, 0($t4) # t2 = array[j]
48
49         bne $t1, $t2, next2
50         addi $t0, $t0, 1 #count++
51     next2:
52         addi $t5, $t5, 1 #j++
53         j loop2
54     endSubloop:
55     addi $t6, $t0, -1
56     beq $t6, $zero, unique #check count == 1 or not
57
58     move $a0, $t1
59     jal checkInsideArray #check if that value is already in duplicate array or not
60     beq $v0, $zero, printDuplicate
61     j loop
62
63 printDuplicate:
64     sw $t1, 0($s4)
65     addi $s4, $s4, 4
66
67     move $a0, $t1
68     li $v0, 1
69     syscall
70
71     la $a0, Duplicated
72     li $v0, 4
73     syscall
74
75     move $a0, $t0
76     li $v0, 1
77     syscall
78
79     la $a0, comma
80     li $v0, 4
81     syscall
82
83     j loop
84 unique:
85     sw $t1, 0($s1)
86     addi $s1, $s1, 4
87     j loop
88 endLoop:
89
90 #print unique
91     la $a0, prompt1
92     li $v0, 4
93     syscall
94
95     la $t0, tempArr
96     printloop:
97         lw $a0, 0($t0)
98         beq $a0, $zero, endPrint
99         li $v0, 1
100        syscall
101
102        la $a0, comma
103        li $v0, 4
104        syscall
105
106        addi $t0, $t0, 4
107        j printloop
108    endPrint:
109        li $v0, 10
110        syscall
111
112    checkInsideArray:
113        subu $sp, $sp, 4
114        sw $s4, 0($sp)
115
116        la $s4, duplicateArr
117    checkloop:
118        lw $t4, 0($s4)
119        beq $t4, $zero, endCheck
120        beq $a0, $t4, isContain
121        addi $s4, $s4, 4
122        j checkloop
123    isContain:
124        li $v0, 1
125        j endCheck1
126    endCheck1:
127
128    endCheck:
129        li $v0, 0
130    endCheck1:
131        lw $s4, 0($sp)
132        addi $sp, $sp, 4
133        jr $ra

```


Test case:

The screenshot displays the SPIM MIPS simulator interface. The main window is divided into several panes:

- FP Regs:** Shows floating-point registers, all of which are zero.
- Int Regs [16]:** Shows integer registers. Registers R0 through R20 are listed with their current values. For example, R0 [r0] = 0, R1 [a0] = 10010000, and R20 [s4] = 1001007c.
- PC:** Program Counter, currently at 40015c.
- EPC:** Exception Program Counter, currently at 0.
- Cause:** Exception Cause, currently at 0.
- BadVAddr:** Bad Virtual Address, currently at 0.
- Status:** Status register, currently at 3000ff10.
- Memory and registers cleared:** A message indicating that the memory and registers have been cleared.
- SPIM Version 9.1.24 of August 1, 2023 (final):** Copyright 1990-2023 by James Larus. All Rights Reserved. SPIM is distributed under a BSD license. See the file README for a full copyright notice. QSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.
- Text:** A list of assembly instructions with their addresses. For example, 34020004 ori \$2, \$0, 4; 72: li \$t0, 4; 73: syscall; 00082021 addu \$4, \$0, \$8; 75: move \$t0, \$t0; 34020001 ori \$2, \$0, 1; 76: li \$t0, 4.
- Console:** A window showing the output of the program. It displays the prompt "Phan Tran Thanh Huy", the array "Given array: 1, 2, 3, 3, 3, 1, 7, 8, 9, 10", the duplicated value "Duplicated value: 1, repeated 2 times: 3, repeated 3 times:", and the unique values "Unique values: 2, 7, 8, 9, 10".
- Assembly Code:** A window showing the assembly code being executed. It includes instructions like "la \$a0, prompt1", "li \$v0, 4", "syscall", "la \$t0, tempArr", "printLoop:", and "j endCheck1".

Exercise 4:

Code:

```

1  //
2  // Name:         Area of a Rectangle (C++)
3  //
4  // Author:        John D. C. (John D. C. (John D. C.))
5  //
6  // Version:      1.0
7  //
8  // Date:         2020-01-01
9  //
10 //
11 //
12 //
13 //
14 //
15 //
16 //
17 //
18 //
19 //
20 //
21 //
22 //
23 //
24 //
25 //
26 //
27 //
28 //
29 //
30 //
31 //
32 //
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //
810 //
811 //
812 //
813 //
814 //
815 //
816 //
817 //
818 //
819 //
820 //
821 //
822 //
823 //
824 //
825 //
826 //
827 //
828 //
829 //
830 //
831 //
832 //
833 //
834 //
835 //
836 //
837 //
838 //
839 //
840 //
841 //
842 //
843 //
844 //
845 //
846 //
847 //
848 //
849 //
850 //
851 //
852 //
853 //
854 //
855 //
856 //
857 //
858 //
859 //
860 //
861 //
862 //
863 //
864 //
865 //
866 //
867 //
868 //
869 //
870 //
871 //
872 //
873 //
874 //
875 //
876 //
877 //
878 //
879 //
880 //
881 //
882 //
883 //
884 //
885 //
886 //
887 //
888 //
889 //
890 //
891 //
892 //
893 //
894 //
895 //
896 //
897 //
898 //
899 //
900 //
901 //
902 //
903 //
904 //
905 //
906 //
907 //
908 //
909 //
910 //
911 //
912 //
913 //
914 //
915 //
916 //
917 //
918 //
919 //
920 //
921 //
922 //
923 //
924 //
925 //
926 //
927 //
928 //
929 //
930 //
931 //
932 //
933 //
934 //
935 //
936 //
937 //
938 //
939 //
940 //
941 //
942 //
943 //
944 //
945 //
946 //
947 //
948 //
949 //
950 //
951 //
952 //
953 //
954 //
955 //
956 //
957 //
958 //
959 //
960 //
961 //
962 //
963 //
964 //
965 //
966 //
967 //
968 //
969 //
970 //
971 //
972 //
973 //
974 //
975 //
976 //
977 //
978 //
979 //
980 //
981 //
982 //
983 //
984 //
985 //
986 //
987 //
988 //
989 //
990 //
991 //
992 //
993 //
994 //
995 //
996 //
997 //
998 //
999 //
1000 //

```

Test case:

```
Int Regs [16]
PC = 4000ac
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
RO [r0] = 0
R1 [at] = 10010000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 100100c5
R5 [a1] = 7fffffa8
R6 [a2] = 7fffffb0
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 1
R17 [s1] = 2
R18 [s2] = 3
R19 [s3] = 4
R20 [s4] = 0
R21 [s5] = 0

Memory and registers cleared
SPIN Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIN is distributed under a BSD license.
See the file README for a full copyright notice.
QSPIN is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

[0040004c] 0000000c syscall ; 41: syscall
[00400050] 34020005 ori $2, $0, 5 ; 43: li $2, 5
[00400054] 0000000c syscall ; 44: syscall
[00400058] 10500004 beq $2, $16, 16 [Rectangle_box-0x004]
[0040005c]
[00400060]
[00400064] Phan Tran Thanh Huy
[00400068] ITCISI22056
[0040006c] 1. Rectangle Box
[00400070] 2. Cube
[00400074] 3. Cylinder
[00400078] 4. Rectangle Pyramid
[0040007c] Enter option: 1
[00400080] Enter length, width and height of rectangle box:
[00400084] 3
[00400088] 5
[0040008c] 1.Volume
[00400090] 2.Surface area
[00400094] Enter option: 1
[00400098] Volume of rectangle box: 60.00000000
[0040009c]
[004000a0]
[004000a4]
[004000a8]
[004000ac]
[004000b0]
[004000b4]
[004000b8]
[004000bc]
[004000c0]
[004000c4]

.data
Name: .asciiz "Phan Tran Thanh Huy\nITCISI22056\n"
prompt_shape:.asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\nEnter option: "
Enter length, width and height of rectangle box: \n
\n1.Volume\n2.Surface area\nEnter option: "
Volume of rectangle box: "
Surface area of rectangle box: "
Enter edge of cube: "
\n1.Volume\n2.Surface area\nEnter option: "
Volume of cube: "
.asciiz "Surface area of cube: "
.asciiz "Enter radius and height of cylinder: \n"
.asciiz "\n1.Volume\n2.Surface area\nEnter option: "
.asciiz "Volume of cylinder: "
.asciiz "Surface area of cylinder: "
.float 3.141592654
.asciiz "Enter length, width and height of rectangle box: "
.asciiz "\n1.Volume\n2.Surface area\nEnter option: "
.asciiz "Volume of rectangle box: "
.asciiz "Surface area of rectangle box: "
```

```
FP Regs
Int Regs [16]
PC = 4000ac
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
RO [r0] = 0
R1 [at] = 10010000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 100100c5
R5 [a1] = 7fffffa8
R6 [a2] = 7fffffb0
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 1
R17 [s1] = 2
R18 [s2] = 3
R19 [s3] = 4
R20 [s4] = 0
R21 [s5] = 0

Memory and registers cleared
SPIN Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIN is distributed under a BSD license.
See the file README for a full copyright notice.
QSPIN is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

[00400034] 34110002 ori $17, $0, 2 ; 35: li $17, 2
[00400038] 34120003 ori $18, $0, 3 ; 36: li $18, 3
[0040003c] 34130004 ori $19, $0, 4 ; 37: li $19, 4
[00400040] 3c011001 lui $1, 4097 [prompt_shape]; 39: la $1, prompt_shape
[00400044]
[00400048] Phan Tran Thanh Huy
[00400050] ITCISI22056
[00400054] 1. Rectangle Box
[00400058] 2. Cube
[0040005c] 3. Cylinder
[00400060] 4. Rectangle Pyramid
[00400064] Enter option: 1
[00400068] Enter length, width and height of rectangle box:
[0040006c] 3
[00400070] 5
[00400074] 1.Volume
[00400078] 2.Surface area
[0040007c] Enter option: 2
[00400080] Surface area of rectangle box: 94.00000000
[00400084]
[00400088]
[00400090]
[00400094]
[00400098]
[004000a0]
[004000a4]
[004000a8]
[004000ac]

.data
Name: .asciiz "Phan Tran Thanh Huy\nITCISI22056\n"
prompt_shape:.asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\nEnter option: "
Enter length, width and height of rectangle box: \n
\n1.Volume\n2.Surface area\nEnter option: "
Volume of rectangle box: "
Surface area of rectangle box: "
Enter edge of cube: "
\n1.Volume\n2.Surface area\nEnter option: "
Volume of cube: "
.asciiz "Surface area of cube: "
.asciiz "Enter radius and height of cylinder: \n"
.asciiz "\n1.Volume\n2.Surface area\nEnter option: "
.asciiz "Volume of cylinder: "
.asciiz "Surface area of cylinder: "
.float 3.141592654
.asciiz "Enter length, width and height of rectangle box: "
.asciiz "\n1.Volume\n2.Surface area\nEnter option: "
.asciiz "Volume of rectangle box: "
.asciiz "Surface area of rectangle box: "
```

FP RegsInt Regs [16]

X

Text

PC = 4000ac

EPC = 0

Cause = 0

BadVAddr = 0

Status = 3000ff10

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 10010000

R2 [v0] = a

R3 [v1] = 0

R4 [a0] = 1001013c

R5 [a1] = 7ffff1a8

R6 [a2] = 7ffff1b0

R7 [a3] = 0

R8 [t0] = 0

R9 [t1] = 0

R10 [t2] = 0

R11 [t3] = 0

R12 [t4] = 0

R13 [t5] = 0

R14 [t6] = 0

R15 [t7] = 0

R16 [s0] = 1

R17 [s1] = 2

R18 [s2] = 3

R19 [s3] = 4

R20 [s4] = 0

R21 [s5] = 0

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

li \$s0, 1

#option 1

.data

Name: .asciiz "Phan Tran Thanh Huy\ntICSUI22056\n"

prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n\nEnter option: "

Enter length, width and height of rectangle box: \n

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of rectangle box: "

"Surface area of rectangle box: "

Enter edge of cube: "

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of cube: "

asciiz "Surface area of cube: "

asciiz "Enter radius and height of cylinder: \n"

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of cylinder: "

asciiz "Surface area of cylinder: "

float 3.141592654

asciiz "Enter length, width and height of rectangle box: "

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of rectangle box: "

asciiz "Surface area of rectangle box: "

FP RegsInt Regs [16]

X

Text

PC = 4000ac

EPC = 0

Cause = 0

BadVAddr = 0

Status = 3000ff10

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 40c00000

R2 [v0] = a

R3 [v1] = 0

R4 [a0] = 1001014d

R5 [a1] = 7ffff1a8

R6 [a2] = 7ffff1b0

R7 [a3] = 0

R8 [t0] = 0

R9 [t1] = 0

R10 [t2] = 0

R11 [t3] = 0

R12 [t4] = 0

R13 [t5] = 0

R14 [t6] = 0

R15 [t7] = 0

R16 [s0] = 1

R17 [s1] = 2

R18 [s2] = 3

R19 [s3] = 4

R20 [s4] = 0

R21 [s5] = 0

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

li \$s0, 1

#option 1

.data

Name: .asciiz "Phan Tran Thanh Huy\ntICSUI22056\n"

prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n\nEnter option: "

Enter length, width and height of rectangle box: \n

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of rectangle box: "

"Surface area of rectangle box: "

Enter edge of cube: "

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of cube: "

asciiz "Surface area of cube: "

asciiz "Enter radius and height of cylinder: \n"

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of cylinder: "

asciiz "Surface area of cylinder: "

float 3.141592654

asciiz "Enter length, width and height of rectangle box: "

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of rectangle box: "

asciiz "Surface area of rectangle box: "

FP RegsInt Regs [16]

X

Text

PC = 4000ac

EPC = 0

Cause = 0

BadVAddr = 0

Status = 3000ff10

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 10010000

R2 [v0] = a

R3 [v1] = 0

R4 [a0] = 100101b3

R5 [a1] = 7ffff1a8

R6 [a2] = 7ffff1b0

R7 [a3] = 0

R8 [t0] = 0

R9 [t1] = 0

R10 [t2] = 0

R11 [t3] = 0

R12 [t4] = 0

R13 [t5] = 0

R14 [t6] = 0

R15 [t7] = 0

R16 [s0] = 1

R17 [s1] = 2

R18 [s2] = 3

R19 [s3] = 4

R20 [s4] = 0

R21 [s5] = 0

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

li \$s0, 1

#option 1

.data

Name: .asciiz "Phan Tran Thanh Huy\ntICSUI22056\n"

prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n\nEnter option: "

Enter length, width and height of rectangle box: \n

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of rectangle box: "

"Surface area of rectangle box: "

Enter edge of cube: "

\n1.Volume\n2.Surface area\n\nEnter option: "

"Volume of cube: "

asciiz "Surface area of cube: "

asciiz "Enter radius and height of cylinder: \n"

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of cylinder: "

asciiz "Surface area of cylinder: "

float 3.141592654

asciiz "Enter length, width and height of rectangle box: "

asciiz "\n1.Volume\n2.Surface area\n\nEnter option: "

asciiz "Volume of rectangle box: "

asciiz "Surface area of rectangle box: "

FP Regs Int Regs [16] # X Text .data

Int Regs [16] # X Text

PC = 4000ac
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000fff0
HI = 0
LO = 0
RO [r0] = 0
R1 [a1] = 10010000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 100101c8
R5 [a1] = 7fffffa8
R6 [a2] = 7fffffb0
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [a0] = 1
R17 [a1] = 2
R18 [a2] = 3
R19 [a3] = 4
R20 [a4] = 0
R21 [a5] = 0

Memory and registers cleared

SPIN Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIN is distributed under a BSD license.
See the file README for a full copyright notice.
QcSPIN is linked to the Qc library, which is distributed under the GNU Lesser General Public License version 3 and verson 2.1.

00000034 34110002 ori \$17, \$0, 2 ; 35: 11 01
00000038 34120003 ori \$18, \$0, 3 ; 36: 11 01
0000003c 34130004 ori \$19, \$0, 4 ; 37: 11 01
00000040 3c011001 lui \$1, 4097 [prompt_shape]; 39: 1a
00000044
0000004c Phan Tran Thanh Huy
00000050 ITC2IU22056
00000054 1. Rectangle Box
00000058 2. Cube
0000005c 3. Cylinder
00000060 4. Rectangle Pyramid
00000064 Enter option: 3
00000068 Enter radius and height of cylinder:
0000006c 2
00000070 5
00000074 1.Volume
00000078 2.Surface area
0000007c Enter option: 2
00000080 Surface area of cylinder: 87.96455961
00000084
00000088
0000008c
00000090
00000094
00000098
0000009c
000000a0
000000a4
000000a8
000000ac

li \$s0, 1 #option 1

.data
Name: .asciiz "Phan Tran Thanh Huy\nITC2IU22056\n"
prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n"\nEnter option: "
"Enter length, width and height of rectangle box: \n"
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of rectangle box: "
"Surface area of rectangle box: "
"Enter edge of cube: "
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of cube: "
".asciiz "Surface area of cube: "
".asciiz "Enter radius and height of cylinder: \n"
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of cylinder: "
".asciiz "Surface area of cylinder: "
float 3.141592654
".asciiz "Enter length, width and height of rectangle box: "
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of rectangle box: "
".asciiz "Surface area of rectangle box: "

FP Regs Int Regs [16] # X Text .data

Int Regs [16] # X Text

PC = 4000ac
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000fff0
HI = 0
LO = 0
RO [r0] = 0
R1 [a1] = 40400000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 10010243
R5 [a1] = 7fffffa8
R6 [a2] = 7fffffb0
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [a0] = 1
R17 [a1] = 2
R18 [a2] = 3
R19 [a3] = 4
R20 [a4] = 0
R21 [a5] = 0

Memory and registers cleared

SPIN Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIN is distributed under a BSD license.
See the file README for a full copyright notice.
QcSPIN is linked to the Qc library, which is distributed under the GNU Lesser General Public License version 3 and verson 2.1.

00000034 34110002 ori \$17, \$0, 2 ; 35: 11 01
00000038 34120003 ori \$18, \$0, 3 ; 36: 11 01
0000003c 34130004 ori \$19, \$0, 4 ; 37: 11 01
00000040 3c011001 lui \$1, 4097 [prompt_shape]; 39: 1a
00000044
0000004c Phan Tran Thanh Huy
00000050 ITC2IU22056
00000054 1. Rectangle Box
00000058 2. Cube
0000005c 3. Cylinder
00000060 4. Rectangle Pyramid
00000064 Enter option: 4
00000068 Enter length, width and height of rectangle box:
0000006c 8
00000070 4
00000074 1.Volume
00000078 2.Surface area
0000007c Enter option: 1
00000080 Volume of rectangle box: 64.00000000
00000084
00000088
0000008c
00000090
00000094
00000098
0000009c
000000a0
000000a4
000000a8
000000ac

li \$s0, 1 #option 1

.data
Name: .asciiz "Phan Tran Thanh Huy\nITC2IU22056\n"
prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n"\nEnter option: "
"Enter length, width and height of rectangle box: \n"
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of rectangle box: "
"Surface area of rectangle box: "
"Enter edge of cube: "
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of cube: "
".asciiz "Surface area of cube: "
".asciiz "Enter radius and height of cylinder: \n"
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of cylinder: "
".asciiz "Surface area of cylinder: "
float 3.141592654
".asciiz "Enter length, width and height of rectangle box: "
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of rectangle box: "
".asciiz "Surface area of rectangle box: "

FP Regs Int Regs [16] # X Text .data

Int Regs [16] # X Text

PC = 4000ac
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000fff0
HI = 0
LO = 0
RO [r0] = 0
R1 [a1] = 40000000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 10010254
R5 [a1] = 7fffffa8
R6 [a2] = 7fffffb0
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [a0] = 1
R17 [a1] = 2
R18 [a2] = 3
R19 [a3] = 4
R20 [a4] = 0
R21 [a5] = 0

Memory and registers cleared

SPIN Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIN is distributed under a BSD license.
See the file README for a full copyright notice.
QcSPIN is linked to the Qc library, which is distributed under the GNU Lesser General Public License version 3 and verson 2.1.

00000034 34110002 ori \$17, \$0, 2 ; 35: 11 01
00000038 34120003 ori \$18, \$0, 3 ; 36: 11 01
0000003c 34130004 ori \$19, \$0, 4 ; 37: 11 01
00000040 3c011001 lui \$1, 4097 [prompt_shape]; 39: 1a
00000044
0000004c Phan Tran Thanh Huy
00000050 ITC2IU22056
00000054 1. Rectangle Box
00000058 2. Cube
0000005c 3. Cylinder
00000060 4. Rectangle Pyramid
00000064 Enter option: 4
00000068 Enter length, width and height of rectangle box:
0000006c 8
00000070 4
00000074 1.Volume
00000078 2.Surface area
0000007c Enter option: 2
00000080 Surface area of rectangle box: 121.94112396
00000084
00000088
0000008c
00000090
00000094
00000098
0000009c
000000a0
000000a4
000000a8
000000ac

li \$s0, 1 #option 1

.data
Name: .asciiz "Phan Tran Thanh Huy\nITC2IU22056\n"
prompt_shape: .asciiz "1. Rectangle Box\n2. Cube\n3. Cylinder\n4. Rectangle Pyramid\n"\nEnter option: "
"Enter length, width and height of rectangle box: \n"
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of rectangle box: "
"Surface area of rectangle box: "
"Enter edge of cube: "
"\n1.Volume\n2.Surface area\n"\nEnter option: "
"Volume of cube: "
".asciiz "Surface area of cube: "
".asciiz "Enter radius and height of cylinder: \n"
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of cylinder: "
".asciiz "Surface area of cylinder: "
float 3.141592654
".asciiz "Enter length, width and height of rectangle box: "
".asciiz "\n1.Volume\n2.Surface area\n"\nEnter option: "
".asciiz "Volume of rectangle box: "
".asciiz "Surface area of rectangle box: "

Exercise 5:

Code:

```

1  .data
2  Name: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3  e: .word 6
4  prompt: .asciiz "Enter a,b,c,d,u and v:\n"
5  printResult: .asciiz "The result is: "
6  .text
7  .globl main
8  main:
9      la $a0, Name
10     li $v0, 4
11     syscall
12
13     la $a0, prompt
14     li $v0, 4
15     syscall
16
17     li $v0, 6
18     syscall
19     mov.s $f5, $f0    #f5 = a
20
21     li $v0, 6
22     syscall
23     mov.s $f4, $f0    #f4 = b
24
25     li $v0, 6
26     syscall
27     mov.s $f3, $f0    #f3 = c
28
29     li $v0, 6
30     syscall
31     mov.s $f2, $f0    #f2 = d
32
33     li $v0, 6
34     syscall
35     mov.s $f1, $f0    #f1 = u
36
37     li $v0, 6
38     syscall           #f0 = v
39
40     mov.s $f12, $f1
41     jal calculateIntegral
42     mov.s $f6, $f12    #F(u)
43
44     mov.s $f12, $f0
45     jal calculateIntegral
46     mov.s $f7, $f12    #F(v)
47
48     sub.s $f6, $f6, $f7    #F(u) - F(v)
49
50     lw $s0, e
51     mtc1 $s0, $f8
52     cvt.s.w $f8, $f8
53
54     mul.s $f8, $f8, $f8    #e^2
55
56     div.s $f6, $f6, $f8    #(F(u) - F(v))/e^2
57
58     la $a0, printResult
59     li $v0, 4
60     syscall
61
62     mov.s $f12, $f6
63     li $v0, 2
64     syscall
65
66     li $v0, 10
67     syscall
68
69
70 calculateIntegral:
71     subu $sp, $sp, 8
72     s.s $f6, 0($sp)
73     s.s $f7, 4($sp)
74
75     li.s $f6, 5.0
76     mul.s $f7, $f12, $f12    #x*x
77     mul.s $f7, $f7, $f7    #x^4
78     mul.s $f7, $f7, $f12    #x^5
79     mul.s $f7, $f7, $f5    #ax^5
80     div.s $f7, $f7, $f6    #ax^5 / 5
81
82
83     li.s $f6, 4.0
84     mul.s $f8, $f12, $f12    #x^2
85     mul.s $f8, $f8, $f8    #x^4
86     mul.s $f8, $f8, $f4    #bx^4
87     div.s $f8, $f8, $f6    #bx^4 / 4
88
89     li.s $f6, 3.0
90     mul.s $f9, $f12, $f12    #x^2
91     mul.s $f9, $f9, $f12    #x^3
92     mul.s $f9, $f9, $f3    #cx^3
93     div.s $f9, $f9, $f6    #cx^3 / 3
94
95     mul.s $f10, $f12, $f2    #dx
96
97     add.s $f10, $f10, $f9
98     add.s $f10, $f10, $f8
99     add.s $f10, $f10, $f7
100    mov.s $f12, $f10
101
102    l.s $f6, 0($sp)
103    l.s $f7, 4($sp)
104    addi $sp, $sp, 8
105    jr $ra

```


Test case:

The screenshot displays the SPIN debugger interface with the following components:

- Registers Panel (Left):** Shows the state of 32 MIPS registers. The PC is at 4000c8. R16 (a0) is 10010028. R17 (a1) is 7ffff1a8. R18 (a2) is 7ffff1b0. R19 (a3) is 6. R20 (s4) is 0. R21 (s5) is 0.
- Memory Panel (Bottom Left):** Displays memory contents and registers cleared.
- Assembly View (Top Right):** Shows MIPS assembly code for the function `calculateIntegral`. The code includes instructions like `li $v0, 6`, `syscall`, `mov.s $f12, $f1`, `jal calculateIntegral`, and `mov.s $f6, $f12`.
- Console (Center):** A window titled "Console" showing the output of the program. It displays the value of `F(u) - F(v)` as `-37.79722214`.
- Assembly View (Bottom Right):** Shows the assembly code for the function `calculateIntegral` again, including the `syscall` instruction.

Exercise 6:

Code:

```
1  .data
2      Name:  .ascii "Phan Tran Thanh Huy\nITCSIU22056\n"
3      array: .float 1.0, 5.0, 6.0, 10.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0
4      array_prompt: .ascii "The define array: 1.0, 5.0, 6.0, 10.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0\nSum: "
5  .text
6  .globl main
7  main:
8      la $a0, Name
9      li $v0, 4
10     syscall
11
12     la $a0, array_prompt
13     li $v0, 4
14     syscall
15
16     la $a0, array
17     li $a1, 10      #size = 10
18     jal sum
19     li $v0, 2
20     syscall
21
22     li $v0, 10
23     syscall
24
25 sum:
26     addi $t2, $a1, -1
27     l.s $f12, 0($a0)
28     beq $t2, $zero, return      #if k == 1
29
30     subu $sp, $sp, 12
31     sw $a0, 0($sp) #save array
32     sw $a1, 4($sp) #save size
33     sw $ra, 8($sp)
34
35     addi $a0, $a0, 4
36     addi $a1, $a1, -1
37     jal sum
38
39     lw $a0, 0($sp) #load array
40     lw $a1, 4($sp) #load size
41     lw $ra, 8($sp)
42     addi $sp, $sp, 12
43
44     l.s $f0, 0($a0)
45     add.s $f12, $f0, $f12
46
47     jr $ra
48 return:
49     jr $ra
```

Test case:

The screenshot displays the SPIM MIPS simulator interface. The main window is divided into several panes:

- FP Regs:** Shows floating-point registers, with `PC = 40005c`.
- Int Regs [16]:** Shows integer registers `R0` through `R31`. `R0` through `R31` are all set to `0`.
- Memory:** Shows memory addresses and their contents. The memory is cleared.
- Assembly Code:** Shows the assembly code being executed. The code includes instructions like `lw $4, 0($29)`, `addiu $5, $29, 4`, `addiu $6, $5, 4`, and `addiu $7, $6, 4`.
- Console:** A small window showing the output of the program. It displays the name `Phan Tran Thanh Huy`, the array `array: .float 1.0, 5.0, 6.0, 10.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0`, and the prompt `array_prompt: .asciiz "The define array: 1.0, 5.0, 6.0, 10.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0\nSum: "`.

The console output shows the sum of the array elements: `Sum: 82.00000000`.

Exercise 7:

Code:

```
1  .data
2      Name: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3      array: .float 1.0, 55.0, 6.0, 55.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0
4      array_prompt: .asciiz "The define array: 1.0, 55.0, 6.0, 55.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0 \nMax: "
5  .text
6  .globl main
7  main:
8      la $a0, Name
9      li $v0, 4
10     syscall
11
12     la $a0, array_prompt
13     li $v0, 4
14     syscall
15
16     la $a0, array
17     li $a1, 10      #size = 10
18     jal max
19     li $v0, 2
20     syscall
21
22     li $v0, 10
23     syscall
24
25  max:
26     addi $t2, $a1, -1
27     l.s $f12, 0($a0)
28     beq $t2, $zero, return      #if k == 1
29
30     subu $sp, $sp, 12
31     sw $a0, 0($sp) #save array
32     sw $a1, 4($sp) #save size
33     sw $ra, 8($sp)
34
35     addi $a0, $a0, 4
36     addi $a1, $a1, -1
37     jal max
38
39     lw $a0, 0($sp) #load array
40     lw $a1, 4($sp) #load size
41     lw $ra, 8($sp)
42     addi $sp, $sp, 12
43
44     l.s $f0, 0($a0)      #load v[0]
45
46     c.lt.s $f0, $f12      #if v[0] < temp --> return
47     bclt return
48
49     mov.s $f12, $f0
50     jr $ra
51  return:
52     jr $ra
```

Test case:

The screenshot displays the SPIM MIPS simulator interface. The left pane shows the register file with the following values:

Register	Value
PC	40005c
EPC	0
Cause	0
BadVAddr	0
Status	3000fff10
HI	0
LO	0
R0 [r0]	0
R1 [at]	10010000
R2 [v0]	a
R3 [v1]	0
R4 [a0]	10010024
R5 [a1]	a
R6 [a2]	7ffff1b0
R7 [a3]	0
R8 [t0]	0
R9 [t1]	0
R10 [t2]	0
R11 [t3]	0
R12 [t4]	0
R13 [t5]	0
R14 [t6]	0
R15 [t7]	0
R16 [s0]	0
R17 [s1]	0
R18 [s2]	0
R19 [s3]	0
R20 [s4]	0
R21 [s5]	0

The middle pane shows the assembly code with the following instructions:

```
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $4, 0($29)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $5, $29, 4
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $6, $5, 4
[0040000c] 00041010 ; 186: addiu $7, $6, 4
[00400010] 00c23010 ; 187: addiu $8, $7, 4
[00400014] 0c100010 Phan Tran Thanh Huy
[00400018] 00000010 ITCSIU22056
[0040001c] 34020010 The define array: 1.0, 55.0, 6.0, 55.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0
[00400020] 00000010 Max: 55.00000000
[00400024] 3c041010 ; 188: addiu $9, $8, 4
[00400028] 34020010 ; 189: addiu $10, $9, 4
[0040002c] 00000010 ; 190: addiu $11, $10, 4
[00400030] 3c011010 ; 191: addiu $12, $11, 4
[00400034] 34240010 ; 192: addiu $13, $12, 4
[00400038] 34020010 ; 193: addiu $14, $13, 4
[0040003c] 00000010 ; 194: addiu $15, $14, 4
[00400040] 3c011010 ; 195: addiu $16, $15, 4
[00400044] 34240010 ; 196: addiu $17, $16, 4
[00400048] 34050010 ; 197: addiu $18, $17, 4
[0040004c] 0c100010 ; 198: addiu $19, $18, 4
[00400050] 34020010 ; 199: addiu $20, $19, 4
[00400054] 00000010 ; 200: addiu $21, $20, 4
[00400058] 34020010 ; 201: addiu $22, $21, 4
[0040005c] 00000010 ; 202: addiu $23, $22, 4
[00400060] 20aaffff ; 203: addiu $24, $23, 4
[00400064] c48a0000 ; 204: addiu $25, $24, 4
[00400068] 11400000 ; 205: addiu $26, $25, 4
[0040006c] 27baffff ; 206: addiu $27, $26, 4
[00400070] afa40000 ; 207: addiu $28, $27, 4
[00400074] afa50000 ; 208: addiu $29, $28, 4
```

The right pane shows the data section with the following entries:

```
.data
Name: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
array: .float 1.0, 55.0, 6.0, 55.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0
array_prompt: .asciiz "The define array: 1.0, 55.0, 6.0, 55.0, 2.0, 15.0, 20.0, 22.0, 1.0, 0.0\nMax: "
#if k == 1
```

The bottom pane shows the status bar with the following text:

```
addi $a0, $a0, 4
```