

# CS7381 Project1: Run Tutorial

Name: Bingying Liang

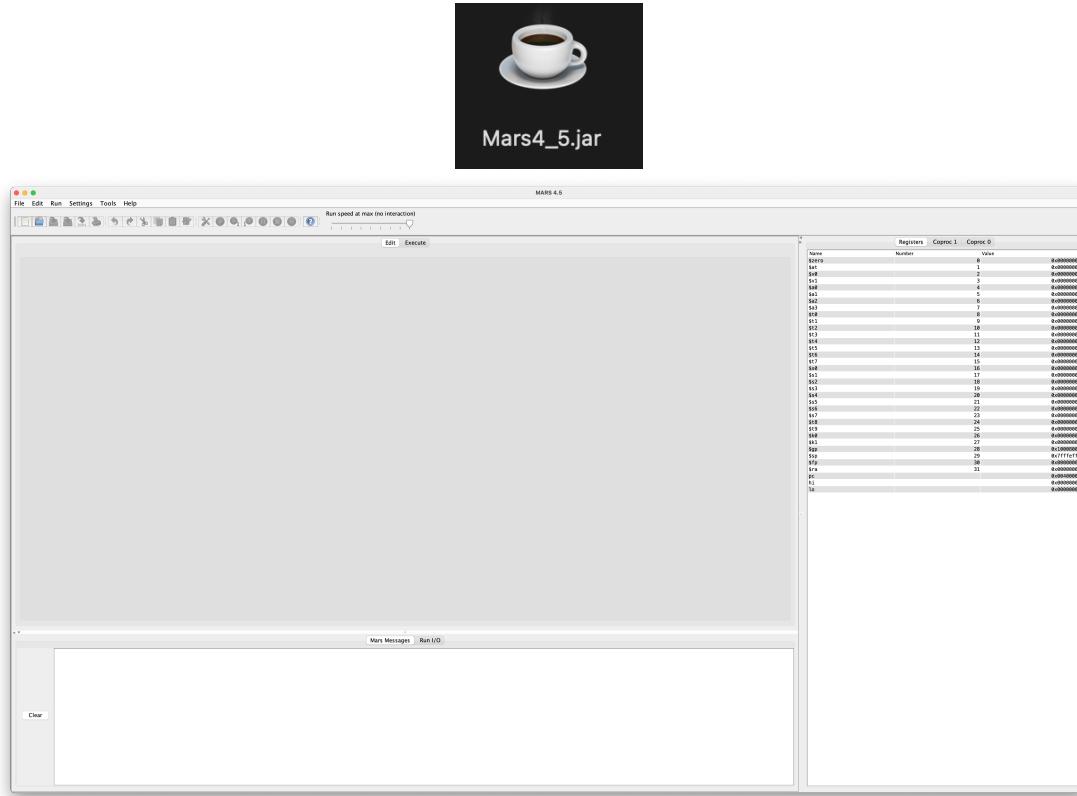
ID: 48999397

Distance

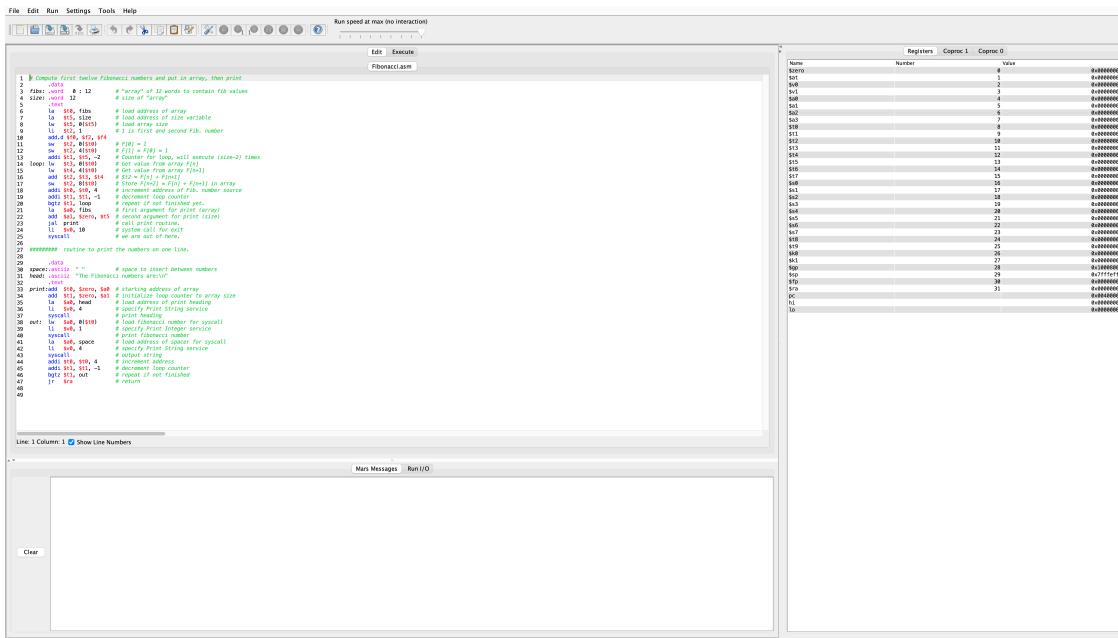
February 17 2023

The example program is **Fibonacci.asm** to compute everyone's favorite number sequence.

1. Start MARS from the Start menu or desktop icon



2. Use the menubar File...Open or the Open icon  to open Fibonacci.asm in the default folder. (All icons have menubar equivalents; the remainder of these steps will use the icon whenever possible.)

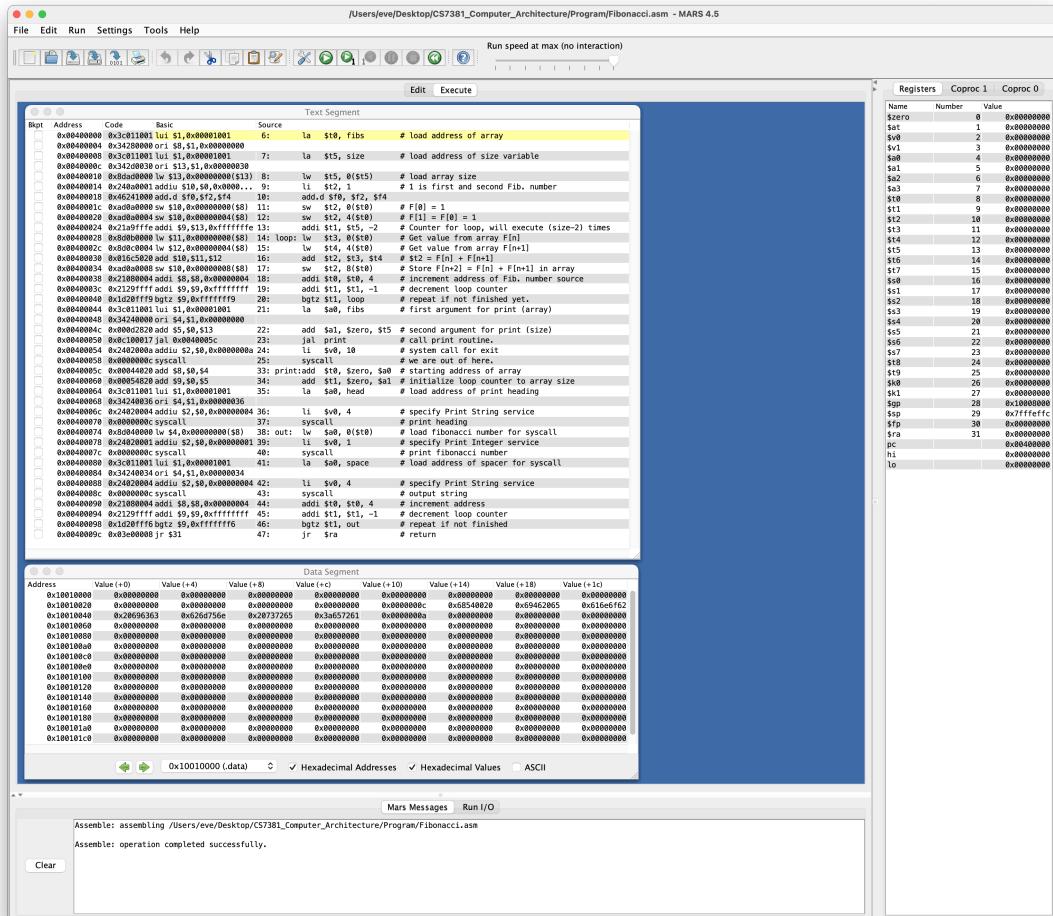


Registers	Coproc 1	Coproc 0
Number		
R00	0	0x00000000
R01	1	0x00000001
R02	2	0x00000002
R03	3	0x00000003
R04	5	0x00000005
R05	8	0x00000008
R06	13	0x0000000D
R07	21	0x00000015
R08	34	0x00000022
R09	55	0x00000035
R10	89	0x00000059
R11	144	0x00000090

```

1  ; Compute first twelve Fibonacci numbers and put in array, then print.
2  Fibo: .word 0, 1, 12      ; "Array" of 12 words to contain fib values
3  .size fibo, 12
4  .text
5  .globl fibo
6  .globl size
7
8  .data
9  fibo: .fill 0, 12
10    .long 0
11    .long 1
12    .long 1
13    .long 2
14    .long 3
15    .long 5
16    .long 8
17    .long 13
18    .long 21
19    .long 34
20    .long 55
21    .long 89
22    .long 144
23
24  size: .word 12
25
26  .text
27  .globl _start
28
29  _start: .short 100
30  .short 40
31  .short 10
32
33  .text
34  .globl fibo
35  .globl head
36  .globl size
37
38  printint: .word 0, 1, 12      ; starting address of array
39  .size printint, 12
40  .text
41  .globl head
42  .globl print
43  .globl size
44
45  .data
46  head: .fill 0, 12
47    .long 0
48    .long 1
49    .long 2
50    .long 3
51    .long 5
52    .long 8
53    .long 13
54    .long 21
55    .long 34
56    .long 55
57    .long 89
58    .long 144
59
60  print: .word 0, 1, 12      ; starting address of print
61  .size print, 12
62
63  .text
64  .globl head
65  .globl print
66  .globl size
67
68  .data
69  .globl fibo
70  .globl head
71  .globl print
72  .globl size
73
74  .text
75  .globl fibo
76  .globl head
77  .globl print
78  .globl size
79
80  .data
81  .globl fibo
82  .globl head
83  .globl print
84  .globl size
85
86  .text
87  .globl fibo
88  .globl head
89  .globl print
90  .globl size
91
92
93  .text
94  .globl fibo
95  .globl head
96  .globl print
97  .globl size
98
99  .text
100 .globl fibo
101 .globl head
102 .globl print
103 .globl size
104
105 .text
106 .globl fibo
107 .globl head
108 .globl print
109 .globl size
110
111 .text
112 .globl fibo
113 .globl head
114 .globl print
115 .globl size
116
117 .text
118 .globl fibo
119 .globl head
120 .globl print
121 .globl size
122
123 .text
124 .globl fibo
125 .globl head
126 .globl print
127 .globl size
128
129 .text
130 .globl fibo
131 .globl head
132 .globl print
133 .globl size
134
135 .text
136 .globl fibo
137 .globl head
138 .globl print
139 .globl size
140
141 .text
142 .globl fibo
143 .globl head
144 .globl print
145 .globl size
146
147 .text
148 .globl fibo
149 .globl head
150 .globl print
151 .globl size
152
153 .text
154 .globl fibo
155 .globl head
156 .globl print
157 .globl size
158
159 .text
160 .globl fibo
161 .globl head
162 .globl print
163 .globl size
164
165 .text
166 .globl fibo
167 .globl head
168 .globl print
169 .globl size
170
171 .text
172 .globl fibo
173 .globl head
174 .globl print
175 .globl size
176
177 .text
178 .globl fibo
179 .globl head
180 .globl print
181 .globl size
182
183 .text
184 .globl fibo
185 .globl head
186 .globl print
187 .globl size
188
189 .text
190 .globl fibo
191 .globl head
192 .globl print
193 .globl size
194
195 .text
196 .globl fibo
197 .globl head
198 .globl print
199 .globl size
200
201 .text
202 .globl fibo
203 .globl head
204 .globl print
205 .globl size
206
207 .text
208 .globl fibo
209 .globl head
210 .globl print
211 .globl size
212
213 .text
214 .globl fibo
215 .globl head
216 .globl print
217 .globl size
218
219 .text
220 .globl fibo
221 .globl head
222 .globl print
223 .globl size
224
225 .text
226 .globl fibo
227 .globl head
228 .globl print
229 .globl size
230
231 .text
232 .globl fibo
233 .globl head
234 .globl print
235 .globl size
236
237 .text
238 .globl fibo
239 .globl head
240 .globl print
241 .globl size
242
243 .text
244 .globl fibo
245 .globl head
246 .globl print
247 .globl size
248
249 .text
250 .globl fibo
251 .globl head
252 .globl print
253 .globl size
254
255 .text
256 .globl fibo
257 .globl head
258 .globl print
259 .globl size
260
261 .text
262 .globl fibo
263 .globl head
264 .globl print
265 .globl size
266
267 .text
268 .globl fibo
269 .globl head
270 .globl print
271 .globl size
272
273 .text
274 .globl fibo
275 .globl head
276 .globl print
277 .globl size
278
279 .text
280 .globl fibo
281 .globl head
282 .globl print
283 .globl size
284
285 .text
286 .globl fibo
287 .globl head
288 .globl print
289 .globl size
290
291 .text
292 .globl fibo
293 .globl head
294 .globl print
295 .globl size
296
297 .text
298 .globl fibo
299 .globl head
300 .globl print
301 .globl size
302
303 .text
304 .globl fibo
305 .globl head
306 .globl print
307 .globl size
308
309 .text
310 .globl fibo
311 .globl head
312 .globl print
313 .globl size
314
315 .text
316 .globl fibo
317 .globl head
318 .globl print
319 .globl size
320
321 .text
322 .globl fibo
323 .globl head
324 .globl print
325 .globl size
326
327 .text
328 .globl fibo
329 .globl head
330 .globl print
331 .globl size
332
333 .text
334 .globl fibo
335 .globl head
336 .globl print
337 .globl size
338
339 .text
340 .globl fibo
341 .globl head
342 .globl print
343 .globl size
344
345 .text
346 .globl fibo
347 .globl head
348 .globl print
349 .globl size
350
351 .text
352 .globl fibo
353 .globl head
354 .globl print
355 .globl size
356
357 .text
358 .globl fibo
359 .globl head
360 .globl print
361 .globl size
362
363 .text
364 .globl fibo
365 .globl head
366 .globl print
367 .globl size
368
369 .text
370 .globl fibo
371 .globl head
372 .globl print
373 .globl size
374
375 .text
376 .globl fibo
377 .globl head
378 .globl print
379 .globl size
380
381 .text
382 .globl fibo
383 .globl head
384 .globl print
385 .globl size
386
387 .text
388 .globl fibo
389 .globl head
390 .globl print
391 .globl size
392
393 .text
394 .globl fibo
395 .globl head
396 .globl print
397 .globl size
398
399 .text
400 .globl fibo
401 .globl head
402 .globl print
403 .globl size
404
405 .text
406 .globl fibo
407 .globl head
408 .globl print
409 .globl size
410
411 .text
412 .globl fibo
413 .globl head
414 .globl print
415 .globl size
416
417 .text
418 .globl fibo
419 .globl head
420 .globl print
421 .globl size
422
423 .text
424 .globl fibo
425 .globl head
426 .globl print
427 .globl size
428
429 .text
430 .globl fibo
431 .globl head
432 .globl print
433 .globl size
434
435 .text
436 .globl fibo
437 .globl head
438 .globl print
439 .globl size
440
441 .text
442 .globl fibo
443 .globl head
444 .globl print
445 .globl size
446
447 .text
448 .globl fibo
449 .globl head
450 .globl print
451 .globl size
452
453 .text
454 .globl fibo
455 .globl head
456 .globl print
457 .globl size
458
459 .text
460 .globl fibo
461 .globl head
462 .globl print
463 .globl size
464
465 .text
466 .globl fibo
467 .globl head
468 .globl print
469 .globl size
470
471 .text
472 .globl fibo
473 .globl head
474 .globl print
475 .globl size
476
477 .text
478 .globl fibo
479 .globl head
480 .globl print
481 .globl size
482
483 .text
484 .globl fibo
485 .globl head
486 .globl print
487 .globl size
488
489 .text
490 .globl fibo
491 .globl head
492 .globl print
493 .globl size
494
495 .text
496 .globl fibo
497 .globl head
498 .globl print
499 .globl size
500
501 .text
502 .globl fibo
503 .globl head
504 .globl print
505 .globl size
506
507 .text
508 .globl fibo
509 .globl head
510 .globl print
511 .globl size
512
513 .text
514 .globl fibo
515 .globl head
516 .globl print
517 .globl size
518
519 .text
520 .globl fibo
521 .globl head
522 .globl print
523 .globl size
524
525 .text
526 .globl fibo
527 .globl head
528 .globl print
529 .globl size
530
531 .text
532 .globl fibo
533 .globl head
534 .globl print
535 .globl size
536
537 .text
538 .globl fibo
539 .globl head
540 .globl print
541 .globl size
542
543 .text
544 .globl fibo
545 .globl head
546 .globl print
547 .globl size
548
549 .text
550 .globl fibo
551 .globl head
552 .globl print
553 .globl size
554
555 .text
556 .globl fibo
557 .globl head
558 .globl print
559 .globl size
560
561 .text
562 .globl fibo
563 .globl head
564 .globl print
565 .globl size
566
567 .text
568 .globl fibo
569 .globl head
570 .globl print
571 .globl size
572
573 .text
574 .globl fibo
575 .globl head
576 .globl print
577 .globl size
578
579 .text
580 .globl fibo
581 .globl head
582 .globl print
583 .globl size
584
585 .text
586 .globl fibo
587 .globl head
588 .globl print
589 .globl size
590
591 .text
592 .globl fibo
593 .globl head
594 .globl print
595 .globl size
596
597 .text
598 .globl fibo
599 .globl head
600 .globl print
601 .globl size
602
603 .text
604 .globl fibo
605 .globl head
606 .globl print
607 .globl size
608
609 .text
610 .globl fibo
611 .globl head
612 .globl print
613 .globl size
614
615 .text
616 .globl fibo
617 .globl head
618 .globl print
619 .globl size
620
621 .text
622 .globl fibo
623 .globl head
624 .globl print
625 .globl size
626
627 .text
628 .globl fibo
629 .globl head
630 .globl print
631 .globl size
632
633 .text
634 .globl fibo
635 .globl head
636 .globl print
637 .globl size
638
639 .text
640 .globl fibo
641 .globl head
642 .globl print
643 .globl size
644
645 .text
646 .globl fibo
647 .globl head
648 .globl print
649 .globl size
650
651 .text
652 .globl fibo
653 .globl head
654 .globl print
655 .globl size
656
657 .text
658 .globl fibo
659 .globl head
660 .globl print
661 .globl size
662
663 .text
664 .globl fibo
665 .globl head
666 .globl print
667 .globl size
668
669 .text
670 .globl fibo
671 .globl head
672 .globl print
673 .globl size
674
675 .text
676 .globl fibo
677 .globl head
678 .globl print
679 .globl size
680
681 .text
682 .globl fibo
683 .globl head
684 .globl print
685 .globl size
686
687 .text
688 .globl fibo
689 .globl head
690 .globl print
691 .globl size
692
693 .text
694 .globl fibo
695 .globl head
696 .globl print
697 .globl size
698
699 .text
700 .globl fibo
701 .globl head
702 .globl print
703 .globl size
704
705 .text
706 .globl fibo
707 .globl head
708 .globl print
709 .globl size
710
711 .text
712 .globl fibo
713 .globl head
714 .globl print
715 .globl size
716
717 .text
718 .globl fibo
719 .globl head
720 .globl print
721 .globl size
722
723 .text
724 .globl fibo
725 .globl head
726 .globl print
727 .globl size
728
729 .text
730 .globl fibo
731 .globl head
732 .globl print
733 .globl size
734
735 .text
736 .globl fibo
737 .globl head
738 .globl print
739 .globl size
740
741 .text
742 .globl fibo
743 .globl head
744 .globl print
745 .globl size
746
747 .text
748 .globl fibo
749 .globl head
750 .globl print
751 .globl size
752
753 .text
754 .globl fibo
755 .globl head
756 .globl print
757 .globl size
758
759 .text
760 .globl fibo
761 .globl head
762 .globl print
763 .globl size
764
765 .text
766 .globl fibo
767 .globl head
768 .globl print
769 .globl size
770
771 .text
772 .globl fibo
773 .globl head
774 .globl print
775 .globl size
776
777 .text
778 .globl fibo
779 .globl head
780 .globl print
781 .globl size
782
783 .text
784 .globl fibo
785 .globl head
786 .globl print
787 .globl size
788
789 .text
790 .globl fibo
791 .globl head
792 .globl print
793 .globl size
794
795 .text
796 .globl fibo
797 .globl head
798 .globl print
799 .globl size
800
801 .text
802 .globl fibo
803 .globl head
804 .globl print
805 .globl size
806
807 .text
808 .globl fibo
809 .globl head
810 .globl print
811 .globl size
812
813 .text
814 .globl fibo
815 .globl head
816 .globl print
817 .globl size
818
819 .text
820 .globl fibo
821 .globl head
822 .globl print
823 .globl size
824
825 .text
826 .globl fibo
827 .globl head
828 .globl print
829 .globl size
830
831 .text
832 .globl fibo
833 .globl head
834 .globl print
835 .globl size
836
837 .text
838 .globl fibo
839 .globl head
840 .globl print
841 .globl size
842
843 .text
844 .globl fibo
845 .globl head
846 .globl print
847 .globl size
848
849 .text
850 .globl fibo
851 .globl head
852 .globl print
853 .globl size
854
855 .text
856 .globl fibo
857 .globl head
858 .globl print
859 .globl size
860
861 .text
862 .globl fibo
863 .globl head
864 .globl print
865 .globl size
866
867 .text
868 .globl fibo
869 .globl head
870 .globl print
871 .globl size
872
873 .text
874 .globl fibo
875 .globl head
876 .globl print
877 .globl size
878
879 .text
880 .globl fibo
881 .globl head
882 .globl print
883 .globl size
884
885 .text
886 .globl fibo
887 .globl head
888 .globl print
889 .globl size
890
891 .text
892 .globl fibo
893 .globl head
894 .globl print
895 .globl size
896
897 .text
898 .globl fibo
899 .globl head
900 .globl print
901 .globl size
902
903 .text
904 .globl fibo
905 .globl head
906 .globl print
907 .globl size
908
909 .text
910 .globl fibo
911 .globl head
912 .globl print
913 .globl size
914
915 .text
916 .globl fibo
917 .globl head
918 .globl print
919 .globl size
920
921 .text
922 .globl fibo
923 .globl head
924 .globl print
925 .globl size
926
927 .text
928 .globl fibo
929 .globl head
930 .globl print
931 .globl size
932
933 .text
934 .globl fibo
935 .globl head
936 .globl print
937 .globl size
938
939 .text
940 .globl fibo
941 .globl head
942 .globl print
943 .globl size
944
945 .text
946 .globl fibo
947 .globl head
948 .globl print
949 .globl size
950
951 .text
952 .globl fibo
953 .globl head
954 .globl print
955 .globl size
956
957 .text
958 .globl fibo
959 .globl head
960 .globl print
961 .globl size
962
963 .text
964 .globl fibo
965 .globl head
966 .globl print
967 .globl size
968
969 .text
970 .globl fibo
971 .globl head
972 .globl print
973 .globl size
974
975 .text
976 .globl fibo
977 .globl head
978 .globl print
979 .globl size
980
981 .text
982 .globl fibo
983 .globl head
984 .globl print
985 .globl size
986
987 .text
988 .globl fibo
989 .globl head
990 .globl print
991 .globl size
992
993 .text
994 .globl fibo
995 .globl head
996 .globl print
997 .globl size
998
999 .text
1000 .globl fibo
1001 .globl head
1002 .globl print
1003 .globl size
1004
1005 .text
1006 .globl fibo
1007 .globl head
1008 .globl print
1009 .globl size
1010
1011 .text
1012 .globl fibo
1013 .globl head
1014 .globl print
1015 .globl size
1016
1017 .text
1018 .globl fibo
1019 .globl head
1020 .globl print
1021 .globl size
1022
1023 .text
1024 .globl fibo
1025 .globl head
1026 .globl print
1027 .globl size
1028
1029 .text
1030 .globl fibo
1031 .globl head
1032 .globl print
1033 .globl size
1034
1035 .text
1036 .globl fibo
1037 .globl head
1038 .globl print
1039 .globl size
1040
1041 .text
1042 .globl fibo
1043 .globl head
1044 .globl print
1045 .globl size
1046
1047 .text
1048 .globl fibo
1049 .globl head
1050 .globl print
1051 .globl size
1052
1053 .text
1054 .globl fibo
1055 .globl head
1056 .globl print
1057 .globl size
1058
1059 .text
1060 .globl fibo
1061 .globl head
1062 .globl print
1063 .globl size
1064
1065 .text
1066 .globl fibo
1067 .globl head
1068 .globl print
1069 .globl size
1070
1071 .text
1072 .globl fibo
1073 .globl head
1074 .globl print
1075 .globl size
1076
1077 .text
1078 .globl fibo
1079 .globl head
1080 .globl print
1081 .globl size
1082
1083 .text
1084 .globl fibo
1085 .globl head
1086 .globl print
1087 .globl size
1088
1089 .text
1090 .globl fibo
1091 .globl head
1092 .globl print
1093 .globl size
1094
1095 .text
1096 .globl fibo
1097 .globl head
1098 .globl print
1099 .globl size
1100
1101 .text
1102 .globl fibo
1103 .globl head
1104 .globl print
1105 .globl size
1106
1107 .text
1108 .globl fibo
1109 .globl head
1110 .globl print
1111 .globl size
1112
1113 .text
1114 .globl fibo
1115 .globl head
1116 .globl print
1117 .globl size
1118
1119 .text
1120 .globl fibo
1121 .globl head
1122 .globl print
1123 .globl size
1124
1125 .text
1126 .globl fibo
1127 .globl head
1128 .globl print
1129 .globl size
1130
1131 .text
1132 .globl fibo
1133 .globl head
1134 .globl print
1135 .globl size
1136
1137 .text
1138 .globl fibo
1139 .globl head
1140 .globl print
1141 .globl size
1142
1143 .text
1144 .globl fibo
1145 .globl head
1146 .globl print
1147 .globl size
1148
1149 .text
1150 .globl fibo
1151 .globl head
1152 .globl print
1153 .globl size
1154
1155 .text
1156 .globl fibo
1157 .globl head
1158 .globl print
1159 .globl size
1160
1161 .text
1162 .globl fibo
1163 .globl head
1164 .globl print
1165 .globl size
1166
1167 .text
1168 .globl fibo
1169 .globl head
1170 .globl print
1171 .globl size
1172
1173 .text
1174 .globl fibo
1175 .globl head
1176 .globl print
1177 .globl size
1178
1179 .text
1180 .globl fibo
1181 .globl head
1182 .globl print
1183 .globl size
1184
1185 .text
1186 .globl fibo
1187 .globl head
1188 .globl print
1189 .globl size
1190
1191 .text
1192 .globl fibo
1193 .globl head
1194 .globl print
1195 .globl size
1196
1197 .text
1198 .globl fibo
1199 .globl head
1200 .globl print
1201 .globl size
1202
1203 .text
1204 .globl fibo
1205 .globl head
1206 .globl print
1207 .globl size
1208
1209 .text
1210 .globl fibo
1211 .globl head
1212 .globl print
1213 .globl size
1214
1215 .text
1216 .globl fibo
1217 .globl head
1218 .globl print
1219 .globl size
1220
1221 .text
1222 .globl fibo
1223 .globl head
1224 .globl print
1225 .globl size
1226
1227 .text
1228 .globl fibo
1229 .globl head
1230 .globl print
1231 .globl size
1232
1233 .text
1234 .globl fibo
1235 .globl head
1236 .globl print
1237 .globl size
1238
1239 .text
1240 .globl fibo
1241 .globl head
1242 .globl print
1243 .globl size
1244
1245 .text
1246 .globl fibo
1247 .globl head
1248 .globl print
1249 .globl size
1250
1251 .text
1252 .globl fibo
1253 .globl head
1254 .globl print
1255 .globl size
1256
1257 .text
1258 .globl fibo
1259 .globl head
1260 .globl print
1261 .globl size
1262
1263 .text
1264 .globl fibo
1265 .globl head
1266 .globl print
1267 .globl size
1268
1269 .text
1270 .globl fibo
1271 .globl head
1272 .globl print
1273 .globl size
1274
1275 .text
1276 .globl fibo
1277 .globl head
1278 .globl print
1279 .globl size
1280
1281 .text
1282 .globl fibo
1283 .globl head
1284 .globl print
1285 .globl size
1286
1287 .text
1288 .globl fibo
1289 .globl head
1290 .globl print
1291 .globl size
1292
1293 .text
1294 .globl fibo
1295 .globl head
1296 .globl print
1297 .globl size
1298
1299 .text
1300 .globl fibo
1301 .globl head
1302 .globl print
1303 .globl size
1304
1305 .text
1306 .globl fibo
1307 .globl head
1308 .globl print
1309 .globl size
1310
1311 .text
1312 .globl fibo
1313 .globl head
1314 .globl print
1315 .globl size
1316
1317 .text
1318 .globl fibo
1319 .globl head
1320 .globl print
1321 .globl size
1322
1323 .text
1324 .globl fibo
1325 .globl head
1326 .globl print
1327 .globl size
1328
1329 .text
1330 .globl fibo
1331 .globl head
1332 .globl print
1333 .globl size
1334
1335 .text
1336 .globl fibo
1337 .globl head
1338 .globl print
1339 .globl size
1340
1341 .text
1342 .globl fibo
1343 .globl head
1344 .globl print
1345 .globl size
1346
1347 .text
1348 .globl fibo
1349 .globl head
1350 .globl print
1351 .globl size
1352
1353 .text
1354 .globl fibo
1355 .globl head
1356 .globl print
1357 .globl size
1358
1359 .text
1360 .globl fibo
1361 .globl head
1362 .globl print
1363 .globl size
1364
1365 .text
1366 .globl fibo
1367 .globl head
1368 .globl print
1369 .globl size
1370
1371 .text
1372 .globl fibo
1373 .globl head
1374 .globl print
1375 .globl size
1376
1377 .text
1378 .globl fibo
1379 .globl head
1380 .globl print
1381 .globl size
1382
1383 .text
1384 .globl fibo
1385 .globl head
1386 .globl print
1387 .globl size
1388
1389 .text
1390 .globl fibo
1391 .globl head
1392 .globl print
1393 .globl size
1394
1395 .text
1396 .globl fibo
1397 .globl head
1398 .globl print
1399 .globl size
1400
1401 .text
1402 .globl fibo
1403 .globl head
1404 .globl print
1405 .globl size
1406
1407 .text
1408 .globl fibo
1409 .globl head
1410 .globl print
1411 .globl size
1412
1413 .text
1414 .globl fibo
1415 .globl head
1416 .globl print
1417 .globl size
1418
1419 .text
1420 .globl fibo
1421 .globl head
1422 .globl print
1423 .globl size
1424
142
```

3. The provided assembly program is complete. Assemble the program using the icon



4. Identify the location and values of the program's initialized data. Use the checkbox to toggle the display format between decimal and hexadecimal  **Hexadecimal Values**

- The twelve-element array `fibs` is initialized to zero, at addresses 0x10010000 ...0x1001002c.

- The data location size has value  $12_{10}(C_{16})$  at 0x10010030. Use the checkbox to toggle the display format between decimal and hexadecimal  **Hexadecimal Values**.

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000c	0x68540020	0x69462065	0x616e6f62
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000a	0x00000000	0x00000000	0x00000000
0x10010040	0x20696363	0x626d756e	0x20737265	0x3a657261	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	12	1750335520	1766203493	1634627426
0x10010040	543777635	1651340654	544436837	979726945	10	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0
0x100101a0	0	0	0	0	0	0	0	0
0x100101c0	0	0	0	0	0	0	0	0

5. Locate the Registers display, which shows the 32 common MIPS registers. Other tabs in the Registers display show the floating-point registers (Coproc 1) and status codes (Coproc 0).

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

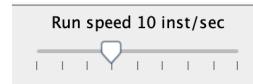
  

Name	Float	Double
\$f0	0x00000000	0x0000000000000000
\$f1	0x00000000	0x0000000000000000
\$f2	0x00000000	0x0000000000000000
\$f3	0x00000000	0x0000000000000000
\$f4	0x00000000	0x0000000000000000
\$f5	0x00000000	0x0000000000000000
\$f6	0x00000000	0x0000000000000000
\$f7	0x00000000	0x0000000000000000
\$f8	0x00000000	0x0000000000000000
\$f9	0x00000000	0x0000000000000000
\$f10	0x00000000	0x0000000000000000
\$f11	0x00000000	0x0000000000000000
\$f12	0x00000000	0x0000000000000000
\$f13	0x00000000	0x0000000000000000
\$f14	0x00000000	0x0000000000000000
\$f15	0x00000000	0x0000000000000000
\$f16	0x00000000	0x0000000000000000
\$f17	0x00000000	0x0000000000000000
\$f18	0x00000000	0x0000000000000000
\$f19	0x00000000	0x0000000000000000
\$f20	0x00000000	0x0000000000000000
\$f21	0x00000000	0x0000000000000000
\$f22	0x00000000	0x0000000000000000
\$f23	0x00000000	0x0000000000000000
\$f24	0x00000000	0x0000000000000000
\$f25	0x00000000	0x0000000000000000
\$f26	0x00000000	0x0000000000000000
\$f27	0x00000000	0x0000000000000000
\$f28	0x00000000	0x0000000000000000
\$f29	0x00000000	0x0000000000000000
\$f30	0x00000000	0x0000000000000000
\$f31	0x00000000	0x0000000000000000

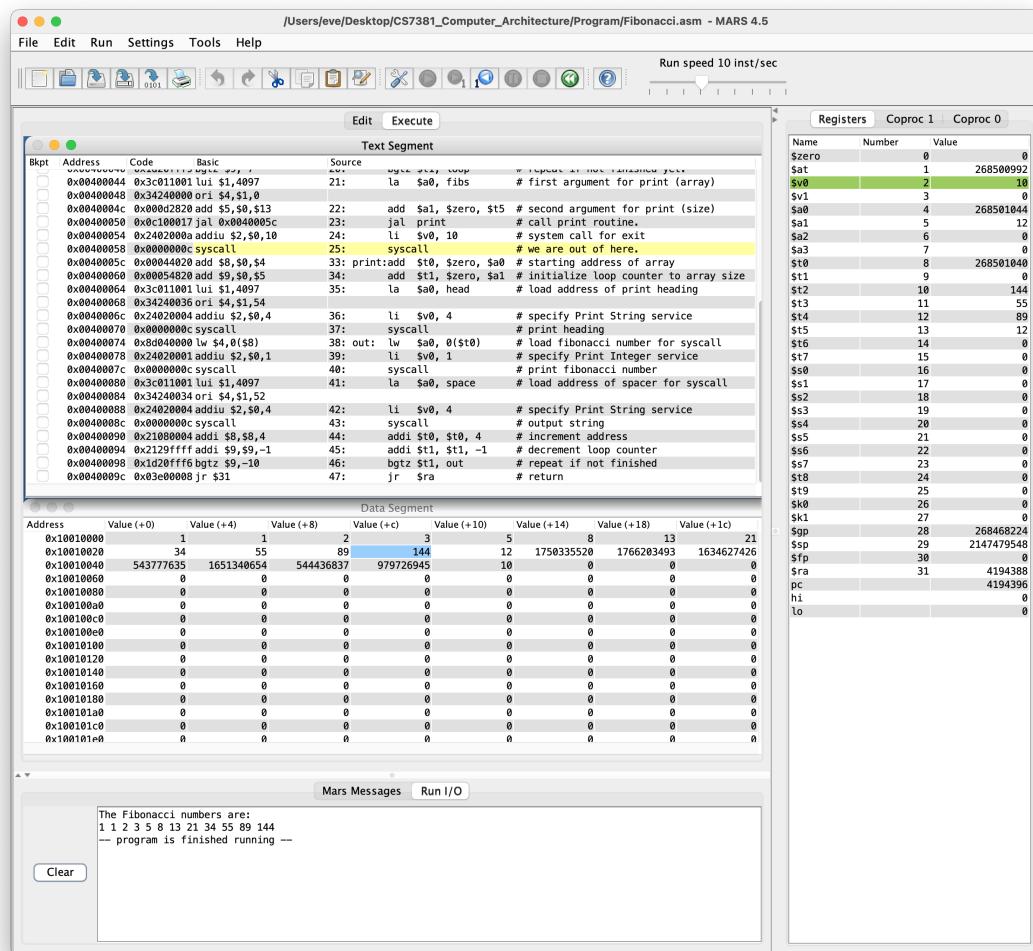
Name	Number	Value
\$8 (vaddr)	8	0x00000000
\$12 (status)	12	0x0000ff11
\$13 (cause)	13	0x00000000
\$14 (epc)	14	0x00000000

6. Use the slider bar to change the run speed to about 10 instructions per second. This allows us to “watch the action” instead of the assembly program finishing directly.

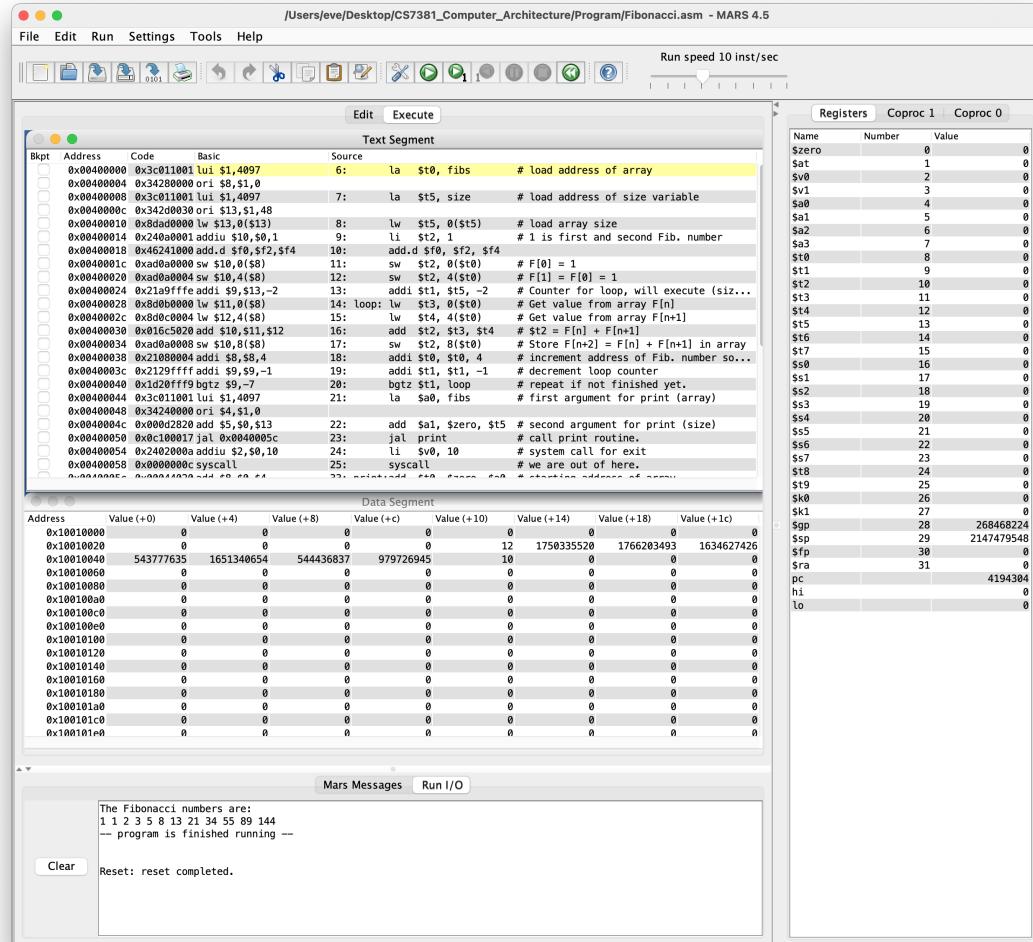


## 7. Choose how you will execute the program:

The icon runs the program to completion. Using this icon, you should observe the yellow highlight showing the program's progress and the values of the Fibonacci sequence appearing in the Data Segment display.



- The  icon resets the program and simulator to initial values. Memory contents are those specified within the program, and register contents are generally zero.



- The icon is “single-step.” Its complement is , “single-step backwards” (undoes each operation).

Screenshot of the MARS 4.5 assembly debugger showing the execution of a Fibonacci program. The assembly code in the Text Segment shows the implementation of a loop to calculate Fibonacci numbers. The Registers window shows the state of various寄存器 (Registers) from \$zero to \$t9, \$k0, and \$gp. The Data Segment window shows memory starting at address 0x10010000. The Mars Messages window displays the output: "The Fibonacci numbers are: 1 2 3 5 8 13 21 34 55 89 144 -- program is finished running --".

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194388
hi		0
lo		0

/Users/eve/Desktop/CS7381\_Computer\_Architecture/Program/Fibonacci.asm - MARS 4.5

File Edit Run Settings Tools Help      Run speed 10 inst/sec

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304
hi		0
lo		0

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001 lui \$1,4097	6:	la \$t0, fibs # load address of array
	0x00400004	0x34280000 ori \$8,\$1,0	7:	la \$t5, size # load address of size variable
	0x00400008	0x3c011001 lui \$1,4097	8:	lw \$t5, 0(\$t5) # load array size
	0x0040000c	0x342d0030 ori \$13,\$1,48	9:	li \$t2, 1 # 1 is first and second Fib. number
	0x00400010	0x8dad0000 lw \$13,0(\$13)	10:	add.d \$t0, \$t2, \$f4 # 1 + 1 = 2
	0x00400014	0x24ba0001 addiu \$t0,\$t2,\$f4	11:	sw \$t2, 0(\$t0) # F[0] = 1
	0x00400018	0x46241000 add.d \$t0,\$t2,\$f4	12:	sw \$t2, 4(\$t0) # F[1] = F[0] = 1
	0x00400022	0xadda0004 sw \$10,4(\$8)	13:	addi \$t1, \$t5, -2 # Counter for loop, will execute (size-2)
	0x00400024	0x21a9ffff addi \$9,\$13,-2	14: loop:	lw \$t3, 0(\$t0) # Get value from array F[n]
	0x00400028	0x21a9ffff addi \$9,\$13,-2	15:	lw \$t4, 4(\$t0) # Get value from array F[n+1]
	0x0040002c	0x8dad0004 lw \$12,0(\$8)	16:	add \$t2, \$t3, \$t4 # t2 = t3 + t4
	0x00400030	0x81fc5020 add \$10,\$t2,\$t2	17:	sw \$t2, 0(\$t0) # Store F[n+2] = F[n] + F[n+1] in array
	0x00400034	0xadda0004 sw \$10,0(\$8)	18:	addi \$t0, \$t0, 4 # increment address of Fib. number so...
	0x00400038	0x21a9ffff addi \$9,\$9,-4	19:	addi \$t1, \$t1, -1 # decrement loop counter
	0x0040003c	0x21a9ffff addi \$9,\$9,-1	20:	bgtz \$t1, loop # repeat if not finished yet.
	0x00400040	0x8dad0004 lw \$10,0(\$8)	21:	la \$00, fibs # first argument for print (array)
	0x00400044	0x3c011001 lui \$1,4097	22:	add \$t1, \$zero, \$t5 # second argument for print (size)
	0x00400048	0x342d2820 add \$5,\$9,\$13	23:	jal print # call print routine.
	0x00400052	0x0c010031 jal 0x0040005c	24:	li \$v0, 10 # call system call for exit
	0x00400056	0x24b00004 addiu \$2,\$0,\$10	25:	syscall # we are out of here.
	0x00400058	0x00000000 syscall	26:	mfhi \$t0 # returning address of main

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	12	1750335520	1766203493	1634627426
0x10010040	543777635	1651340654	544436837	979726945	10	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0
0x100101a0	0	0	0	0	0	0	0	0
0x100101c0	0	0	0	0	0	0	0	0
0x100101e0	0	0	0	0	0	0	0	0

Mars Messages Run I/O

The Fibonacci numbers are:  
1 1 2 3 5 8 13 21 34 55 89 144  
-- program is finished running --

Clear Reset: reset completed.

8. Observe the output of the program in the Run I/O display window:

The Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144

- program is finished running -

The screenshot shows the MARS 4.5 assembly debugger interface. The assembly code window displays the assembly code for the Fibonacci program, including instructions like `la $a0, fibs`, `add $a1, $zero, $t5`, and `jal print`. The registers window shows various寄存器 (Registers) with their values. The run I/O window at the bottom shows the output "The Fibonacci numbers are: 1 1 2 3 5 8 13 21 34 55 89 144 -- program is finished running --".

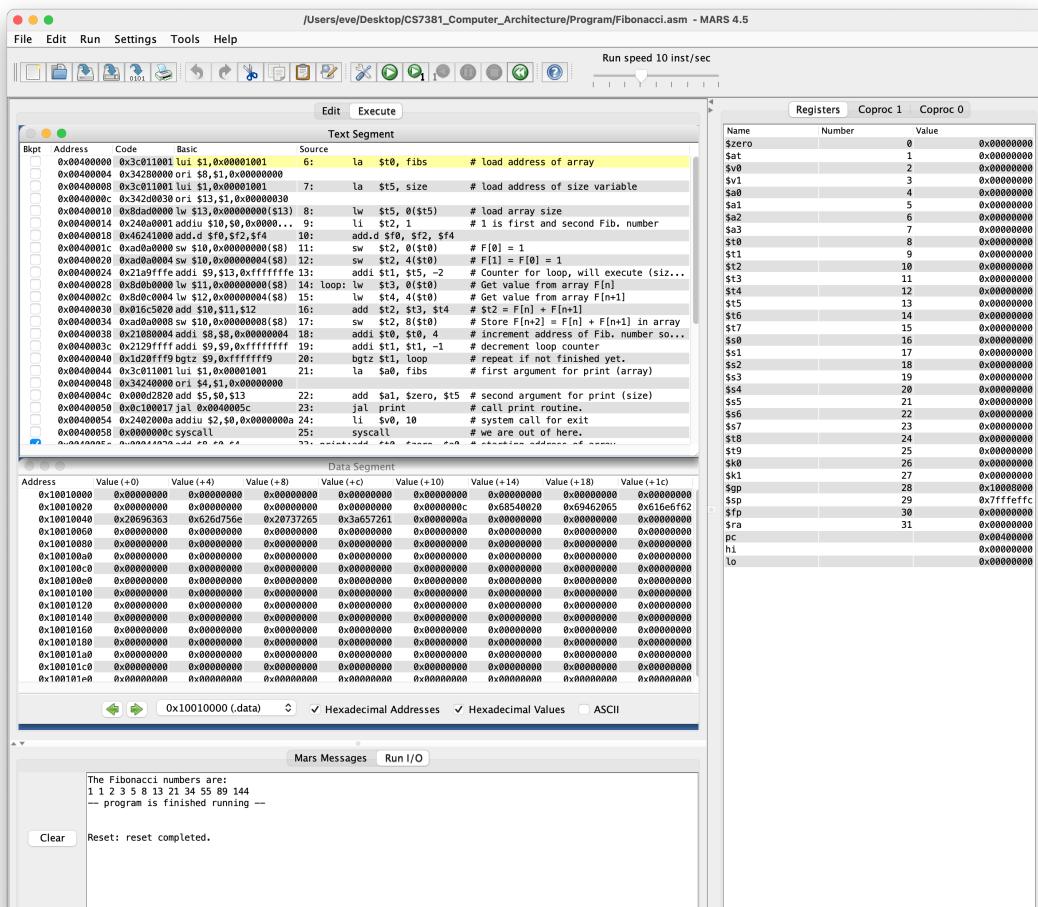
Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x10010034
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x10010030
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000037
\$t4	12	0x00000059
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t10	26	0x00000000
\$k1	27	0x00000000
\$sp	28	0x10003800
\$gp	29	0xfffffeffc
\$fp	30	0x00000000
\$ra	31	0x04000054
pc		0x0400005c
hi		0x00000000
lo		0x00000000

## 9. Modify the contents of memory. (Modifying a register value is exactly the same.)

- Set a breakpoint at the first instruction of the subroutine which prints results. Use the checkbox at the left of the instruction whose address is 0x0040005c.

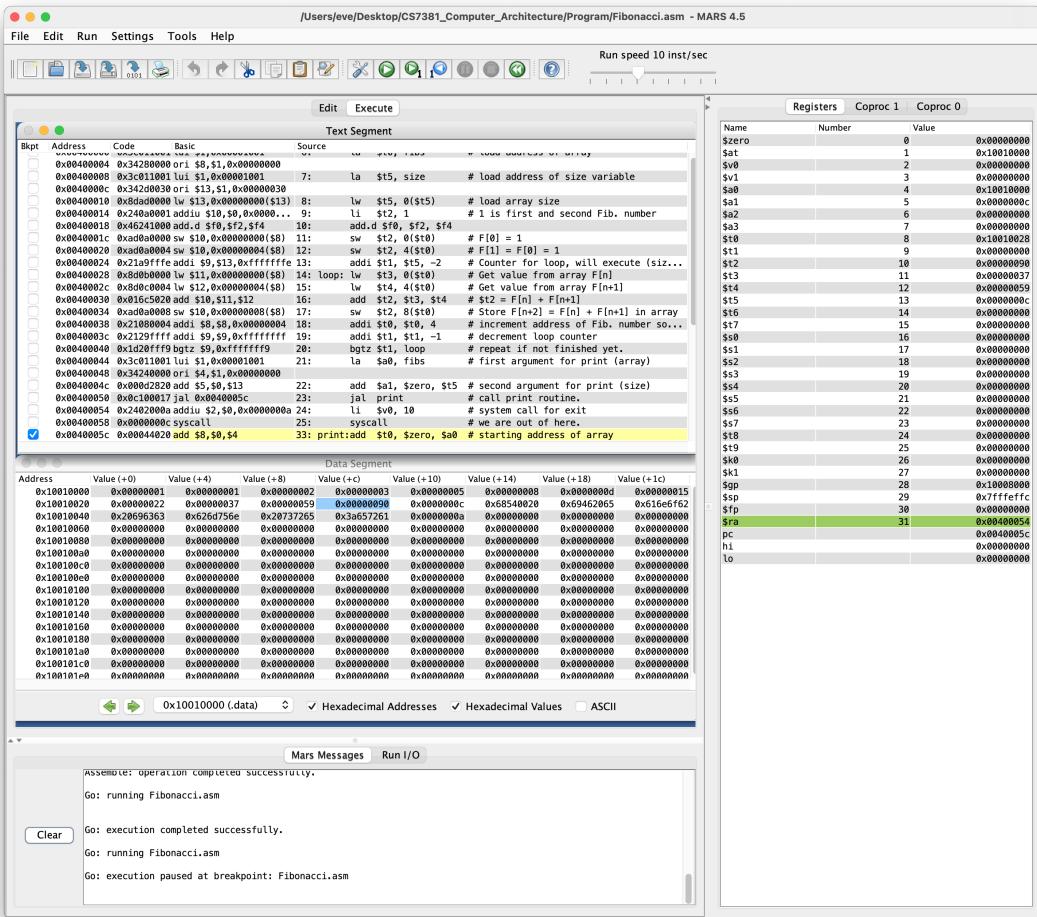
Text Segment					
Bkpt	Address	Code	Basic	Source	
	0x0040001c	0xad0a0000 sw \$10,0x0000000(\$8)	11:	sw \$t2, 0(\$t0) # F[0] = 1	
	0x00400020	0xad0a0004 sw \$10,0x00000004(\$8)	12:	sw \$t2, 4(\$t0) # F[1] = F[0] = 1	
	0x00400024	0x21a9ffff addi \$9,\$13,0xfffffff1e	13:	addi \$t1, \$t5, -2 # Counter for loop, will execute (size...)	
	0x00400028	0x8d0b0000 lw \$11,0x0000000(\$8)	14: loop:	lw \$t3, 0(\$t0) # Get value from array F[n]	
	0x0040002c	0x8d0c0004 lw \$12,0x00000004(\$8)	15:	lw \$t4, 4(\$t0) # Get value from array F[n+1]	
	0x00400030	0x016c5020 add \$10,\$11,\$12	16:	add \$t2, \$t3, \$t4 # \$t2 = F[n] + F[n+1]	
	0x00400034	0xad0a0008 sw \$10,0x00000008(\$8)	17:	sw \$t2, 8(\$t0) # Store F[n+2] = F[n] + F[n+1] in array	
	0x00400038	0x21080004 addi \$8,\$8,0xfffffff1e	18:	addi \$t0, \$t0, 4 # increment address of Fib. number so...	
	0x0040003c	0x2129ffff addi \$9,\$9,0xfffffff1f	19:	addi \$t1, \$t1, -1 # decrement loop counter	
	0x00400040	0x1d20ffff bgtz \$9,0xfffffff1f	20:	bgtz \$t1, loop # repeat if not finished yet.	
	0x00400044	0x3c011001 lui \$1,0x00001001	21:	la \$a0, fibs # first argument for print (array)	
	0x00400048	0x34240000 ori \$4,\$1,0x00000000			
	0x0040004c	0x0d02820 add \$5,\$0,\$13	22:	add \$a1, \$zero, \$t5 # second argument for print (size)	
	0x00400050	0xc100017 jal 0x0040005c	23:	jal print # call print routine.	
	0x00400054	0x2402000a addiu \$2,\$0,0x0000000a	24:	li \$v0, 10 # system call for exit	
<input checked="" type="checkbox"/>	0x00400058	0x0000000c syscall	25:	syscall # we are out of here.	
	0x0040005c	0x00404020 add \$8,\$0,\$4	33: print:add	\$t0, \$zero, \$a0 # starting address of array	
	0x00400060	0x00054820 add \$9,\$0,\$5	34:	add \$t1, \$zero, \$a1 # initialize loop counter to array size	
	0x00400064	0x3c011001 lui \$1,0x00001001	35:	la \$a0, head # load address of print heading	
	0x00400068	0x34240036 ori \$4,\$1,0x00000036			
	0x0040006c	0x24020004 addiu \$2,\$0,0x00000004	36:	li \$v0, 4 # specify Print String service	
	0x00400070	0x0000000c syscall	37:	syscall # print heading	
	0x00400074	0x8d040000 lw \$4,0x0000000(\$8)	38: out:	lw \$a0, 0(\$t0) # load fibonacci number for syscall	
	0x00400078	0x21020001 addiu \$2,\$0,0x00000001	39:	add \$a0, 1 # specify print. Entering routine	

- Reset  and re-run  the program, which stops at the breakpoint.



The Fibonacci numbers are:  
1 1 2 3 5 8 13 21 34 55 89 144  
-- program is finished running --

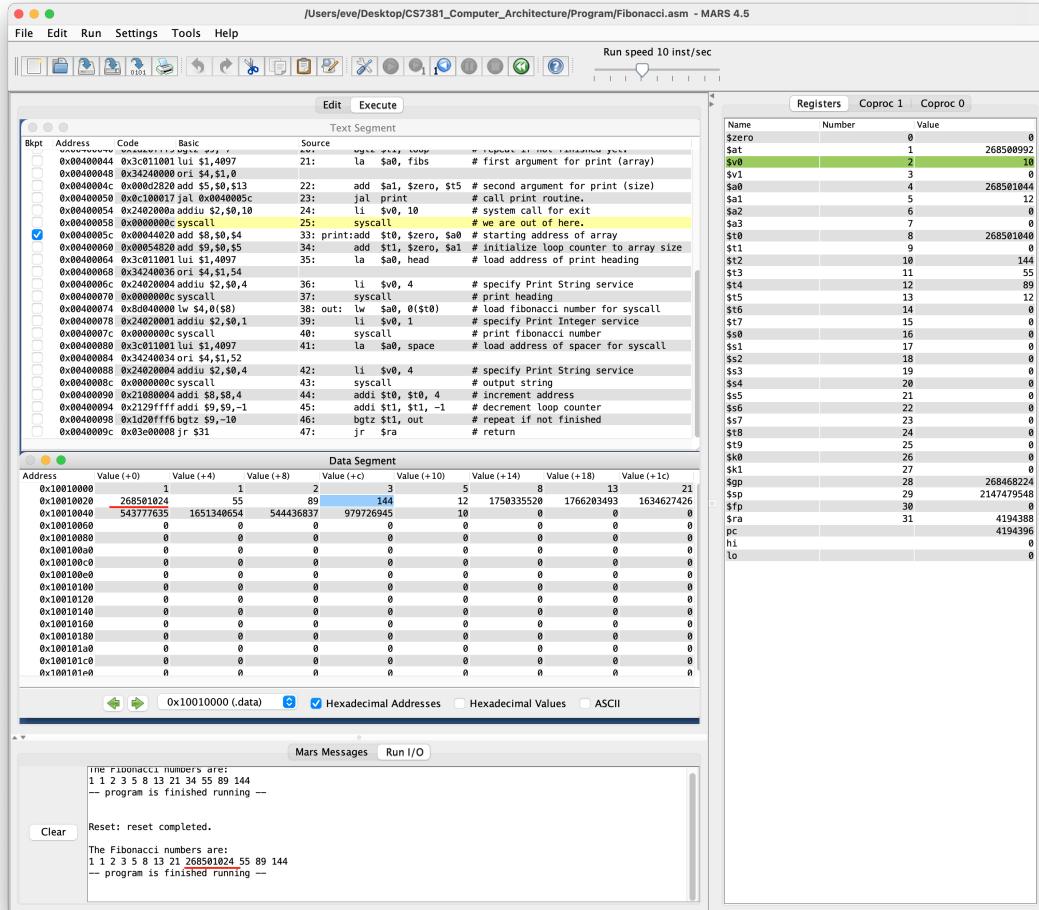
Reset: reset completed.



- Double-click in one of the memory locations containing the computed Fibonacci numbers. The cell will be highlighted and will accept keyboard entry, similar to a spreadsheet. Enter some noticeably different value, and use the Enter key or click outside the cell to indicate that the change is complete. Example: Memory address  $0x10010020 = 268501024_{ten}$  presently contains data  $0x000000022 = 34_{ten}$ .

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	1	1	2	3	5	8	13	21	
0x10010020	268501024	55	89	144	12	1750335520	1766203493	1634627426	
0x10010040	543777635	1651340654	544436837	979726945	10	0	0	0	
0x10010060	0	0	0	0	0	0	0	0	
0x10010080	0	0	0	0	0	0	0	0	
0x100100a0	0	0	0	0	0	0	0	0	
0x100100c0	0	0	0	0	0	0	0	0	
0x100100e0	0	0	0	0	0	0	0	0	
0x10010100	0	0	0	0	0	0	0	0	
0x10010120	0	0	0	0	0	0	0	0	
0x10010140	0	0	0	0	0	0	0	0	
0x10010160	0	0	0	0	0	0	0	0	
0x10010180	0	0	0	0	0	0	0	0	
0x100101a0	0	0	0	0	0	0	0	0	
0x100101c0	0	0	0	0	0	0	0	0	
0x100101e0	0	0	0	0	0	0	0	0	

- Click to continue from the breakpoint. The program output includes your entered value instead of the computed Fibonacci number.



10. Open the Help  for information on MIPS instructions, pseudoinstructions, directives, and syscalls.

