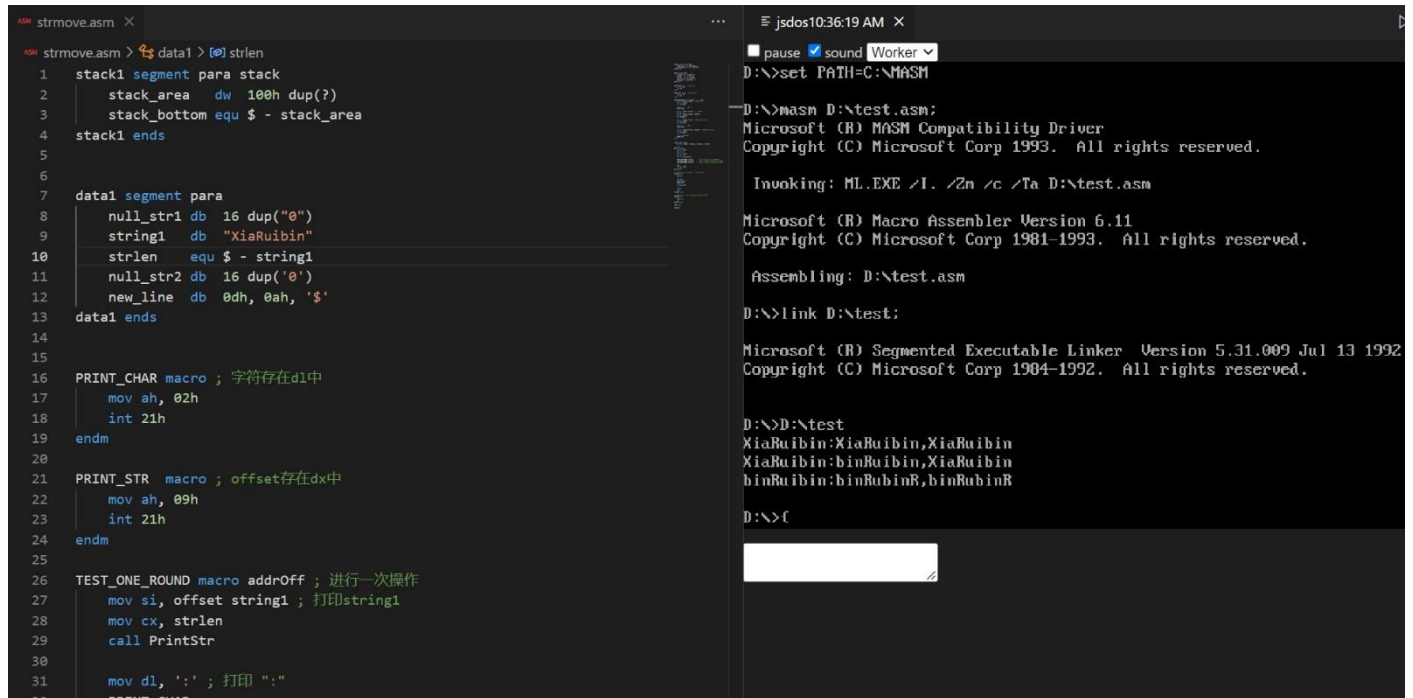


程序运行截图

第 1 题: strmove



```
strmove.asm X
strmove.asm > data1 > [0] strlen
1  stack1 segment para stack
2      stack_area dw 100h dup(?)
3      stack_bottom equ $ - stack_area
4  stack1 ends
5
6
7  data1 segment para
8      null_str1 db 16 dup("0")
9      string1 db "XiaRuibin"
10     strlen equ $ - string1
11     null_str2 db 16 dup('0')
12     new_line db 0dh, 0ah, '$'
13 data1 ends
14
15
16 PRINT_CHAR macro ; 字符存在d1中
17     mov ah, 02h
18     int 21h
19 endm
20
21 PRINT_STR macro ; offset存在dx中
22     mov ah, 09h
23     int 21h
24 endm
25
26 TEST_ONE_ROUND macro addrOff ; 进行一次操作
27     mov si, offset string1 ; 打印string1
28     mov cx, strlen
29     call PrintStr
30
31     mov d1, ':' ; 打印 ":"
32     call PrintChar
33 endm
34
35 code segment para
36     jmp main
37 code ends
38
39 main:
40     mov dx, offset string1
41     mov cx, strlen
42     call TEST_ONE_ROUND
43     jmp main
44
45 main:
46     mov dx, offset string1
47     mov cx, strlen
48     call TEST_ONE_ROUND
49     jmp main
50
51 main:
52     mov dx, offset string1
53     mov cx, strlen
54     call TEST_ONE_ROUND
55     jmp main
56
57 main:
58     mov dx, offset string1
59     mov cx, strlen
60     call TEST_ONE_ROUND
61     jmp main
62
63 main:
64     mov dx, offset string1
65     mov cx, strlen
66     call TEST_ONE_ROUND
67     jmp main
68
69 main:
70     mov dx, offset string1
71     mov cx, strlen
72     call TEST_ONE_ROUND
73     jmp main
74
75 main:
76     mov dx, offset string1
77     mov cx, strlen
78     call TEST_ONE_ROUND
79     jmp main
80
81 main:
82     mov dx, offset string1
83     mov cx, strlen
84     call TEST_ONE_ROUND
85     jmp main
86
87 main:
88     mov dx, offset string1
89     mov cx, strlen
90     call TEST_ONE_ROUND
91     jmp main
92
93 main:
94     mov dx, offset string1
95     mov cx, strlen
96     call TEST_ONE_ROUND
97     jmp main
98
99 main:
100    mov dx, offset string1
101    mov cx, strlen
102    call TEST_ONE_ROUND
103    jmp main
104
105 main:
106    mov dx, offset string1
107    mov cx, strlen
108    call TEST_ONE_ROUND
109    jmp main
110
111 main:
112    mov dx, offset string1
113    mov cx, strlen
114    call TEST_ONE_ROUND
115    jmp main
116
117 main:
118    mov dx, offset string1
119    mov cx, strlen
120    call TEST_ONE_ROUND
121    jmp main
122
123 main:
124    mov dx, offset string1
125    mov cx, strlen
126    call TEST_ONE_ROUND
127    jmp main
128
129 main:
130    mov dx, offset string1
131    mov cx, strlen
132    call TEST_ONE_ROUND
133    jmp main
134
135 main:
136    mov dx, offset string1
137    mov cx, strlen
138    call TEST_ONE_ROUND
139    jmp main
140
141 main:
142    mov dx, offset string1
143    mov cx, strlen
144    call TEST_ONE_ROUND
145    jmp main
146
147 main:
148    mov dx, offset string1
149    mov cx, strlen
150    call TEST_ONE_ROUND
151    jmp main
152
153 main:
154    mov dx, offset string1
155    mov cx, strlen
156    call TEST_ONE_ROUND
157    jmp main
158
159 main:
160    mov dx, offset string1
161    mov cx, strlen
162    call TEST_ONE_ROUND
163    jmp main
164
165 main:
166    mov dx, offset string1
167    mov cx, strlen
168    call TEST_ONE_ROUND
169    jmp main
170
171 main:
172    mov dx, offset string1
173    mov cx, strlen
174    call TEST_ONE_ROUND
175    jmp main
176
177 main:
178    mov dx, offset string1
179    mov cx, strlen
180    call TEST_ONE_ROUND
181    jmp main
182
183 main:
184    mov dx, offset string1
185    mov cx, strlen
186    call TEST_ONE_ROUND
187    jmp main
188
189 main:
190    mov dx, offset string1
191    mov cx, strlen
192    call TEST_ONE_ROUND
193    jmp main
194
195 main:
196    mov dx, offset string1
197    mov cx, strlen
198    call TEST_ONE_ROUND
199    jmp main
200
201 main:
202    mov dx, offset string1
203    mov cx, strlen
204    call TEST_ONE_ROUND
205    jmp main
206
207 main:
208    mov dx, offset string1
209    mov cx, strlen
210    call TEST_ONE_ROUND
211    jmp main
212
213 main:
214    mov dx, offset string1
215    mov cx, strlen
216    call TEST_ONE_ROUND
217    jmp main
218
219 main:
220    mov dx, offset string1
221    mov cx, strlen
222    call TEST_ONE_ROUND
223    jmp main
224
225 main:
226    mov dx, offset string1
227    mov cx, strlen
228    call TEST_ONE_ROUND
229    jmp main
230
231 main:
232    mov dx, offset string1
233    mov cx, strlen
234    call TEST_ONE_ROUND
235    jmp main
236
237 main:
238    mov dx, offset string1
239    mov cx, strlen
240    call TEST_ONE_ROUND
241    jmp main
242
243 main:
244    mov dx, offset string1
245    mov cx, strlen
246    call TEST_ONE_ROUND
247    jmp main
248
249 main:
250    mov dx, offset string1
251    mov cx, strlen
252    call TEST_ONE_ROUND
253    jmp main
254
255 main:
256    mov dx, offset string1
257    mov cx, strlen
258    call TEST_ONE_ROUND
259    jmp main
260
261 main:
262    mov dx, offset string1
263    mov cx, strlen
264    call TEST_ONE_ROUND
265    jmp main
266
267 main:
268    mov dx, offset string1
269    mov cx, strlen
270    call TEST_ONE_ROUND
271    jmp main
272
273 main:
274    mov dx, offset string1
275    mov cx, strlen
276    call TEST_ONE_ROUND
277    jmp main
278
279 main:
280    mov dx, offset string1
281    mov cx, strlen
282    call TEST_ONE_ROUND
283    jmp main
284
285 main:
286    mov dx, offset string1
287    mov cx, strlen
288    call TEST_ONE_ROUND
289    jmp main
290
291 main:
292    mov dx, offset string1
293    mov cx, strlen
294    call TEST_ONE_ROUND
295    jmp main
296
297 main:
298    mov dx, offset string1
299    mov cx, strlen
300    call TEST_ONE_ROUND
301    jmp main
302
303 main:
304    mov dx, offset string1
305    mov cx, strlen
306    call TEST_ONE_ROUND
307    jmp main
308
309 main:
310    mov dx, offset string1
311    mov cx, strlen
312    call TEST_ONE_ROUND
313    jmp main
314
315 main:
316    mov dx, offset string1
317    mov cx, strlen
318    call TEST_ONE_ROUND
319    jmp main
320
321 main:
322    mov dx, offset string1
323    mov cx, strlen
324    call TEST_ONE_ROUND
325    jmp main
326
327 main:
328    mov dx, offset string1
329    mov cx, strlen
330    call TEST_ONE_ROUND
331    jmp main
332
333 main:
334    mov dx, offset string1
335    mov cx, strlen
336    call TEST_ONE_ROUND
337    jmp main
338
339 main:
340    mov dx, offset string1
341    mov cx, strlen
342    call TEST_ONE_ROUND
343    jmp main
344
345 main:
346    mov dx, offset string1
347    mov cx, strlen
348    call TEST_ONE_ROUND
349    jmp main
350
351 main:
352    mov dx, offset string1
353    mov cx, strlen
354    call TEST_ONE_ROUND
355    jmp main
356
357 main:
358    mov dx, offset string1
359    mov cx, strlen
360    call TEST_ONE_ROUND
361    jmp main
362
363 main:
364    mov dx, offset string1
365    mov cx, strlen
366    call TEST_ONE_ROUND
367    jmp main
368
369 main:
370    mov dx, offset string1
371    mov cx, strlen
372    call TEST_ONE_ROUND
373    jmp main
374
375 main:
376    mov dx, offset string1
377    mov cx, strlen
378    call TEST_ONE_ROUND
379    jmp main
380
381 main:
382    mov dx, offset string1
383    mov cx, strlen
384    call TEST_ONE_ROUND
385    jmp main
386
387 main:
388    mov dx, offset string1
389    mov cx, strlen
390    call TEST_ONE_ROUND
391    jmp main
392
393 main:
394    mov dx, offset string1
395    mov cx, strlen
396    call TEST_ONE_ROUND
397    jmp main
398
399 main:
400    mov dx, offset string1
401    mov cx, strlen
402    call TEST_ONE_ROUND
403    jmp main
404
405 main:
406    mov dx, offset string1
407    mov cx, strlen
408    call TEST_ONE_ROUND
409    jmp main
410
411 main:
412    mov dx, offset string1
413    mov cx, strlen
414    call TEST_ONE_ROUND
415    jmp main
416
417 main:
418    mov dx, offset string1
419    mov cx, strlen
420    call TEST_ONE_ROUND
421    jmp main
422
423 main:
424    mov dx, offset string1
425    mov cx, strlen
426    call TEST_ONE_ROUND
427    jmp main
428
429 main:
430    mov dx, offset string1
431    mov cx, strlen
432    call TEST_ONE_ROUND
433    jmp main
434
435 main:
436    mov dx, offset string1
437    mov cx, strlen
438    call TEST_ONE_ROUND
439    jmp main
440
441 main:
442    mov dx, offset string1
443    mov cx, strlen
444    call TEST_ONE_ROUND
445    jmp main
446
447 main:
448    mov dx, offset string1
449    mov cx, strlen
450    call TEST_ONE_ROUND
451    jmp main
452
453 main:
454    mov dx, offset string1
455    mov cx, strlen
456    call TEST_ONE_ROUND
457    jmp main
458
459 main:
460    mov dx, offset string1
461    mov cx, strlen
462    call TEST_ONE_ROUND
463    jmp main
464
465 main:
466    mov dx, offset string1
467    mov cx, strlen
468    call TEST_ONE_ROUND
469    jmp main
470
471 main:
472    mov dx, offset string1
473    mov cx, strlen
474    call TEST_ONE_ROUND
475    jmp main
476
477 main:
478    mov dx, offset string1
479    mov cx, strlen
480    call TEST_ONE_ROUND
481    jmp main
482
483 main:
484    mov dx, offset string1
485    mov cx, strlen
486    call TEST_ONE_ROUND
487    jmp main
488
489 main:
490    mov dx, offset string1
491    mov cx, strlen
492    call TEST_ONE_ROUND
493    jmp main
494
495 main:
496    mov dx, offset string1
497    mov cx, strlen
498    call TEST_ONE_ROUND
499    jmp main
500
501 main:
502    mov dx, offset string1
503    mov cx, strlen
504    call TEST_ONE_ROUND
505    jmp main
506
507 main:
508    mov dx, offset string1
509    mov cx, strlen
510    call TEST_ONE_ROUND
511    jmp main
512
513 main:
514    mov dx, offset string1
515    mov cx, strlen
516    call TEST_ONE_ROUND
517    jmp main
518
519 main:
520    mov dx, offset string1
521    mov cx, strlen
522    call TEST_ONE_ROUND
523    jmp main
524
525 main:
526    mov dx, offset string1
527    mov cx, strlen
528    call TEST_ONE_ROUND
529    jmp main
530
531 main:
532    mov dx, offset string1
533    mov cx, strlen
534    call TEST_ONE_ROUND
535    jmp main
536
537 main:
538    mov dx, offset string1
539    mov cx, strlen
540    call TEST_ONE_ROUND
541    jmp main
542
543 main:
544    mov dx, offset string1
545    mov cx, strlen
546    call TEST_ONE_ROUND
547    jmp main
548
549 main:
550    mov dx, offset string1
551    mov cx, strlen
552    call TEST_ONE_ROUND
553    jmp main
554
555 main:
556    mov dx, offset string1
557    mov cx, strlen
558    call TEST_ONE_ROUND
559    jmp main
560
561 main:
562    mov dx, offset string1
563    mov cx, strlen
564    call TEST_ONE_ROUND
565    jmp main
566
567 main:
568    mov dx, offset string1
569    mov cx, strlen
570    call TEST_ONE_ROUND
571    jmp main
572
573 main:
574    mov dx, offset string1
575    mov cx, strlen
576    call TEST_ONE_ROUND
577    jmp main
578
579 main:
580    mov dx, offset string1
581    mov cx, strlen
582    call TEST_ONE_ROUND
583    jmp main
584
585 main:
586    mov dx, offset string1
587    mov cx, strlen
588    call TEST_ONE_ROUND
589    jmp main
590
591 main:
592    mov dx, offset string1
593    mov cx, strlen
594    call TEST_ONE_ROUND
595    jmp main
596
597 main:
598    mov dx, offset string1
599    mov cx, strlen
600    call TEST_ONE_ROUND
601    jmp main
602
603 main:
604    mov dx, offset string1
605    mov cx, strlen
606    call TEST_ONE_ROUND
607    jmp main
608
609 main:
610    mov dx, offset string1
611    mov cx, strlen
612    call TEST_ONE_ROUND
613    jmp main
614
615 main:
616    mov dx, offset string1
617    mov cx, strlen
618    call TEST_ONE_ROUND
619    jmp main
620
621 main:
622    mov dx, offset string1
623    mov cx, strlen
624    call TEST_ONE_ROUND
625    jmp main
626
627 main:
628    mov dx, offset string1
629    mov cx, strlen
630    call TEST_ONE_ROUND
631    jmp main
632
633 main:
634    mov dx, offset string1
635    mov cx, strlen
636    call TEST_ONE_ROUND
637    jmp main
638
639 main:
640    mov dx, offset string1
641    mov cx, strlen
642    call TEST_ONE_ROUND
643    jmp main
644
645 main:
646    mov dx, offset string1
647    mov cx, strlen
648    call TEST_ONE_ROUND
649    jmp main
650
651 main:
652    mov dx, offset string1
653    mov cx, strlen
654    call TEST_ONE_ROUND
655    jmp main
656
657 main:
658    mov dx, offset string1
659    mov cx, strlen
660    call TEST_ONE_ROUND
661    jmp main
662
663 main:
664    mov dx, offset string1
665    mov cx, strlen
666    call TEST_ONE_ROUND
667    jmp main
668
669 main:
670    mov dx, offset string1
671    mov cx, strlen
672    call TEST_ONE_ROUND
673    jmp main
674
675 main:
676    mov dx, offset string1
677    mov cx, strlen
678    call TEST_ONE_ROUND
679    jmp main
680
681 main:
682    mov dx, offset string1
683    mov cx, strlen
684    call TEST_ONE_ROUND
685    jmp main
686
687 main:
688    mov dx, offset string1
689    mov cx, strlen
690    call TEST_ONE_ROUND
691    jmp main
692
693 main:
694    mov dx, offset string1
695    mov cx, strlen
696    call TEST_ONE_ROUND
697    jmp main
698
699 main:
700    mov dx, offset string1
701    mov cx, strlen
702    call TEST_ONE_ROUND
703    jmp main
704
705 main:
706    mov dx, offset string1
707    mov cx, strlen
708    call TEST_ONE_ROUND
709    jmp main
710
711 main:
712    mov dx, offset string1
713    mov cx, strlen
714    call TEST_ONE_ROUND
715    jmp main
716
717 main:
718    mov dx, offset string1
719    mov cx, strlen
720    call TEST_ONE_ROUND
721    jmp main
722
723 main:
724    mov dx, offset string1
725    mov cx, strlen
726    call TEST_ONE_ROUND
727    jmp main
728
729 main:
730    mov dx, offset string1
731    mov cx, strlen
732    call TEST_ONE_ROUND
733    jmp main
734
735 main:
736    mov dx, offset string1
737    mov cx, strlen
738    call TEST_ONE_ROUND
739    jmp main
740
741 main:
742    mov dx, offset string1
743    mov cx, strlen
744    call TEST_ONE_ROUND
745    jmp main
746
747 main:
748    mov dx, offset string1
749    mov cx, strlen
750    call TEST_ONE_ROUND
751    jmp main
752
753 main:
754    mov dx, offset string1
755    mov cx, strlen
756    call TEST_ONE_ROUND
757    jmp main
758
759 main:
760    mov dx, offset string1
761    mov cx, strlen
762    call TEST_ONE_ROUND
763    jmp main
764
765 main:
766    mov dx, offset string1
767    mov cx, strlen
768    call TEST_ONE_ROUND
769    jmp main
770
771 main:
772    mov dx, offset string1
773    mov cx, strlen
774    call TEST_ONE_ROUND
775    jmp main
776
777 main:
778    mov dx, offset string1
779    mov cx, strlen
780    call TEST_ONE_ROUND
781    jmp main
782
783 main:
784    mov dx, offset string1
785    mov cx, strlen
786    call TEST_ONE_ROUND
787    jmp main
788
789 main:
790    mov dx, offset string1
791    mov cx, strlen
792    call TEST_ONE_ROUND
793    jmp main
794
795 main:
796    mov dx, offset string1
797    mov cx, strlen
798    call TEST_ONE_ROUND
799    jmp main
800
801 main:
802    mov dx, offset string1
803    mov cx, strlen
804    call TEST_ONE_ROUND
805    jmp main
806
807 main:
808    mov dx, offset string1
809    mov cx, strlen
810    call TEST_ONE_ROUND
811    jmp main
812
813 main:
814    mov dx, offset string1
815    mov cx, strlen
816    call TEST_ONE_ROUND
817    jmp main
818
819 main:
820    mov dx, offset string1
821    mov cx, strlen
822    call TEST_ONE_ROUND
823    jmp main
824
825 main:
826    mov dx, offset string1
827    mov cx, strlen
828    call TEST_ONE_ROUND
829    jmp main
830
831 main:
832    mov dx, offset string1
833    mov cx, strlen
834    call TEST_ONE_ROUND
835    jmp main
836
837 main:
838    mov dx, offset string1
839    mov cx, strlen
840    call TEST_ONE_ROUND
841    jmp main
842
843 main:
844    mov dx, offset string1
845    mov cx, strlen
846    call TEST_ONE_ROUND
847    jmp main
848
849 main:
850    mov dx, offset string1
851    mov cx, strlen
852    call TEST_ONE_ROUND
853    jmp main
854
855 main:
856    mov dx, offset string1
857    mov cx, strlen
858    call TEST_ONE_ROUND
859    jmp main
860
861 main:
862    mov dx, offset string1
863    mov cx, strlen
864    call TEST_ONE_ROUND
865    jmp main
866
867 main:
868    mov dx, offset string1
869    mov cx, strlen
870    call TEST_ONE_ROUND
871    jmp main
872
873 main:
874    mov dx, offset string1
875    mov cx, strlen
876    call TEST_ONE_ROUND
877    jmp main
878
879 main:
880    mov dx, offset string1
881    mov cx, strlen
882    call TEST_ONE_ROUND
883    jmp main
884
885 main:
886    mov dx, offset string1
887    mov cx, strlen
888    call TEST_ONE_ROUND
889    jmp main
890
891 main:
892    mov dx, offset string1
893    mov cx, strlen
894    call TEST_ONE_ROUND
895    jmp main
896
897 main:
898    mov dx, offset string1
899    mov cx, strlen
900    call TEST_ONE_ROUND
901    jmp main
902
903 main:
904    mov dx, offset string1
905    mov cx, strlen
906    call TEST_ONE_ROUND
907    jmp main
908
909 main:
910    mov dx, offset string1
911    mov cx, strlen
912    call TEST_ONE_ROUND
913    jmp main
914
915 main:
916    mov dx, offset string1
917    mov cx, strlen
918    call TEST_ONE_ROUND
919    jmp main
920
921 main:
922    mov dx, offset string1
923    mov cx, strlen
924    call TEST_ONE_ROUND
925    jmp main
926
927 main:
928    mov dx, offset string1
929    mov cx, strlen
930    call TEST_ONE_ROUND
931    jmp main
932
933 main:
934    mov dx, offset string1
935    mov cx, strlen
936    call TEST_ONE_ROUND
937    jmp main
938
939 main:
940    mov dx, offset string1
941    mov cx, strlen
942    call TEST_ONE_ROUND
943    jmp main
944
945 main:
946    mov dx, offset string1
947    mov cx, strlen
948    call TEST_ONE_ROUND
949    jmp main
950
951 main:
952    mov dx, offset string1
953    mov cx, strlen
954    call TEST_ONE_ROUND
955    jmp main
956
957 main:
958    mov dx, offset string1
959    mov cx, strlen
960    call TEST_ONE_ROUND
961    jmp main
962
963 main:
964    mov dx, offset string1
965    mov cx, strlen
966    call TEST_ONE_ROUND
967    jmp main
968
969 main:
970    mov dx, offset string1
971    mov cx, strlen
972    call TEST_ONE_ROUND
973    jmp main
974
975 main:
976    mov dx, offset string1
977    mov cx, strlen
978    call TEST_ONE_ROUND
979    jmp main
980
981 main:
982    mov dx, offset string1
983    mov cx, strlen
984    call TEST_ONE_ROUND
985    jmp main
986
987 main:
988    mov dx, offset string1
989    mov cx, strlen
990    call TEST_ONE_ROUND
991    jmp main
992
993 main:
994    mov dx, offset string1
995    mov cx, strlen
996    call TEST_ONE_ROUND
997    jmp main
998
999 main:
1000   mov dx, offset string1
1001   mov cx, strlen
1002   call TEST_ONE_ROUND
1003   jmp main
1004
1005 main:
1006   mov dx, offset string1
1007   mov cx, strlen
1008   call TEST_ONE_ROUND
1009   jmp main
1010
1011 main:
1012   mov dx, offset string1
1013   mov cx, strlen
1014   call TEST_ONE_ROUND
1015   jmp main
1016
1017 main:
1018   mov dx, offset string1
1019   mov cx, strlen
1020   call TEST_ONE_ROUND
1021   jmp main
1022
1023 main:
1024   mov dx, offset string1
1025   mov cx, strlen
1026   call TEST_ONE_ROUND
1027   jmp main
1028
1029 main:
1030   mov dx, offset string1
1031   mov cx, strlen
1032   call TEST_ONE_ROUND
1033   jmp main
1034
1035 main:
1036   mov dx, offset string1
1037   mov cx, strlen
1038   call TEST_ONE_ROUND
1039   jmp main
1040
1041 main:
1042   mov dx, offset string1
1043   mov cx, strlen
1044   call TEST_ONE_ROUND
1045   jmp main
1046
1047 main:
1048   mov dx, offset string1
1049   mov cx, strlen
1050   call TEST_ONE_ROUND
1051   jmp main
1052
1053 main:
1054   mov dx, offset string1
1055   mov cx, strlen
1056   call TEST_ONE_ROUND
1057   jmp main
1058
1059 main:
1060   mov dx, offset string1
1061   mov cx, strlen
1062   call TEST_ONE_ROUND
1063   jmp main
1064
1065 main:
1066   mov dx, offset string1
1067   mov cx, strlen
1068   call TEST_ONE_ROUND
1069   jmp main
1070
1071 main:
1072   mov dx, offset string1
1073   mov cx, strlen
1074   call TEST_ONE_ROUND
1075   jmp main
1076
1077 main:
1078   mov dx, offset string1
1079   mov cx, strlen
1080   call TEST_ONE_ROUND
1081   jmp main
1082
1083 main:
1084   mov dx, offset string1
1085   mov cx, strlen
1086   call TEST_ONE_ROUND
1087   jmp main
1088
1089 main:
1090   mov dx, offset string1
1091   mov cx, strlen
1092   call TEST_ONE_ROUND
1093   jmp main
1094
1095 main:
1096   mov dx, offset string1
1097   mov cx, strlen
1098   call TEST_ONE_ROUND
1099   jmp main
1100
1101 main:
1102   mov dx, offset string1
1103   mov cx, strlen
1104   call TEST_ONE_ROUND
1105   jmp main
1106
1107 main:
1108   mov dx, offset string1
1109   mov cx, strlen
1110   call TEST_ONE_ROUND
1111   jmp main
1112
1113 main:
1114   mov dx, offset string1
1115   mov cx, strlen
1116   call TEST_ONE_ROUND
1117   jmp main
1118
1119 main:
1120   mov dx, offset string1
1121   mov cx, strlen
1122   call TEST_ONE_ROUND
1123   jmp main
1124
1125 main:
1126   mov dx, offset string1
1127   mov cx, strlen
1128   call TEST_ONE_ROUND
1129   jmp main
1130
1131 main:
1132   mov dx, offset string1
1133   mov cx, strlen
1134   call TEST_ONE_ROUND
1135   jmp main
1136
1137 main:
1138   mov dx, offset string1
1139   mov cx, strlen
1140   call TEST_ONE_ROUND
1141   jmp main
1142
1143 main:
1144   mov dx, offset string1
1145   mov cx, strlen
1146   call TEST_ONE_ROUND
1147   jmp main
1148
1149 main:
1150   mov dx, offset string1
1151   mov cx, strlen
1152   call TEST_ONE_ROUND
1153   jmp main
1154
1155 main:
1156   mov dx, offset string1
1157   mov cx, strlen
1158   call TEST_ONE_ROUND
1159   jmp main
1160
1161 main:
1162   mov dx, offset string1
1163   mov cx, strlen
1164   call TEST_ONE_ROUND
1165   jmp main
1166
1167 main:
1168   mov dx, offset string1
1169   mov cx, strlen
1170   call TEST_ONE_ROUND
1171   jmp main
1172
1173 main:
1174   mov dx, offset string1
1175   mov cx, strlen
1176   call TEST_ONE_ROUND
1177   jmp main
1178
1179 main:
1180   mov dx, offset string1
1181   mov cx, strlen
1182   call TEST_ONE_ROUND
1183   jmp main
1184
1185 main:
1186   mov dx, offset string1
1187   mov cx, strlen
1188   call TEST_ONE_ROUND
1189   jmp main
1190
1191 main:
1192   mov dx, offset string1
1193   mov cx, strlen
1194   call TEST_ONE_ROUND
1195   jmp main
1196
1197 main:
1198   mov dx, offset string1
1199   mov cx, strlen
1200   call TEST_ONE_ROUND
1201   jmp main
1202
1203 main:
1204   mov dx, offset string1
1205   mov cx, strlen
1206   call TEST_ONE_ROUND
1207   jmp main
1208
1209 main:
1210   mov dx, offset string1
1211   mov cx, strlen
1212   call TEST_ONE_ROUND
1213   jmp main
1214
1215 main:
1216   mov dx, offset string1
1217   mov cx, strlen
1218   call TEST_ONE_ROUND
1219   jmp main
1220
1221 main:
1222   mov dx, offset string1
1223   mov cx, strlen
1224   call TEST_ONE_ROUND
1225   jmp main
1226
1227 main:
1228   mov dx, offset string1
1229   mov cx, strlen
1230   call TEST_ONE_ROUND
1231   jmp main
1232
1233 main:
1234   mov dx, offset string1
1235   mov cx, strlen
1236   call TEST_ONE_ROUND
1237   jmp main
1238
1239 main:
1240   mov dx, offset string1
1241   mov cx, strlen
1242   call TEST_ONE_ROUND
1243   jmp main
1244
1245 main:
1246   mov dx, offset string1
1247   mov cx, strlen
1248   call TEST_ONE_ROUND
1249   jmp main
1250
1251 main:
1252   mov dx, offset string1
1253   mov cx, strlen
1254   call TEST_ONE_ROUND
1255   jmp main
1256
1257 main:
1258   mov dx, offset string1
1259   mov cx, strlen
1260   call TEST_ONE_ROUND
1261   jmp main
1262
1263 main:
1264   mov dx, offset string1
1265   mov cx, strlen
1266   call TEST_ONE_ROUND
1267   jmp main
1268
1269 main:
1270   mov dx, offset string1
1271   mov cx, strlen
1272   call TEST_ONE_ROUND
1273   jmp main
1274
1275 main:
1276   mov dx, offset string1
1277   mov cx, strlen
1278   call TEST_ONE_ROUND
1279   jmp main
1280
1281 main:
1282   mov dx, offset string1
1283   mov cx, strlen
1284   call TEST_ONE_ROUND
1285   jmp main
1286
1287 main:
1288   mov dx, offset string1
1289   mov cx, strlen
1290   call TEST_ONE_ROUND
1291   jmp main
1292
1293 main:
1294   mov dx, offset string1
1295   mov cx, strlen
1296   call TEST_ONE_ROUND
1297   jmp main
1298
1299 main:
1300   mov dx, offset string1
1301   mov cx, strlen
1302   call TEST_ONE_ROUND
1303   jmp main
1304
1305 main:
1306   mov dx, offset string1
1307   mov cx, strlen
1308   call TEST_ONE_ROUND
1309   jmp main
1310
1311 main:
1312   mov dx, offset string1
1313   mov cx, strlen
1314   call TEST_ONE_ROUND
1315   jmp main
1316
1317 main:
1318   mov dx, offset string1
1319   mov cx, strlen
1320   call TEST_ONE_ROUND
1321   jmp main
1322
1323 main:
1324   mov dx, offset string1
1325   mov cx, strlen
1326   call TEST_ONE_ROUND
1327   jmp main
1328
1329 main:
1330   mov dx, offset string1
1331   mov cx, strlen
1332   call TEST_ONE_ROUND
1333   jmp main
1334
1335 main:
1336   mov dx, offset string1
1337   mov cx, strlen
1338   call TEST_ONE_ROUND
1339   jmp main
1340
1341 main:
1342   mov dx, offset string1
1343   mov cx, strlen
1344   call TEST_ONE_ROUND
1345   jmp main
1346
1347 main:
1348   mov dx, offset string1
1349   mov cx, strlen
1350   call TEST_ONE_ROUND
1351   jmp main
1352
1353 main:
1354   mov dx, offset string1
1355   mov cx, strlen
1356   call TEST_ONE_ROUND
1357   jmp main
1358
1359 main:
1360   mov dx, offset string1
1361   mov cx, strlen
1362   call TEST_ONE_ROUND
1363   jmp main
1364
1365 main:
1366   mov dx, offset string1
1367   mov cx, strlen
1368   call TEST_ONE_ROUND
1369   jmp main
1370
1371 main:
1372   mov dx, offset string1
1373   mov cx, strlen
1374   call TEST_ONE_ROUND
1375   jmp main
1376
1377 main:
1378   mov dx, offset string1
1379   mov cx, strlen
1380   call TEST_ONE_ROUND
1381   jmp main
1382
1383 main:
1384   mov dx, offset string1
1385   mov cx, strlen
1386   call TEST_ONE_ROUND
1387   jmp main
1388
1389 main:
1390   mov dx, offset string1
1391   mov cx, strlen
1392   call TEST_ONE_ROUND
1393   jmp main
1394
1395 main:
1396   mov dx, offset string1
1397   mov cx, strlen
1398   call TEST_ONE_ROUND
1399   jmp main
1400
1401 main:
1402   mov dx, offset string1
1403   mov cx, strlen
1404   call TEST_ONE_ROUND
1405   jmp main
1406
1407 main:
1408   mov dx, offset string1
1409   mov cx, strlen
1410   call TEST_ONE_ROUND
1411   jmp main
1412
1413 main:
1414   mov dx, offset string1
1415   mov cx, strlen
1416   call TEST_ONE_ROUND
1417   jmp main
1418
1419 main:
1420   mov dx, offset string1
1421   mov cx, strlen
1422   call TEST_ONE_ROUND
1423   jmp main
1424
1425 main:
1426   mov dx, offset string1
1427   mov cx, strlen
1428   call TEST_ONE_ROUND
1429   jmp main
1430
1431 main:
1432   mov dx, offset string1
1433   mov cx, strlen
1434   call TEST_ONE_ROUND
1435   jmp main
1436
1437 main:
1438   mov dx, offset string1
1439   mov cx, strlen
1440   call TEST_ONE_ROUND
1441   jmp main
1442
1443 main:
1444   mov dx, offset string1
1445   mov cx, strlen
1446   call TEST_ONE_ROUND
1447   jmp main
1448
1449 main:
1450   mov dx, offset string1
1451   mov cx, strlen
1452   call TEST_ONE_ROUND
1453   jmp main
1454
1455 main:
1456   mov dx, offset string1
1457   mov cx, strlen
1458   call TEST_ONE_ROUND
1459   jmp main
1460
1461 main:
1462   mov dx, offset string1
1463   mov cx, strlen
1464   call TEST_ONE_ROUND
1465   jmp main
1466
1467 main:
1468   mov dx, offset string1
1469   mov cx, strlen
1470   call TEST_ONE_ROUND
1471   jmp main
1472
1473 main:
1474   mov dx, offset string1
1475   mov cx, strlen
1476   call TEST_ONE_ROUND
1477   jmp main
1478
1479 main:
1480   mov dx, offset string1
1481   mov cx, strlen
1482   call TEST_ONE_ROUND
1483   jmp main
1484
1485 main:
1486   mov dx, offset string1
1487   mov cx, strlen
1488   call TEST_ONE_ROUND
1489   jmp main
1490
1491 main:
1492   mov dx, offset string1
1493   mov cx, strlen
1494   call TEST_ONE_ROUND
1495   jmp main
1496
1497 main:
1498   mov dx, offset string1
1499   mov cx, strlen
1500   call TEST_ONE_ROUND
1501   jmp main
1502
1503 main:
1504   mov dx, offset string1
1505   mov cx, strlen
1506   call TEST_ONE_ROUND
1507   jmp main
1508
1509 main:
1510   mov dx, offset string1
1511   mov cx, strlen
1512   call TEST_ONE_ROUND
1513   jmp main
1514
1515 main:
1516   mov dx, offset string1
1517   mov cx, strlen
1518   call TEST_ONE_ROUND
1519   jmp main
1520
1521 main:
1522   mov dx, offset string1
1523   mov cx, strlen
1524   call TEST_ONE_ROUND
1525   jmp main
1526
1527 main:
1528   mov dx, offset string1
1529   mov cx, strlen
1530   call TEST_ONE
```

第 3 题: strsearch

```
strsearch.asm X
1 stack1 segment para stack
2     stack_area dw 100h dup(?)
3     stack_bottom equ $ - stack_area
4 stack1 ends
5
6
7 data1 segment para
8     input_str db 30, ?, 30 dup('$')
9     tar_char db '#' ; 待查找字符, 放入a1
10    next_line db 0dh, 0ah, '$'
11    result db 5 dup(?), '$'
12 data1 ends
13
14
15 GET_STR macro ; 字符串offset存在dx中
16     mov ah, 0ah
17     int 21h
18 endm
19
20 PRINT_STR macro ; 字符串offset存在dx中
21     mov ah, 09h
22     int 21h
23 endm
24
25
26 code1 segment para
27     assume cs:code1, ds:data1, es:data1, ss:stack1
28
29 Main proc far
30     mov ax, data1
31     mov ds, ax
```

```
jsdos10:39:59 AM X
[ ] pause [x] sound Worker
D:\>masm D:\test.asm;
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

Invoking: ML.EXE /I. /Zm /c /Ta D:\test.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: D:\test.asm

D:\>link D:\test;

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

D:\>D:\test
asd###dfw# #d
5
D:\>test
asdfqwrs
0
D:\>{
```

去除了次数的的前导零, 进行了两轮测试 (非零/零)

第 4 题: dictionary

```
dictionary.asm X
1 stack1 segment para stack
2     stack_area dw 100h dup(?)
3     stack_bottom equ $ - stack_area
4 stack1 ends
5
6
7 data1 segment para
8     buffer db 16 dup(0)
9     si_store dw ?
10    di_store dw ?
11    new_line db 0dh, 0ah, '$'
12    sentence1 db '[before sort]', 0dh, 0ah, '$'
13    sentence2 db '[after sort]', 0dh, 0ah, '$'
14    sentence3 db '[after insert]', 0dh, 0ah, '$'
15
16    my_word db 16, ?, 16 dup('$'), 12 dup(0)
17
18    list_len dw 6
19    words_list db 9, 'crescent', '$'
20               db 12, 'attenuation', '$'
21               db 8, 'persona', '$'
22               db 4, 'ebb', '$'
23               db 9, 'yachting', '$'
24               db 9, 'paradigm', '$'
25    next_word_off equ $ - words_list
26 data1 ends
27
28
29 GET_STR macro ; 字符串offset存在dx中
30     mov ah, 0ah
31     int 21h
```

```
jsdos10:26:37 AM X
[ ] pause [x] sound Worker
Assembling: D:\test.asm

D:\>link D:\test;

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

D:\>D:\test
[before sort]
crescent attenuation persona ebb yachting paradigm
[after sort]
attenuation crescent ebb paradigm persona yachting
[after insert]
attenuation crescent ebb paradigm persona xiaoruibin yachting

D:\>test
[before sort]
crescent attenuation persona ebb yachting paradigm
[after sort]
attenuation crescent ebb paradigm persona yachting
[after insert]
Xiaoruibin attenuation crescent ebb paradigm persona yachting

D:\>{
```

进行了两轮测试, 区分大小写

手写程序

第1题: strmove

strmove.asm

```
stack1 segment para stack
    stack-area dw 100h dup(?)
    stack-bottom equ $-stack-area
stack1 ends
```

```
stack1 data
data1 segment para
    null-str1 db 16 dup('0')
    string1 db "XiaRuibin"
    strlen equ $-string1
    null-str2 db 16 dup('0')
    new-line db 0dh, 0ah, '$'
data1 ends
```

PRINT-CHAR macro

```
    mov ah, 02h
    int 21h
endm
```

PRINT-STR macro

```
    mov ah, 09h
    int 21h
endm
```

TEST-ONE-ROUND macro addroff

```
    mov si, offset string1
    mov cx, strlen
    call PrintStr
    mov dl, ':'
    PRINT-CHAR
```

```
    mov si, offset string1
```

~~mov di, si~~

```
    mov di, offset string1+addroff
```

```
    mov cx, strlen
```

```
    call MemMove
```

```
    mov si, offset string1
```

```
    mov cx, strlen
```

```
    call PrintStr
```

```
    mov dl, ','
```

```
    PRINT-CHAR
```

```
    mov si, offset string1+addroff
```

```
    mov cx, strlen
```

```
    call PrintStr
```

```
    lea dx, new-line
```

```
    PRINT-STR
```

```
endm
```

```
code1 segment para
```

```
    assume ds:data1, es:data1,
           cs:code1, ss:stack1
```

```
Main proc far
```

```
    mov ax, data1
```

```
    mov ds, ax
```

```
    mov es, ax
```

```
    mov ax, stack1
```

```
    mov ss, ax
```

```
    mov sp, stack-bottom
```

- TEST-ONE-ROUND strlen+2
 - TEST-ONE-ROUND -strlen+3
 - TEST-ONE-ROUND strlen-4
- exit ... 相关相同操作

(续)

; PrintStr(si, cx) 打印从 si 开始的 cx 个字符

PrintStr proc

push ax

push dx

lp-print:
{ mov dl, [si]
PRINT-CHAR
inc si
loop lp-print

pop dx

pop ax

ret

PrintStr proc

; MemMove(si, di, cx) 将 si 处的 cx 个字符移动到 di 处

MemMove proc

cld

rep movsb

ret

MemMove endp

code1 ends

end Main

第2题: strcmp

strcmp.asm

stack1 (与之前相同)
stack1 ends

data1 segment para

string1 db 'XiabinRui', '\$'

strlen equ \$-string1-1

string2 db 20, ?, 20 dup('\$') ; '\$' 比大写字母, 数字都小, 不影响比较

new-line db 0dh, 0ah, '\$'

data1 ends

~~GETSTR macro~~ ~~与之前相同~~ ah, 0ah

GET-STR macro

mov ah, 0ah

int 21h

endm

~~PRINT-STR macro~~

mov ah, 09h

int 21h

endm

PRINT-CHAR macro

mov ah, 02h

int 21h

endm

code1 segment para

assume cs:code1, ds:data1, es:data1, ss:stack1

Main proc far

mov ax, data1

mov ds, ax

mov es, ax

mov ax, stack1

mov ss, ax

mov sp, stack_bottom

; 读入 string2

lea dx, string2

GET-STR

; 读入 string1

lea dx, string1

PRINT-STR

; string1 与 string2 比较

lea si, string1

lea di, string2+2

mov cx, strlen

cld

repz cmpsb

jb smaller

ja larger

xor ah, ah

mov al, byte ptr string2+1

cmp ax, strlen

ja smaller

jmp equal

smaller:

mov dl, '<'

jmp print_result

larger:

mov dl, '>'

jmp print_result

equal:

mov dl, '='

print_result:

PRINT-CHAR

; 读入 string2

lea dx, string2+2

PRINT-STR

exit: ~~move~~

退出操作

code1 ends

end Main