; Course Project - Option B: Average 5 grades, display min, max, and average

.ORIG X3000

START

1. Set R2 to 0
2. Add 5 to R2
3. Store R2 in memory location SCORESENTERED
4. Load base address from BASE into R6
5. Load the prompt string address into R0
6. Print the prompt strings
7. Set R4 to 0
8. Store R4 in TESTSCORE
9. Store R4 in CHAR1
10. Store R4 in CHAR2
11. Store R4 in CHAR3
12. Add 3 to R4 (start with 3 characters to enter)
13. Get character input into R0
14. Set R1 to 0
15. Copy R0 to R1
16. Load -45 into R2
17. Add R2 to R1
18. If result is 0, go to CANTBENEG
19. Load -48 into R2
20. Add R2 to R1
21. If result is negative, go to BADCHAR
22. Load -58 into R2
23. Add R2 to R1
24. If result is 0 or positive, go to BADCHAR
25. Load -48 into R2
26. Add R2 to R1 (convert ASCII to digit)
27. Store R1 in CHAR1
28. Load 48 into R2
29. Load value from CHAR1 into R0
30. Add R2 to R0 (convert digit back to ASCII for echo)
31. Print character
32. Subtract 1 from R4 (now 2 characters left)
33. Get character input into R0
34. Set R1 to 0
35. Copy R0 to R1
36. Load -10 into R2
37. Add R2 to R1
38. If result is 0, go to VALIDNUMCHAR
39. Load -32 into R2
40. Add R2 to R1
41. If result is 0, go to VALIDNUMCHAR
42. Load -48 into R2
43. Add R2 to R1
44. If result is negative, go to BADCHAR
45. Load -58 into R2
46. Add R2 to R1
47. If result is 0 or positive, go to BADCHAR
48. Load -48 into R2
49. Add R2 to R1 (convert ASCII to digit)
50. Store R1 in CHAR2
51. Load 48 into R2
52. Load value from CHAR2 into R0
53. Add R2 to R0
54. Print character
55. Subtract 1 from R4 (1 character left)
56. Get character input into R0
57. Set R1 to 0
58. Copy R0 to R1
59. Load -10 into R2
60. Add R2 to R1
61. If result is 0, go to VALIDNUMCHAR
62. Load -32 into R2
63. Add R2 to R1
64. oif result is 0, go to VALIDNUMCHAR
65. Load -48 into R2
66. Add R2 to R1
67. If result is negative, go to BADCHAR
68. Load -58 into R2
69. Add R2 to R1
70. If result is 0 or positive, go to BADCHAR
71. Load -48 into R2
72. Add R2 to R1 (convert ASCII to digit)
73. Store R1 in CHAR3
74. Load 48 into R2
75. Load value from CHAR3 into R0
76. Add R2 to R0
77. Print character
78. Subtract 1 from R4 (now 0 characters left)
79. Load newline character into R0
80. Print newline
81. jump to subroutine PREPNUMCHAR
82. Subtract 3 from R3
83. If result is 0 or positive, go to CONTINUE
84. Else, go to CHARQTYERR
85. Jump to subroutine CALCULATESCORE
86. Jump to subroutine DISPLAYRESULT
87. HALT
88. Load address of BADCHAR\_MSG into R0
89. Print BADCHAR\_MSG
90. HALT
91. Load address of NEGATIVE\_MSG into R0
92. Print NEGATIVE\_MSG
93. HALT
94. Load address of CHARQTY\_MSG into R0
95. Print CHARQTY\_MSG
96. HALT
97. Label PREPNUMCHAR
98. Load CHAR1 into R1
99. Multiply R1 by 100 (shift left, then add)
100. Load CHAR2 into R2
101. Multiply R2 by 10 (shift and add)
102. Add R2 to R1
103. Load CHAR3 into R3
104. Add R3 to R1
105. Store R1 into TESTSCORE
106. Return from subroutine
107. Label CALCULATESCORE (RET)
108. Load TESTSCORE into R1
109. Load SCORESENTERED into R2
110. Add R1 to R2
111. Store result in SCORESENTERED
112. Return from subroutine (RET)
113. Label DISPLAYRESULT
114. Load address of RESULT\_MSG into R0
115. Print RESULT\_MSG
116. Load SCORESENTERED into R1
117. Convert R1 to ASCII string (loop or call helper if needed)
118. Print value in R1 (character-by-character)
119. Load newline into R0
120. Print newline
121. Return from subroutine (RET
122. Fill labels W/ placement numbers