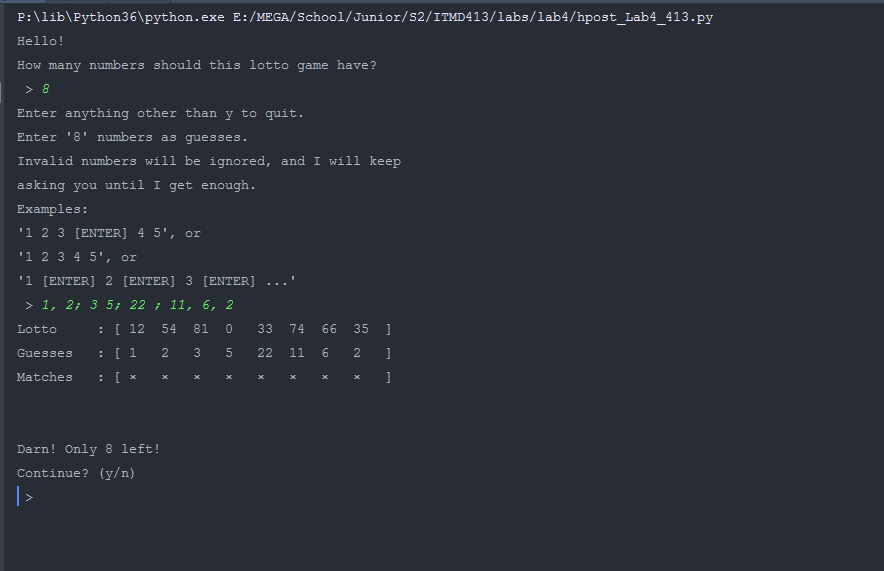
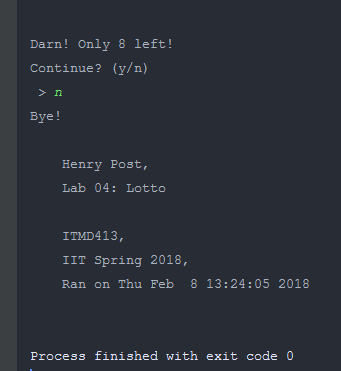
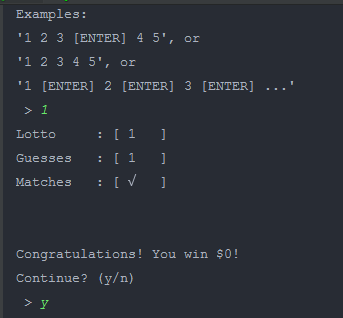
Henry Post

ITMD413

Lab 4, Lotto







1. **import** time
2. **from** random **import** \*
4. \_\_info\_\_ = \
5. f"""
6. Henry Post,
7. Lab 04: Lotto
9. ITMD413,
10. IIT Spring 2018,
11. Ran on {time.strftime("%c")}
12. """
14. \_\_cont = 'y'
15. \_\_sd = {True: "√", False: "×"}

18. **def** generate\_nums(bot: int = 0, top: int = 99, n: int = 3) -> {}:
19. """
20. Generate an {n}-long list of numbers between {bot} and {top}.
21. """
22. **return** [randint(bot, top) **for** i **in** range(n)]

25. **def** prompt\_guesses(bot: int = 0, top: int = 99, n: int = 3) -> {}:
26. """
27. Generate an {n}-long list of numbers between {bot} and {top} from user input.
28. """
29. **print**(f"Enter '{n}' numbers as guesses.\n"
30. f"Invalid numbers will be ignored, and I will keep\n"
31. f"asking you until I get enough.\n"
32. f"Examples: \n"
33. f"'1 2 3 [ENTER] 4 5', or \n"
34. f"'1 2 3 4 5', or \n"
35. f"'1 [ENTER] 2 [ENTER] 3 [ENTER] ...'")
37. nums = []
38. inp = None
40. # keep asking them until list is large enough
41. **while** len(nums) < n:
42. inp = input(" > ")
43. inp = inp.replace(",", " ").replace(";", " ")  # strip away undesired chars
45. # if they enter a list of numbers
46. **if** len(inp.split(" ")) > 1:
47. # print(f'Their list: {str(inp.split(" "))}')
49. # go through their list
50. idx = 0
51. **for** item **in** inp.split(" "):
52. item = item.replace(" ", "")  # strip whitespace
54. # print(f"item {idx} is {item}")
56. # try to treat it as a number
57. **try**:
58. **if** int(item) >= bot **and** int(item) <= top:
59. nums.append(int(item))
60. idx += 1
61. **else**:
62. **print**(f"'{inp}' is not between {bot} and {top}")
63. **except**:
64. **if** len(item) > 0:  # if it isn't an empty string
65. **print**(f"'{item}' is not a whole number.")
67. **if** idx > 0:
68. **print**(f"However, we did get your previous {idx} items and any after it.")
69. idx = 0
71. # else, assume they have entered one number.
72. **else**:
73. **try**:
74. **if** int(inp) >= bot **and** int(inp) <= top:
75. nums.append(int(inp))
76. **else**:
77. **print**(f"'{inp}' is not between {bot} and {top}")
78. **except**:
79. **print**(f"'{inp}' is not a whole number.")
81. # if they're partially done, show them what they got so far.
82. **if** len(nums) < n:
83. **print**(f"Guesses so far: {str(nums[0:n])}")
85. **return** nums[0:n]

88. **def** prompt\_numbers() -> int:
89. """
90. Get how many numbers lotto game should have from the user.
91. """
92. **return** int(input("How many numbers should this lotto game have?\n > "))

95. **def** prompt\_bounds() -> (int, int):
96. """
97. Get the bounds of the lotto numbers.
98. """
99. **print**("Enter the lower and upper bounds that this game should have, separated by a space.")
101. inp = input().split(" ")
102. **return** (int(inp[0]), int(inp[1]),)

105. **def** check\_matches(guesses, actual) -> {}:
106. """
107. Checks a list of guesses against the actual results.
108. :param guesses: List of guessed numbers.
109. :param actual: List of lotto's result numbers.
110. :return: List of True/False that tells you which numbers match.
111. """
112. **return** [guesses[i] == actual[i] **for** i **in** range(len(guesses))]

115. **def** \_\_main\_\_():
116. """
117. Main function.
118. """
119. inp = "y"
121. **print**("Hello!")
122. n = None
124. **while** True:
125. **try**:
126. n = prompt\_numbers()
127. **except**:
128. **print**("Input a valid number of lotto balls.")
129. **else**:
130. **if** n <= 0:
131. **print**("Input a positive number of lotto balls.")
132. **else**:
133. **break**
135. **while** True:
136. **try**:
137. b = prompt\_bounds()
138. bot = b[0]
139. top = b[1]
140. **except**:
141. **print**("Input valid lotto top & bottom.")
142. **else**:
143. **if** top <= bot:
144. **print**("Top can't be lower than bottom, and vice-versa.")
145. **else**:
146. **break**


150. **print**(f"Enter anything other than {\_\_cont} to quit.")
152. **while** len(inp) > 0 **and** inp[0].upper() == \_\_cont.upper():
153. correct = 0  # assume they are not a winner
155. lotto = generate\_nums(bot, top, n)  # generate some numbers
156. guesses = prompt\_guesses(bot, top, n)  # ask them to guess
157. matches = check\_matches(guesses, lotto)  # generate T/F array
159. maxlen = len(str(top))  # to format things
160. formats = (" {:^" + str(maxlen) + "} ")  # format specifier for one number.
161. # Will make things look like:
162. # [  3   2   4   5  ]
163. prefs = ("{:<10}: [")  # format specifier for 'Tag    :  ['
165. # present them with their results
167. # print the lotto numbers
168. **print**(prefs.format("Lotto"), end='')
169. **for** i **in** range(len(lotto)):
170. **print**(formats.format(lotto[i]), end='')
171. **print**(" ]")
173. # print their numbers
174. **print**(prefs.format("Guesses"), end='')
175. **for** i **in** range(len(guesses)):
176. **print**(formats.format(guesses[i]), end='')
177. **print**(" ]")
179. # print which are correct
180. **print**(prefs.format("Matches"), end='')
181. **for** i **in** range(len(matches)):
182. **print**(formats.format(\_\_sd[matches[i]]), end='')
183. **if**(matches[i]):
184. correct += 1
185. **print**(" ]\n\n")
187. **if** correct >= n:
188. **print**("Congratulations! You win $0!")
189. **else**:
190. **print**(f"Darn! Only {n-correct} left!")
192. **print**(f"Continue? ({\_\_cont}/n)")
193. inp = input(" > ")
195. **print**("Bye!")
197. **print**(\_\_info\_\_)

200. \_\_main\_\_()