1000 Python Practice Questions + Concise Solutions

Lists • Tuples • Sets • Dictionaries • Mixed/Advanced

Part A — Questions

Lists

Q1. Get the last element of list a = [1, 2, 2, 3, 1].

Q2. Reverse list a = [1, 2, 2, 3, 1] in-place.

Q3. Return a sorted copy of a = [1, 2, 2, 3, 1].

Q4. Append 1 to list a = [1, 2, 2, 3, 1].

Q5. Remove first occurrence of 1 from a = [1, 2, 2, 3, 1].

Q6. Get slice of a = [1, 2, 2, 3, 1] from index 1 to 3.

Q7. Compute sum of a = [1, 2, 2, 3, 1].

Q8. Find unique elements of a = [1, 2, 2, 3, 1] preserving order.

Q9. Merge two lists a=[1, 2, 2, 3, 1] and b=[2, 3, 4, 6, 2].

Q10. Find intersection of lists a=[1, 2, 2, 3, 1] and b=[2, 3, 4, 6, 2].

Q11. Count occurrences of 1 in a = [1, 2, 2, 3, 1].

Q12. Replace index 2 in a=[1, 2, 2, 3, 1] with 99.

Q13. Get even numbers from nums = [1, 2, 3, 4, 5, 6].

Q14. Square each number in nums = [1, 2, 3, 4, 5, 6].

Q15. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q16. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6].

Q17. Rotate list a=[1, 2, 2, 3, 1] right by 2.

Q18. Get second largest from nums = [1, 2, 3, 4, 5, 6].

Q19. Remove None values from [1,None,2,None,3].

Q20. Zip [1, 2, 3, 4, 5, 6] and ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] into list of tuples.

Q21. Get the last element of list a = [2, 3, 4, 6, 2].

Q22. Reverse list a = [2, 3, 4, 6, 2] in-place.

Q23. Return a sorted copy of a = [2, 3, 4, 6, 2].

Q24. Append 2 to list a = [2, 3, 4, 6, 2].

Q25. Remove first occurrence of 2 from a = [2, 3, 4, 6, 2].

Q26. Get slice of a = [2, 3, 4, 6, 2] from index 1 to 3.

Q27. Compute sum of a = [2, 3, 4, 6, 2].

Q28. Find unique elements of a = [2, 3, 4, 6, 2] preserving order.

Q29. Merge two lists a=[2, 3, 4, 6, 2] and b=[3, 4, 6, 9, 3].

Q30. Find intersection of lists a=[2, 3, 4, 6, 2] and b=[3, 4, 6, 9, 3].

Q31. Count occurrences of 2 in a = [2, 3, 4, 6, 2].

Q32. Replace index 2 in a=[2, 3, 4, 6, 2] with 99.

Q33. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7].

Q34. Square each number in nums = [1, 2, 3, 4, 5, 6, 7].

Q35. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q36. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7].

Q37. Rotate list a=[2, 3, 4, 6, 2] right by 2.

Q38. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7].

Q39. Remove None values from [1,None,2,None,3].

Q40. Zip [1, 2, 3, 4, 5, 6, 7] and ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] into list of tuples.

Q41. Get the last element of list a = [3, 4, 6, 9, 3].

Q42. Reverse list a = [3, 4, 6, 9, 3] in-place.

Q43. Return a sorted copy of a = [3, 4, 6, 9, 3].

Q44. Append 3 to list a = [3, 4, 6, 9, 3].

Q45. Remove first occurrence of 3 from a = [3, 4, 6, 9, 3].

Q46. Get slice of a = [3, 4, 6, 9, 3] from index 1 to 3.

Q47. Compute sum of a = [3, 4, 6, 9, 3].

Q48. Find unique elements of a = [3, 4, 6, 9, 3] preserving order.

Q49. Merge two lists a=[3, 4, 6, 9, 3] and b=[4, 5, 8, 12, 4].

Q50. Find intersection of lists a=[3, 4, 6, 9, 3] and b=[4, 5, 8, 12, 4].

Q51. Count occurrences of 3 in a = [3, 4, 6, 9, 3].

Q52. Replace index 2 in a=[3, 4, 6, 9, 3] with 99.

Q53. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q54. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q55. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q56. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q57. Rotate list a=[3, 4, 6, 9, 3] right by 2.

Q58. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q59. Remove None values from [1,None,2,None,3].

Q60. Zip [1, 2, 3, 4, 5, 6, 7, 8] and ['delta', 'epsilon', 'zeta', 'eta', 'theta'] into list of tuples.

Q61. Get the last element of list a = [4, 5, 8, 12, 4].

Q62. Reverse list a = [4, 5, 8, 12, 4] in-place.

Q63. Return a sorted copy of a = [4, 5, 8, 12, 4].

Q64. Append 4 to list a = [4, 5, 8, 12, 4].

Q65. Remove first occurrence of 4 from a = [4, 5, 8, 12, 4].

Q66. Get slice of a = [4, 5, 8, 12, 4] from index 1 to 3.

Q67. Compute sum of a = [4, 5, 8, 12, 4].

Q68. Find unique elements of a = [4, 5, 8, 12, 4] preserving order.

Q69. Merge two lists a=[4, 5, 8, 12, 4] and b=[5, 6, 10, 2, 0].

Q70. Find intersection of lists a=[4, 5, 8, 12, 4] and b=[5, 6, 10, 2, 0].

Q71. Count occurrences of 4 in a = [4, 5, 8, 12, 4].

Q72. Replace index 2 in a=[4, 5, 8, 12, 4] with 99.

Q73. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q74. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q75. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q76. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q77. Rotate list a=[4, 5, 8, 12, 4] right by 2.

Q78. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q79. Remove None values from [1,None,2,None,3].

Q80. Zip [1, 2, 3, 4, 5, 6, 7, 8, 9] and ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] into list of tuples.

Q81. Get the last element of list a = [5, 6, 10, 2, 0].

Q82. Reverse list a = [5, 6, 10, 2, 0] in-place.

Q83. Return a sorted copy of a = [5, 6, 10, 2, 0].

Q84. Append 5 to list a = [5, 6, 10, 2, 0].

Q85. Remove first occurrence of 0 from a = [5, 6, 10, 2, 0].

Q86. Get slice of a = [5, 6, 10, 2, 0] from index 1 to 3.

Q87. Compute sum of a = [5, 6, 10, 2, 0].

Q88. Find unique elements of a = [5, 6, 10, 2, 0] preserving order.

Q89. Merge two lists a=[5, 6, 10, 2, 0] and b=[6, 7, 1, 5, 1].

Q90. Find intersection of lists a=[5, 6, 10, 2, 0] and b=[6, 7, 1, 5, 1].

Q91. Count occurrences of 5 in a = [5, 6, 10, 2, 0].

Q92. Replace index 2 in a=[5, 6, 10, 2, 0] with 99.

Q93. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q94. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q95. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q96. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q97. Rotate list a=[5, 6, 10, 2, 0] right by 2.

Q98. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q99. Remove None values from [1,None,2,None,3].

Q100. Zip [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and ['zeta', 'eta', 'theta', 'alpha', 'beta'] into list of tuples.

Q101. Get the last element of list a = [6, 7, 1, 5, 1].

Q102. Reverse list a = [6, 7, 1, 5, 1] in-place.

Q103. Return a sorted copy of a = [6, 7, 1, 5, 1].

Q104. Append 6 to list a = [6, 7, 1, 5, 1].

Q105. Remove first occurrence of 1 from a = [6, 7, 1, 5, 1].

Q106. Get slice of a = [6, 7, 1, 5, 1] from index 1 to 3.

Q107. Compute sum of a = [6, 7, 1, 5, 1].

Q108. Find unique elements of a = [6, 7, 1, 5, 1] preserving order.

Q109. Merge two lists a=[6, 7, 1, 5, 1] and b=[0, 8, 3, 8, 2].

Q110. Find intersection of lists a=[6, 7, 1, 5, 1] and b=[0, 8, 3, 8, 2].

Q111. Count occurrences of 6 in a = [6, 7, 1, 5, 1].

Q112. Replace index 2 in a=[6, 7, 1, 5, 1] with 99.

Q113. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q114. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q115. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q116. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q117. Rotate list a=[6, 7, 1, 5, 1] right by 2.

Q118. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q119. Remove None values from [1,None,2,None,3].

Q120. Zip [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] and ['eta', 'theta', 'alpha', 'beta', 'gamma'] into list of tuples.

Q121. Get the last element of list a = [0, 8, 3, 8, 2].

Q122. Reverse list a = [0, 8, 3, 8, 2] in-place.

Q123. Return a sorted copy of a = [0, 8, 3, 8, 2].

Q124. Append 7 to list a = [0, 8, 3, 8, 2].

Q125. Remove first occurrence of 2 from a = [0, 8, 3, 8, 2].

Q126. Get slice of a = [0, 8, 3, 8, 2] from index 1 to 3.

Q127. Compute sum of a = [0, 8, 3, 8, 2].

Q128. Find unique elements of a = [0, 8, 3, 8, 2] preserving order.

Q129. Merge two lists a=[0, 8, 3, 8, 2] and b=[1, 0, 5, 11, 3].

Q130. Find intersection of lists a=[0, 8, 3, 8, 2] and b=[1, 0, 5, 11, 3].

Q131. Count occurrences of 0 in a = [0, 8, 3, 8, 2].

Q132. Replace index 2 in a=[0, 8, 3, 8, 2] with 99.

Q133. Get even numbers from nums = [1, 2, 3, 4, 5].

Q134. Square each number in nums = [1, 2, 3, 4, 5].

Q135. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q136. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5].

Q137. Rotate list a=[0, 8, 3, 8, 2] right by 2.

Q138. Get second largest from nums = [1, 2, 3, 4, 5].

Q139. Remove None values from [1,None,2,None,3].

Q140. Zip [1, 2, 3, 4, 5] and ['theta', 'alpha', 'beta', 'gamma', 'delta'] into list of tuples.

Q141. Get the last element of list a = [1, 0, 5, 11, 3].

Q142. Reverse list a = [1, 0, 5, 11, 3] in-place.

Q143. Return a sorted copy of a = [1, 0, 5, 11, 3].

Q144. Append 8 to list a = [1, 0, 5, 11, 3].

Q145. Remove first occurrence of 3 from a = [1, 0, 5, 11, 3].

Q146. Get slice of a = [1, 0, 5, 11, 3] from index 1 to 3.

Q147. Compute sum of a = [1, 0, 5, 11, 3].

Q148. Find unique elements of a = [1, 0, 5, 11, 3] preserving order.

Q149. Merge two lists a=[1, 0, 5, 11, 3] and b=[2, 1, 7, 1, 4].

Q150. Find intersection of lists a=[1, 0, 5, 11, 3] and b=[2, 1, 7, 1, 4].

Q151. Count occurrences of 1 in a = [1, 0, 5, 11, 3].

Q152. Replace index 2 in a=[1, 0, 5, 11, 3] with 99.

Q153. Get even numbers from nums = [1, 2, 3, 4, 5, 6].

Q154. Square each number in nums = [1, 2, 3, 4, 5, 6].

Q155. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q156. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6].

Q157. Rotate list a=[1, 0, 5, 11, 3] right by 2.

Q158. Get second largest from nums = [1, 2, 3, 4, 5, 6].

Q159. Remove None values from [1,None,2,None,3].

Q160. Zip [1, 2, 3, 4, 5, 6] and ['alpha', 'beta', 'gamma', 'delta', 'epsilon'] into list of tuples.

Q161. Get the last element of list a = [2, 1, 7, 1, 4].

Q162. Reverse list a = [2, 1, 7, 1, 4] in-place.

Q163. Return a sorted copy of a = [2, 1, 7, 1, 4].

Q164. Append 9 to list a = [2, 1, 7, 1, 4].

Q165. Remove first occurrence of 4 from a = [2, 1, 7, 1, 4].

Q166. Get slice of a = [2, 1, 7, 1, 4] from index 1 to 3.

Q167. Compute sum of a = [2, 1, 7, 1, 4].

Q168. Find unique elements of a = [2, 1, 7, 1, 4] preserving order.

Q169. Merge two lists a=[2, 1, 7, 1, 4] and b=[3, 2, 9, 4, 0].

Q170. Find intersection of lists a=[2, 1, 7, 1, 4] and b=[3, 2, 9, 4, 0].

Q171. Count occurrences of 2 in a = [2, 1, 7, 1, 4].

Q172. Replace index 2 in a=[2, 1, 7, 1, 4] with 99.

Q173. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7].

Q174. Square each number in nums = [1, 2, 3, 4, 5, 6, 7].

Q175. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q176. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7].

Q177. Rotate list a=[2, 1, 7, 1, 4] right by 2.

Q178. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7].

Q179. Remove None values from [1,None,2,None,3].

Q180. Zip [1, 2, 3, 4, 5, 6, 7] and ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] into list of tuples.

Q181. Get the last element of list a = [3, 2, 9, 4, 0].

Q182. Reverse list a = [3, 2, 9, 4, 0] in-place.

Q183. Return a sorted copy of a = [3, 2, 9, 4, 0].

Q184. Append 0 to list a = [3, 2, 9, 4, 0].

Q185. Remove first occurrence of 0 from a = [3, 2, 9, 4, 0].

Q186. Get slice of a = [3, 2, 9, 4, 0] from index 1 to 3.

Q187. Compute sum of a = [3, 2, 9, 4, 0].

Q188. Find unique elements of a = [3, 2, 9, 4, 0] preserving order.

Q189. Merge two lists a=[3, 2, 9, 4, 0] and b=[4, 3, 0, 7, 1].

Q190. Find intersection of lists a=[3, 2, 9, 4, 0] and b=[4, 3, 0, 7, 1].

Q191. Count occurrences of 3 in a = [3, 2, 9, 4, 0].

Q192. Replace index 2 in a=[3, 2, 9, 4, 0] with 99.

Q193. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q194. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q195. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q196. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q197. Rotate list a=[3, 2, 9, 4, 0] right by 2.

Q198. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8].

Q199. Remove None values from [1,None,2,None,3].

Q200. Zip [1, 2, 3, 4, 5, 6, 7, 8] and ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] into list of tuples.

Q201. Get the last element of list a = [4, 3, 0, 7, 1].

Q202. Reverse list a = [4, 3, 0, 7, 1] in-place.

Q203. Return a sorted copy of a = [4, 3, 0, 7, 1].

Q204. Append 1 to list a = [4, 3, 0, 7, 1].

Q205. Remove first occurrence of 1 from a = [4, 3, 0, 7, 1].

Q206. Get slice of a = [4, 3, 0, 7, 1] from index 1 to 3.

Q207. Compute sum of a = [4, 3, 0, 7, 1].

Q208. Find unique elements of a = [4, 3, 0, 7, 1] preserving order.

Q209. Merge two lists a=[4, 3, 0, 7, 1] and b=[5, 4, 2, 10, 2].

Q210. Find intersection of lists a=[4, 3, 0, 7, 1] and b=[5, 4, 2, 10, 2].

Q211. Count occurrences of 4 in a = [4, 3, 0, 7, 1].

Q212. Replace index 2 in a=[4, 3, 0, 7, 1] with 99.

Q213. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q214. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q215. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q216. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q217. Rotate list a=[4, 3, 0, 7, 1] right by 2.

Q218. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9].

Q219. Remove None values from [1,None,2,None,3].

Q220. Zip [1, 2, 3, 4, 5, 6, 7, 8, 9] and ['delta', 'epsilon', 'zeta', 'eta', 'theta'] into list of tuples.

Q221. Get the last element of list a = [5, 4, 2, 10, 2].

Q222. Reverse list a = [5, 4, 2, 10, 2] in-place.

Q223. Return a sorted copy of a = [5, 4, 2, 10, 2].

Q224. Append 2 to list a = [5, 4, 2, 10, 2].

Q225. Remove first occurrence of 2 from a = [5, 4, 2, 10, 2].

Q226. Get slice of a = [5, 4, 2, 10, 2] from index 1 to 3.

Q227. Compute sum of a = [5, 4, 2, 10, 2].

Q228. Find unique elements of a = [5, 4, 2, 10, 2] preserving order.

Q229. Merge two lists a=[5, 4, 2, 10, 2] and b=[6, 5, 4, 0, 3].

Q230. Find intersection of lists a=[5, 4, 2, 10, 2] and b=[6, 5, 4, 0, 3].

Q231. Count occurrences of 5 in a = [5, 4, 2, 10, 2].

Q232. Replace index 2 in a=[5, 4, 2, 10, 2] with 99.

Q233. Get even numbers from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q234. Square each number in nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q235. Flatten nested list [[1,2],[3,4],[5]] in one line.

Q236. Create list of pairs (x, x^2) for nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q237. Rotate list a=[5, 4, 2, 10, 2] right by 2.

Q238. Get second largest from nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q239. Remove None values from [1,None,2,None,3].

Q240. Zip [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] into list of tuples.

Q241. Get the last element of list a = [6, 5, 4, 0, 3].

Q242. Reverse list a = [6, 5, 4, 0, 3] in-place.

Q243. Return a sorted copy of a = [6, 5, 4, 0, 3].

Q244. Append 3 to list a = [6, 5, 4, 0, 3].

Q245. Remove first occurrence of 3 from a = [6, 5, 4, 0, 3].

Q246. Get slice of a = [6, 5, 4, 0, 3] from index 1 to 3.

Q247. Compute sum of a = [6, 5, 4, 0, 3].

Q248. Find unique elements of a = [6, 5, 4, 0, 3] preserving order.

Q249. Merge two lists a=[6, 5, 4, 0, 3] and b=[0, 6, 6, 3, 4].

Q250. Find intersection of lists a=[6, 5, 4, 0, 3] and b=[0, 6, 6, 3, 4].

Tuples

Q251. Access third element of t = (1, 4, 2, 3).

Q252. Slice t = (1, 4, 2, 3) from 1 to 3.

Q253. Concatenate tuples t=(1, 4, 2, 3) and u=(2, 5, 4, 6).

Q254. Unpack t = (1, 4, 2, 3) into a,b,c,d.

Q255. Count 1 in t = (1, 4, 2, 3).

Q256. Find index of first 1 in t = (1, 4, 2, 3).

Q257. Convert list [1,2,3] to tuple.

Q258. Check if 3 is in (1,2,3).

Q259. Swap values of x=5 and y=7 using tuple unpacking.

Q260. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q261. Access third element of t = (2, 5, 4, 6).

Q262. Slice t = (2, 5, 4, 6) from 1 to 3.

Q263. Concatenate tuples t=(2, 5, 4, 6) and u=(3, 6, 6, 9).

Q264. Unpack t = (2, 5, 4, 6) into a,b,c,d.

Q265. Count 2 in t = (2, 5, 4, 6).

Q266. Find index of first 2 in t = (2, 5, 4, 6).

Q267. Convert list [1,2,3] to tuple.

Q268. Check if 3 is in (1,2,3).
Q269. Swap values of x=5 and y=7 using tuple unpacking.
Q270. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q271. Access third element of t = (3, 6, 6, 9).
Q272. Slice t = (3, 6, 6, 9) from 1 to 3.
Q273. Concatenate tuples t=(3, 6, 6, 9) and u=(4, 0, 8, 1).
Q274. Unpack t = (3, 6, 6, 9) into a,b,c,d.
Q275. Count 0 in t = (3, 6, 6, 9).
Q276. Find index of first 3 in t = (3, 6, 6, 9).
Q277. Convert list [1,2,3] to tuple.
Q278. Check if 3 is in (1,2,3).
Q279. Swap values of x=5 and y=7 using tuple unpacking.
Q280. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q281. Access third element of t = (4, 0, 8, 1).
Q282. Slice t = (4, 0, 8, 1) from 1 to 3.
Q283. Concatenate tuples t=(4, 0, 8, 1) and u=(0, 1, 1, 4).
Q284. Unpack t = (4, 0, 8, 1) into a,b,c,d.
Q285. Count 1 in t = (4, 0, 8, 1).
Q286. Find index of first 4 in t = (4, 0, 8, 1).
Q287. Convert list [1,2,3] to tuple.
Q288. Check if 3 is in (1,2,3).
Q289. Swap values of x=5 and y=7 using tuple unpacking.
Q290. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q291. Access third element of t = (0, 1, 1, 4).
Q292. Slice t = (0, 1, 1, 4) from 1 to 3.
Q293. Concatenate tuples t=(0, 1, 1, 4) and u=(1, 2, 3, 7).
Q294. Unpack t = (0, 1, 1, 4) into a,b,c,d.
Q295. Count 2 in t = (0, 1, 1, 4).
Q296. Find index of first 0 in t = (0, 1, 1, 4).
Q297. Convert list [1,2,3] to tuple.
Q298. Check if 3 is in (1,2,3).
Q299. Swap values of x=5 and y=7 using tuple unpacking.
Q300. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q301. Access third element of t = (1, 2, 3, 7).
Q302. Slice t = (1, 2, 3, 7) from 1 to 3.
Q303. Concatenate tuples t=(1, 2, 3, 7) and u=(2, 3, 5, 10).
Q304. Unpack t = (1, 2, 3, 7) into a,b,c,d.
Q305. Count 0 in t = (1, 2, 3, 7).
Q306. Find index of first 1 in t = (1, 2, 3, 7).
Q307. Convert list [1,2,3] to tuple.
Q308. Check if 3 is in (1,2,3).
Q309. Swap values of x=5 and y=7 using tuple unpacking.
Q310. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q311. Access third element of t = (2, 3, 5, 10).
Q312. Slice t = (2, 3, 5, 10) from 1 to 3.
Q313. Concatenate tuples t=(2, 3, 5, 10) and u=(3, 4, 7, 2).

Q314. Unpack t = (2, 3, 5, 10) into a,b,c,d.
Q315. Count 1 in t = (2, 3, 5, 10).
Q316. Find index of first 2 in t = (2, 3, 5, 10).
Q317. Convert list [1,2,3] to tuple.
Q318. Check if 3 is in (1,2,3).
Q319. Swap values of x=5 and y=7 using tuple unpacking.
Q320. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q321. Access third element of t = (3, 4, 7, 2).
Q322. Slice t = (3, 4, 7, 2) from 1 to 3.
Q323. Concatenate tuples t=(3, 4, 7, 2) and u=(4, 5, 0, 5).
Q324. Unpack t = (3, 4, 7, 2) into a,b,c,d.
Q325. Count 2 in t = (3, 4, 7, 2).
Q326. Find index of first 3 in t = (3, 4, 7, 2).
Q327. Convert list [1,2,3] to tuple.
Q328. Check if 3 is in (1,2,3).
Q329. Swap values of x=5 and y=7 using tuple unpacking.
Q330. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q331. Access third element of t = (4, 5, 0, 5).
Q332. Slice t = (4, 5, 0, 5) from 1 to 3.
Q333. Concatenate tuples t=(4, 5, 0, 5) and u=(0, 6, 2, 8).
Q334. Unpack t = (4, 5, 0, 5) into a,b,c,d.
Q335. Count 0 in t = (4, 5, 0, 5).
Q336. Find index of first 4 in t = (4, 5, 0, 5).
Q337. Convert list [1,2,3] to tuple.
Q338. Check if 3 is in (1,2,3).
Q339. Swap values of x=5 and y=7 using tuple unpacking.
Q340. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q341. Access third element of t = (0, 6, 2, 8).
Q342. Slice t = (0, 6, 2, 8) from 1 to 3.
Q343. Concatenate tuples t=(0, 6, 2, 8) and u=(1, 0, 4, 0).
Q344. Unpack t = (0, 6, 2, 8) into a,b,c,d.
Q345. Count 1 in t = (0, 6, 2, 8).
Q346. Find index of first 0 in t = (0, 6, 2, 8).
Q347. Convert list [1,2,3] to tuple.
Q348. Check if 3 is in (1,2,3).
Q349. Swap values of x=5 and y=7 using tuple unpacking.
Q350. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q351. Access third element of t = (1, 0, 4, 0).
Q352. Slice t = (1, 0, 4, 0) from 1 to 3.
Q353. Concatenate tuples t=(1, 0, 4, 0) and u=(2, 1, 6, 3).
Q354. Unpack t = (1, 0, 4, 0) into a,b,c,d.
Q355. Count 2 in t = (1, 0, 4, 0).
Q356. Find index of first 1 in t = (1, 0, 4, 0).
Q357. Convert list [1,2,3] to tuple.
Q358. Check if 3 is in (1,2,3).
Q359. Swap values of x=5 and y=7 using tuple unpacking.
Q360. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q361. Access third element of t = (2, 1, 6, 3).

Q362. Slice t = (2, 1, 6, 3) from 1 to 3.

Q363. Concatenate tuples t=(2, 1, 6, 3) and u=(3, 2, 8, 6).

Q364. Unpack t = (2, 1, 6, 3) into a,b,c,d.

Q365. Count 0 in t = (2, 1, 6, 3).

Q366. Find index of first 2 in t = (2, 1, 6, 3).

Q367. Convert list [1,2,3] to tuple.

Q368. Check if 3 is in (1,2,3).

Q369. Swap values of x=5 and y=7 using tuple unpacking.

Q370. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q371. Access third element of t = (3, 2, 8, 6).

Q372. Slice t = (3, 2, 8, 6) from 1 to 3.

Q373. Concatenate tuples t=(3, 2, 8, 6) and u=(4, 3, 1, 9).

Q374. Unpack t = (3, 2, 8, 6) into a,b,c,d.

Q375. Count 1 in t = (3, 2, 8, 6).

Q376. Find index of first 3 in t = (3, 2, 8, 6).

Q377. Convert list [1,2,3] to tuple.

Q378. Check if 3 is in (1,2,3).

Q379. Swap values of x=5 and y=7 using tuple unpacking.

Q380. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q381. Access third element of t = (4, 3, 1, 9).

Q382. Slice t = (4, 3, 1, 9) from 1 to 3.

Q383. Concatenate tuples t=(4, 3, 1, 9) and u=(0, 4, 3, 1).

Q384. Unpack t = (4, 3, 1, 9) into a,b,c,d.

Q385. Count 2 in t = (4, 3, 1, 9).

Q386. Find index of first 4 in t = (4, 3, 1, 9).

Q387. Convert list [1,2,3] to tuple.

Q388. Check if 3 is in (1,2,3).

Q389. Swap values of x=5 and y=7 using tuple unpacking.

Q390. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q391. Access third element of t = (0, 4, 3, 1).

Q392. Slice t = (0, 4, 3, 1) from 1 to 3.

Q393. Concatenate tuples t=(0, 4, 3, 1) and u=(1, 5, 5, 4).

Q394. Unpack t = (0, 4, 3, 1) into a,b,c,d.

Q395. Count 0 in t = (0, 4, 3, 1).

Q396. Find index of first 0 in t = (0, 4, 3, 1).

Q397. Convert list [1,2,3] to tuple.

Q398. Check if 3 is in (1,2,3).

Q399. Swap values of x=5 and y=7 using tuple unpacking.

Q400. Nested access: get 3 from t=(1,(2,3),(4,5)).

Q401. Access third element of t = (1, 5, 5, 4).

Q402. Slice t = (1, 5, 5, 4) from 1 to 3.

Q403. Concatenate tuples t=(1, 5, 5, 4) and u=(2, 6, 7, 7).

Q404. Unpack t = (1, 5, 5, 4) into a,b,c,d.

Q405. Count 1 in t = (1, 5, 5, 4).

Q406. Find index of first 1 in t = (1, 5, 5, 4).

Q407. Convert list [1,2,3] to tuple.
Q408. Check if 3 is in (1,2,3).
Q409. Swap values of x=5 and y=7 using tuple unpacking.
Q410. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q411. Access third element of t = (2, 6, 7, 7).
Q412. Slice t = (2, 6, 7, 7) from 1 to 3.
Q413. Concatenate tuples t=(2, 6, 7, 7) and u=(3, 0, 0, 10).
Q414. Unpack t = (2, 6, 7, 7) into a,b,c,d.
Q415. Count 2 in t = (2, 6, 7, 7).
Q416. Find index of first 2 in t = (2, 6, 7, 7).
Q417. Convert list [1,2,3] to tuple.
Q418. Check if 3 is in (1,2,3).
Q419. Swap values of x=5 and y=7 using tuple unpacking.
Q420. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q421. Access third element of t = (3, 0, 0, 10).
Q422. Slice t = (3, 0, 0, 10) from 1 to 3.
Q423. Concatenate tuples t=(3, 0, 0, 10) and u=(4, 1, 2, 2).
Q424. Unpack t = (3, 0, 0, 10) into a,b,c,d.
Q425. Count 0 in t = (3, 0, 0, 10).
Q426. Find index of first 3 in t = (3, 0, 0, 10).
Q427. Convert list [1,2,3] to tuple.
Q428. Check if 3 is in (1,2,3).
Q429. Swap values of x=5 and y=7 using tuple unpacking.
Q430. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q431. Access third element of t = (4, 1, 2, 2).
Q432. Slice t = (4, 1, 2, 2) from 1 to 3.
Q433. Concatenate tuples t=(4, 1, 2, 2) and u=(0, 2, 4, 5).
Q434. Unpack t = (4, 1, 2, 2) into a,b,c,d.
Q435. Count 1 in t = (4, 1, 2, 2).
Q436. Find index of first 4 in t = (4, 1, 2, 2).
Q437. Convert list [1,2,3] to tuple.
Q438. Check if 3 is in (1,2,3).
Q439. Swap values of x=5 and y=7 using tuple unpacking.
Q440. Nested access: get 3 from t=(1,(2,3),(4,5)).
Q441. Access third element of t = (0, 2, 4, 5).
Q442. Slice t = (0, 2, 4, 5) from 1 to 3.
Q443. Concatenate tuples t=(0, 2, 4, 5) and u=(1, 3, 6, 8).
Q444. Unpack t = (0, 2, 4, 5) into a,b,c,d.
Q445. Count 2 in t = (0, 2, 4, 5).
Q446. Find index of first 0 in t = (0, 2, 4, 5).
Q447. Convert list [1,2,3] to tuple.
Q448. Check if 3 is in (1,2,3).
Q449. Swap values of x=5 and y=7 using tuple unpacking.
Q450. Nested access: get 3 from t=(1,(2,3),(4,5)).

Sets

Q451. Union of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q452. Intersection of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q453. Difference a−b for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q454. Symmetric difference for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q455. Add element 1 to set a={1, 2, 3, 4, 5, 6}.

Q456. Remove element 1 from set a={1, 2, 3, 4, 5, 6} if present.

Q457. Check if a={1, 2, 3, 4, 5, 6} is subset of b={1, 2, 3, 4, 5, 6, 7, 8}.

Q458. Unique values from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q459. Set comprehension: squares of 1..5.

Q460. Find duplicates in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q461. Union of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q462. Intersection of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q463. Difference a−b for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q464. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q465. Add element 2 to set a={1, 2, 3, 4, 5, 6, 7}.

Q466. Remove element 2 from set a={1, 2, 3, 4, 5, 6, 7} if present.

Q467. Check if a={1, 2, 3, 4, 5, 6, 7} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q468. Unique values from list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q469. Set comprehension: squares of 1..5.

Q470. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q471. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q472. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q473. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q474. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q475. Add element 3 to set a={1, 2, 3, 4, 5, 6, 7, 8}.

Q476. Remove element 3 from set a={1, 2, 3, 4, 5, 6, 7, 8} if present.

Q477. Check if a={1, 2, 3, 4, 5, 6, 7, 8} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q478. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q479. Set comprehension: squares of 1..5.

Q480. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q481. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q482. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q483. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q484. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q485. Add element 4 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q486. Remove element 4 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9} if present.

Q487. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q488. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q489. Set comprehension: squares of 1..5.

Q490. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q491. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q492. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q493. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q494. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q495. Add element 5 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q496. Remove element 0 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} if present.

Q497. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} is subset of b={1, 2, 3, 4, 5}.

Q498. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q499. Set comprehension: squares of 1..5.

Q500. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q501. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q502. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q503. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q504. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q505. Add element 6 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q506. Remove element 1 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} if present.

Q507. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} is subset of b={1, 2, 3, 4, 5, 6}.

Q508. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q509. Set comprehension: squares of 1..5.

Q510. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q511. Union of sets a={1, 2, 3, 4, 5} and b={1, 2, 3, 4, 5, 6, 7}.

Q512. Intersection of sets a={1, 2, 3, 4, 5} and b={1, 2, 3, 4, 5, 6, 7}.

Q513. Difference a−b for a={1, 2, 3, 4, 5}, b={1, 2, 3, 4, 5, 6, 7}.

Q514. Symmetric difference for a={1, 2, 3, 4, 5}, b={1, 2, 3, 4, 5, 6, 7}.

Q515. Add element 7 to set a={1, 2, 3, 4, 5}.

Q516. Remove element 2 from set a={1, 2, 3, 4, 5} if present.

Q517. Check if a={1, 2, 3, 4, 5} is subset of b={1, 2, 3, 4, 5, 6, 7}.

Q518. Unique values from list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5].

Q519. Set comprehension: squares of 1..5.

Q520. Find duplicates in list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5].

Q521. Union of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q522. Intersection of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q523. Difference a−b for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q524. Symmetric difference for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q525. Add element 8 to set a={1, 2, 3, 4, 5, 6}.

Q526. Remove element 3 from set a={1, 2, 3, 4, 5, 6} if present.

Q527. Check if a={1, 2, 3, 4, 5, 6} is subset of b={1, 2, 3, 4, 5, 6, 7, 8}.

Q528. Unique values from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q529. Set comprehension: squares of 1..5.

Q530. Find duplicates in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q531. Union of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q532. Intersection of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q533. Difference a−b for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q534. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q535. Add element 9 to set a={1, 2, 3, 4, 5, 6, 7}.

Q536. Remove element 4 from set a={1, 2, 3, 4, 5, 6, 7} if present.

Q537. Check if a={1, 2, 3, 4, 5, 6, 7} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q538. Unique values from list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q539. Set comprehension: squares of 1..5.

Q540. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q541. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q542. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q543. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q544. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q545. Add element 0 to set a={1, 2, 3, 4, 5, 6, 7, 8}.

Q546. Remove element 0 from set a={1, 2, 3, 4, 5, 6, 7, 8} if present.

Q547. Check if a={1, 2, 3, 4, 5, 6, 7, 8} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q548. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q549. Set comprehension: squares of 1..5.

Q550. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q551. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q552. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q553. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q554. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q555. Add element 1 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q556. Remove element 1 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9} if present.

Q557. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q558. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q559. Set comprehension: squares of 1..5.

Q560. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q561. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q562. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q563. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q564. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q565. Add element 2 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q566. Remove element 2 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} if present.

Q567. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} is subset of b={1, 2, 3, 4, 5}.

Q568. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q569. Set comprehension: squares of 1..5.

Q570. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q571. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q572. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q573. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q574. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q575. Add element 3 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q576. Remove element 3 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} if present.

Q577. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} is subset of b={1, 2, 3, 4, 5, 6}.

Q578. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q579. Set comprehension: squares of 1..5.

Q580. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q581. Union of sets a={1, 2, 3, 4, 5} and b={1, 2, 3, 4, 5, 6, 7}.

Q582. Intersection of sets a={1, 2, 3, 4, 5} and b={1, 2, 3, 4, 5, 6, 7}.

Q583. Difference a−b for a={1, 2, 3, 4, 5}, b={1, 2, 3, 4, 5, 6, 7}.

Q584. Symmetric difference for a={1, 2, 3, 4, 5}, b={1, 2, 3, 4, 5, 6, 7}.

Q585. Add element 4 to set a={1, 2, 3, 4, 5}.

Q586. Remove element 4 from set a={1, 2, 3, 4, 5} if present.

Q587. Check if a={1, 2, 3, 4, 5} is subset of b={1, 2, 3, 4, 5, 6, 7}.

Q588. Unique values from list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5].

Q589. Set comprehension: squares of 1..5.

Q590. Find duplicates in list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5].

Q591. Union of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q592. Intersection of sets a={1, 2, 3, 4, 5, 6} and b={1, 2, 3, 4, 5, 6, 7, 8}.

Q593. Difference a−b for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q594. Symmetric difference for a={1, 2, 3, 4, 5, 6}, b={1, 2, 3, 4, 5, 6, 7, 8}.

Q595. Add element 5 to set a={1, 2, 3, 4, 5, 6}.

Q596. Remove element 0 from set a={1, 2, 3, 4, 5, 6} if present.

Q597. Check if a={1, 2, 3, 4, 5, 6} is subset of b={1, 2, 3, 4, 5, 6, 7, 8}.

Q598. Unique values from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q599. Set comprehension: squares of 1..5.

Q600. Find duplicates in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].

Q601. Union of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q602. Intersection of sets a={1, 2, 3, 4, 5, 6, 7} and b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q603. Difference a−b for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q604. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7}, b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q605. Add element 6 to set a={1, 2, 3, 4, 5, 6, 7}.

Q606. Remove element 1 from set a={1, 2, 3, 4, 5, 6, 7} if present.

Q607. Check if a={1, 2, 3, 4, 5, 6, 7} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q608. Unique values from list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q609. Set comprehension: squares of 1..5.

Q610. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].

Q611. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q612. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q613. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q614. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q615. Add element 7 to set a={1, 2, 3, 4, 5, 6, 7, 8}.

Q616. Remove element 2 from set a={1, 2, 3, 4, 5, 6, 7, 8} if present.

Q617. Check if a={1, 2, 3, 4, 5, 6, 7, 8} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q618. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q619. Set comprehension: squares of 1..5.

Q620. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8].

Q621. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q622. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9} and b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q623. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q624. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9}, b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q625. Add element 8 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9}.

Q626. Remove element 3 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9} if present.

Q627. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9} is subset of b={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q628. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q629. Set comprehension: squares of 1..5.

Q630. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Q631. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q632. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and b={1, 2, 3, 4, 5}.

Q633. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q634. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, b={1, 2, 3, 4, 5}.

Q635. Add element 9 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Q636. Remove element 4 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} if present.

Q637. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} is subset of b={1, 2, 3, 4, 5}.

Q638. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q639. Set comprehension: squares of 1..5.

Q640. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Q641. Union of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q642. Intersection of sets a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} and b={1, 2, 3, 4, 5, 6}.

Q643. Difference a−b for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q644. Symmetric difference for a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, b={1, 2, 3, 4, 5, 6}.

Q645. Add element 0 to set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Q646. Remove element 0 from set a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} if present.

Q647. Check if a={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} is subset of b={1, 2, 3, 4, 5, 6}.

Q648. Unique values from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Q649. Set comprehension: squares of 1..5.

Q650. Find duplicates in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Dictionaries

Q651. Get value for key 'k2' in d={'k1': 2, 'k2': 3, 'k3': 4} with default 0.

Q652. Update key 'k1' to 99 in d={'k1': 2, 'k2': 3, 'k3': 4}.

Q653. Merge dicts d={'k1': 2, 'k2': 3, 'k3': 4} and e={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8} (Python 3.9+).

Q654. Keys of d={'k1': 2, 'k2': 3, 'k3': 4}.

Q655. Values of d={'k1': 2, 'k2': 3, 'k3': 4}.

Q656. Items of d={'k1': 2, 'k2': 3, 'k3': 4}.

Q657. Invert dict {'a':1,'b':2} (values unique).

Q658. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q659. Max key by value in d={'a':3,'b':7,'c':5}.

Q660. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q661. Get value for key 'k2' in d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6} with default 0.

Q662. Update key 'k1' to 99 in d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6}.

Q663. Merge dicts d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6} and e={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10} (Python 3.9+).

Q664. Keys of d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6}.

Q665. Values of d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6}.

Q666. Items of d={'k1': 3, 'k2': 4, 'k3': 5, 'k4': 6}.

Q667. Invert dict {'a':1,'b':2} (values unique).

Q668. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q669. Max key by value in d={'a':3,'b':7,'c':5}.

Q670. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q671. Get value for key 'k2' in d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8} with default 0.

Q672. Update key 'k1' to 99 in d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8}.

Q673. Merge dicts d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8} and e={'k1': 6, 'k2': 7} (Python 3.9+).

Q674. Keys of d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8}.

Q675. Values of d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8}.

Q676. Items of d={'k1': 4, 'k2': 5, 'k3': 6, 'k4': 7, 'k5': 8}.

Q677. Invert dict {'a':1,'b':2} (values unique).

Q678. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q679. Max key by value in d={'a':3,'b':7,'c':5}.

Q680. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q681. Get value for key 'k2' in d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10} with default 0.

Q682. Update key 'k1' to 99 in d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10}.

Q683. Merge dicts d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10} and e={'k1': 7, 'k2': 8, 'k3': 9} (Python 3.9+).

Q684. Keys of d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10}.

Q685. Values of d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10}.

Q686. Items of d={'k1': 5, 'k2': 6, 'k3': 7, 'k4': 8, 'k5': 9, 'k6': 10}.

Q687. Invert dict {'a':1,'b':2} (values unique).

Q688. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q689. Max key by value in d={'a':3,'b':7,'c':5}.

Q690. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q691. Get value for key 'k2' in d={'k1': 6, 'k2': 7} with default 0.

Q692. Update key 'k1' to 99 in d={'k1': 6, 'k2': 7}.

Q693. Merge dicts d={'k1': 6, 'k2': 7} and e={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11} (Python 3.9+).

Q694. Keys of d={'k1': 6, 'k2': 7}.

Q695. Values of d={'k1': 6, 'k2': 7}.

Q696. Items of d={'k1': 6, 'k2': 7}.

Q697. Invert dict {'a':1,'b':2} (values unique).

Q698. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q699. Max key by value in d={'a':3,'b':7,'c':5}.

Q700. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q701. Get value for key 'k2' in d={'k1': 7, 'k2': 8, 'k3': 9} with default 0.

Q702. Update key 'k1' to 99 in d={'k1': 7, 'k2': 8, 'k3': 9}.

Q703. Merge dicts d={'k1': 7, 'k2': 8, 'k3': 9} and e={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13} (Python 3.9+).

Q704. Keys of d={'k1': 7, 'k2': 8, 'k3': 9}.

Q705. Values of d={'k1': 7, 'k2': 8, 'k3': 9}.

Q706. Items of d={'k1': 7, 'k2': 8, 'k3': 9}.

Q707. Invert dict {'a':1,'b':2} (values unique).

Q708. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q709. Max key by value in d={'a':3,'b':7,'c':5}.

Q710. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q711. Get value for key 'k2' in d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11} with default 0.

Q712. Update key 'k1' to 99 in d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11}.

Q713. Merge dicts d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11} and e={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15} (Python 3.9+).

Q714. Keys of d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11}.

Q715. Values of d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11}.

Q716. Items of d={'k1': 8, 'k2': 9, 'k3': 10, 'k4': 11}.

Q717. Invert dict {'a':1,'b':2} (values unique).

Q718. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q719. Max key by value in d={'a':3,'b':7,'c':5}.

Q720. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q721. Get value for key 'k2' in d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13} with default 0.

Q722. Update key 'k1' to 99 in d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13}.

Q723. Merge dicts d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13} and e={'k1': 11, 'k2': 12} (Python 3.9+).

Q724. Keys of d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13}.

Q725. Values of d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13}.

Q726. Items of d={'k1': 9, 'k2': 10, 'k3': 11, 'k4': 12, 'k5': 13}.

Q727. Invert dict {'a':1,'b':2} (values unique).

Q728. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q729. Max key by value in d={'a':3,'b':7,'c':5}.

Q730. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q731. Get value for key 'k2' in d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15} with default 0.

Q732. Update key 'k1' to 99 in d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15}.

Q733. Merge dicts d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15} and e={'k1': 12, 'k2': 13, 'k3': 14} (Python 3.9+).

Q734. Keys of d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15}.

Q735. Values of d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15}.

Q736. Items of d={'k1': 10, 'k2': 11, 'k3': 12, 'k4': 13, 'k5': 14, 'k6': 15}.

Q737. Invert dict {'a':1,'b':2} (values unique).

Q738. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q739. Max key by value in d={'a':3,'b':7,'c':5}.

Q740. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q741. Get value for key 'k2' in d={'k1': 11, 'k2': 12} with default 0.

Q742. Update key 'k1' to 99 in d={'k1': 11, 'k2': 12}.

Q743. Merge dicts d={'k1': 11, 'k2': 12} and e={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16} (Python 3.9+).

Q744. Keys of d={'k1': 11, 'k2': 12}.

Q745. Values of d={'k1': 11, 'k2': 12}.

Q746. Items of d={'k1': 11, 'k2': 12}.

Q747. Invert dict {'a':1,'b':2} (values unique).

Q748. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q749. Max key by value in d={'a':3,'b':7,'c':5}.

Q750. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q751. Get value for key 'k2' in d={'k1': 12, 'k2': 13, 'k3': 14} with default 0.

Q752. Update key 'k1' to 99 in d={'k1': 12, 'k2': 13, 'k3': 14}.

Q753. Merge dicts d={'k1': 12, 'k2': 13, 'k3': 14} and e={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18} (Python 3.9+).

Q754. Keys of d={'k1': 12, 'k2': 13, 'k3': 14}.

Q755. Values of d={'k1': 12, 'k2': 13, 'k3': 14}.

Q756. Items of d={'k1': 12, 'k2': 13, 'k3': 14}.

Q757. Invert dict {'a':1,'b':2} (values unique).

Q758. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q759. Max key by value in d={'a':3,'b':7,'c':5}.

Q760. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q761. Get value for key 'k2' in d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16} with default 0.

Q762. Update key 'k1' to 99 in d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16}.

Q763. Merge dicts d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16} and e={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20} (Python 3.9+).

Q764. Keys of d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16}.

Q765. Values of d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16}.

Q766. Items of d={'k1': 13, 'k2': 14, 'k3': 15, 'k4': 16}.

Q767. Invert dict {'a':1,'b':2} (values unique).

Q768. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q769. Max key by value in d={'a':3,'b':7,'c':5}.

Q770. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q771. Get value for key 'k2' in d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18} with default 0.

Q772. Update key 'k1' to 99 in d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18}.

Q773. Merge dicts d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18} and e={'k1': 16, 'k2': 17} (Python 3.9+).

Q774. Keys of d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18}.

Q775. Values of d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18}.

Q776. Items of d={'k1': 14, 'k2': 15, 'k3': 16, 'k4': 17, 'k5': 18}.

Q777. Invert dict {'a':1,'b':2} (values unique).

Q778. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q779. Max key by value in d={'a':3,'b':7,'c':5}.

Q780. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q781. Get value for key 'k2' in d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20} with default 0.

Q782. Update key 'k1' to 99 in d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20}.

Q783. Merge dicts d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20} and e={'k1': 17, 'k2': 18, 'k3': 19} (Python 3.9+).

Q784. Keys of d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20}.

Q785. Values of d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20}.

Q786. Items of d={'k1': 15, 'k2': 16, 'k3': 17, 'k4': 18, 'k5': 19, 'k6': 20}.

Q787. Invert dict {'a':1,'b':2} (values unique).

Q788. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q789. Max key by value in d={'a':3,'b':7,'c':5}.

Q790. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q791. Get value for key 'k2' in d={'k1': 16, 'k2': 17} with default 0.

Q792. Update key 'k1' to 99 in d={'k1': 16, 'k2': 17}.

Q793. Merge dicts d={'k1': 16, 'k2': 17} and e={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21} (Python 3.9+).

Q794. Keys of d={'k1': 16, 'k2': 17}.

Q795. Values of d={'k1': 16, 'k2': 17}.

Q796. Items of d={'k1': 16, 'k2': 17}.

Q797. Invert dict {'a':1,'b':2} (values unique).

Q798. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q799. Max key by value in d={'a':3,'b':7,'c':5}.

Q800. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q801. Get value for key 'k2' in d={'k1': 17, 'k2': 18, 'k3': 19} with default 0.

Q802. Update key 'k1' to 99 in d={'k1': 17, 'k2': 18, 'k3': 19}.

Q803. Merge dicts d={'k1': 17, 'k2': 18, 'k3': 19} and e={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23} (Python 3.9+).

Q804. Keys of d={'k1': 17, 'k2': 18, 'k3': 19}.

Q805. Values of d={'k1': 17, 'k2': 18, 'k3': 19}.

Q806. Items of d={'k1': 17, 'k2': 18, 'k3': 19}.

Q807. Invert dict {'a':1,'b':2} (values unique).

Q808. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q809. Max key by value in d={'a':3,'b':7,'c':5}.

Q810. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q811. Get value for key 'k2' in d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21} with default 0.

Q812. Update key 'k1' to 99 in d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21}.

Q813. Merge dicts d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21} and e={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25} (Python 3.9+).

Q814. Keys of d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21}.

Q815. Values of d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21}.

Q816. Items of d={'k1': 18, 'k2': 19, 'k3': 20, 'k4': 21}.

Q817. Invert dict {'a':1,'b':2} (values unique).

Q818. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q819. Max key by value in d={'a':3,'b':7,'c':5}.

Q820. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q821. Get value for key 'k2' in d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23} with default 0.

Q822. Update key 'k1' to 99 in d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23}.

Q823. Merge dicts d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23} and e={'k1': 21, 'k2': 22} (Python 3.9+).

Q824. Keys of d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23}.

Q825. Values of d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23}.

Q826. Items of d={'k1': 19, 'k2': 20, 'k3': 21, 'k4': 22, 'k5': 23}.

Q827. Invert dict {'a':1,'b':2} (values unique).

Q828. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q829. Max key by value in d={'a':3,'b':7,'c':5}.

Q830. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q831. Get value for key 'k2' in d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25} with default 0.

Q832. Update key 'k1' to 99 in d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25}.

Q833. Merge dicts d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25} and e={'k1': 22, 'k2': 23, 'k3': 24} (Python 3.9+).

Q834. Keys of d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25}.

Q835. Values of d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25}.

Q836. Items of d={'k1': 20, 'k2': 21, 'k3': 22, 'k4': 23, 'k5': 24, 'k6': 25}.

Q837. Invert dict {'a':1,'b':2} (values unique).

Q838. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q839. Max key by value in d={'a':3,'b':7,'c':5}.

Q840. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Q841. Get value for key 'k2' in d={'k1': 21, 'k2': 22} with default 0.

Q842. Update key 'k1' to 99 in d={'k1': 21, 'k2': 22}.

Q843. Merge dicts d={'k1': 21, 'k2': 22} and e={'k1': 23, 'k2': 24, 'k3': 25, 'k4': 26} (Python 3.9+).

Q844. Keys of d={'k1': 21, 'k2': 22}.

Q845. Values of d={'k1': 21, 'k2': 22}.

Q846. Items of d={'k1': 21, 'k2': 22}.

Q847. Invert dict {'a':1,'b':2} (values unique).

Q848. Count frequency of elements in list [1,2,1,3,2,1] using dict.

Q849. Max key by value in d={'a':3,'b':7,'c':5}.

Q850. Sort dict {'b':2,'a':3,'c':1} by value ascending (list of tuples).

Mixed/Advanced (Combination)

Q851. Map two lists to dict: keys=['beta', 'gamma', 'delta', 'epsilon', 'zeta'], values=[1, 2, 3, 4, 5].

Q852. From sentence 'a a b c b a', build frequency dict.

Q853. Remove duplicates from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] but keep order.

Q854. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q855. Group words ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] by first letter into dict.

Q856. Create set of vowels present in word 'explanation'.

Q857. Find common items between list [1, 2, 3, 4, 5, 6] and tuple (1, 2, 3, 4, 5, 6).

Q858. Build list of dicts from keys ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] with value length of key.

Q859. Sort list of dicts by key 'age'.

Q860. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] without Counter.

Q861. Map two lists to dict: keys=['gamma', 'delta', 'epsilon', 'zeta', 'eta'], values=[1, 2, 3, 4, 5].

Q862. From sentence 'a a b c b a', build frequency dict.

Q863. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] but keep order.

Q864. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q865. Group words ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] by first letter into dict.

Q866. Create set of vowels present in word 'explanation'.

Q867. Find common items between list [1, 2, 3, 4, 5, 6, 7] and tuple (1, 2, 3, 4, 5, 6, 7).

Q868. Build list of dicts from keys ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] with value length of key.

Q869. Sort list of dicts by key 'age'.

Q870. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] without Counter.

Q871. Map two lists to dict: keys=['delta', 'epsilon', 'zeta', 'eta', 'theta'], values=[1, 2, 3, 4, 5].

Q872. From sentence 'a a b c b a', build frequency dict.

Q873. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] but keep order.

Q874. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q875. Group words ['delta', 'epsilon', 'zeta', 'eta', 'theta'] by first letter into dict.

Q876. Create set of vowels present in word 'explanation'.

Q877. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8] and tuple (1, 2, 3, 4, 5, 6, 7, 8).

Q878. Build list of dicts from keys ['delta', 'epsilon', 'zeta', 'eta', 'theta'] with value length of key.

Q879. Sort list of dicts by key 'age'.

Q880. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] without Counter.

Q881. Map two lists to dict: keys=['epsilon', 'zeta', 'eta', 'theta', 'alpha'], values=[1, 2, 3, 4, 5].

Q882. From sentence 'a a b c b a', build frequency dict.

Q883. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] but keep order.

Q884. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q885. Group words ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] by first letter into dict.

Q886. Create set of vowels present in word 'explanation'.

Q887. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9).

Q888. Build list of dicts from keys ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] with value length of key.

Q889. Sort list of dicts by key 'age'.

Q890. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] without Counter.

Q891. Map two lists to dict: keys=['zeta', 'eta', 'theta', 'alpha', 'beta'], values=[1, 2, 3, 4, 5].

Q892. From sentence 'a a b c b a', build frequency dict.

Q893. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] but keep order.

Q894. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q895. Group words ['zeta', 'eta', 'theta', 'alpha', 'beta'] by first letter into dict.

Q896. Create set of vowels present in word 'explanation'.

Q897. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10).

Q898. Build list of dicts from keys ['zeta', 'eta', 'theta', 'alpha', 'beta'] with value length of key.

Q899. Sort list of dicts by key 'age'.

Q900. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] without Counter.

Q901. Map two lists to dict: keys=['eta', 'theta', 'alpha', 'beta', 'gamma'], values=[1, 2, 3, 4, 5].

Q902. From sentence 'a a b c b a', build frequency dict.

Q903. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] but keep order.

Q904. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q905. Group words ['eta', 'theta', 'alpha', 'beta', 'gamma'] by first letter into dict.

Q906. Create set of vowels present in word 'explanation'.

Q907. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11).

Q908. Build list of dicts from keys ['eta', 'theta', 'alpha', 'beta', 'gamma'] with value length of key.

Q909. Sort list of dicts by key 'age'.

Q910. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] without Counter.
Q911. Map two lists to dict: keys=['theta', 'alpha', 'beta', 'gamma', 'delta'], values=[1, 2, 3, 4, 5].
Q912. From sentence 'a a b c b a', build frequency dict.
Q913. Remove duplicates from list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] but keep order.
Q914. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.
Q915. Group words ['theta', 'alpha', 'beta', 'gamma', 'delta'] by first letter into dict.
Q916. Create set of vowels present in word 'explanation'.
Q917. Find common items between list [1, 2, 3, 4, 5] and tuple (1, 2, 3, 4, 5).
Q918. Build list of dicts from keys ['theta', 'alpha', 'beta', 'gamma', 'delta'] with value length of key.
Q919. Sort list of dicts by key 'age'.
Q920. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] without Counter.
Q921. Map two lists to dict: keys=['alpha', 'beta', 'gamma', 'delta', 'epsilon'], values=[1, 2, 3, 4, 5].
Q922. From sentence 'a a b c b a', build frequency dict.
Q923. Remove duplicates from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] but keep order.
Q924. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.
Q925. Group words ['alpha', 'beta', 'gamma', 'delta', 'epsilon'] by first letter into dict.
Q926. Create set of vowels present in word 'explanation'.
Q927. Find common items between list [1, 2, 3, 4, 5, 6] and tuple (1, 2, 3, 4, 5, 6).
Q928. Build list of dicts from keys ['alpha', 'beta', 'gamma', 'delta', 'epsilon'] with value length of key.
Q929. Sort list of dicts by key 'age'.
Q930. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] without Counter.
Q931. Map two lists to dict: keys=['beta', 'gamma', 'delta', 'epsilon', 'zeta'], values=[1, 2, 3, 4, 5].
Q932. From sentence 'a a b c b a', build frequency dict.
Q933. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] but keep order.
Q934. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.
Q935. Group words ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] by first letter into dict.
Q936. Create set of vowels present in word 'explanation'.
Q937. Find common items between list [1, 2, 3, 4, 5, 6, 7] and tuple (1, 2, 3, 4, 5, 6, 7).
Q938. Build list of dicts from keys ['beta', 'gamma', 'delta', 'epsilon', 'zeta'] with value length of key.
Q939. Sort list of dicts by key 'age'.
Q940. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] without Counter.
Q941. Map two lists to dict: keys=['gamma', 'delta', 'epsilon', 'zeta', 'eta'], values=[1, 2, 3, 4, 5].
Q942. From sentence 'a a b c b a', build frequency dict.
Q943. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] but keep order.
Q944. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.
Q945. Group words ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] by first letter into dict.
Q946. Create set of vowels present in word 'explanation'.
Q947. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8] and tuple (1, 2, 3, 4, 5, 6, 7, 8).
Q948. Build list of dicts from keys ['gamma', 'delta', 'epsilon', 'zeta', 'eta'] with value length of key.
Q949. Sort list of dicts by key 'age'.
Q950. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] without Counter.
Q951. Map two lists to dict: keys=['delta', 'epsilon', 'zeta', 'eta', 'theta'], values=[1, 2, 3, 4, 5].
Q952. From sentence 'a a b c b a', build frequency dict.
Q953. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] but keep order.
Q954. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.
Q955. Group words ['delta', 'epsilon', 'zeta', 'eta', 'theta'] by first letter into dict.
Q956. Create set of vowels present in word 'explanation'.

Q957. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9).

Q958. Build list of dicts from keys ['delta', 'epsilon', 'zeta', 'eta', 'theta'] with value length of key.

Q959. Sort list of dicts by key 'age'.

Q960. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] without Counter.

Q961. Map two lists to dict: keys=['epsilon', 'zeta', 'eta', 'theta', 'alpha'], values=[1, 2, 3, 4, 5].

Q962. From sentence 'a a b c b a', build frequency dict.

Q963. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] but keep order.

Q964. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q965. Group words ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] by first letter into dict.

Q966. Create set of vowels present in word 'explanation'.

Q967. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10).

Q968. Build list of dicts from keys ['epsilon', 'zeta', 'eta', 'theta', 'alpha'] with value length of key.

Q969. Sort list of dicts by key 'age'.

Q970. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] without Counter.

Q971. Map two lists to dict: keys=['zeta', 'eta', 'theta', 'alpha', 'beta'], values=[1, 2, 3, 4, 5].

Q972. From sentence 'a a b c b a', build frequency dict.

Q973. Remove duplicates from list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] but keep order.

Q974. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q975. Group words ['zeta', 'eta', 'theta', 'alpha', 'beta'] by first letter into dict.

Q976. Create set of vowels present in word 'explanation'.

Q977. Find common items between list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] and tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11).

Q978. Build list of dicts from keys ['zeta', 'eta', 'theta', 'alpha', 'beta'] with value length of key.

Q979. Sort list of dicts by key 'age'.

Q980. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] without Counter.

Q981. Map two lists to dict: keys=['eta', 'theta', 'alpha', 'beta', 'gamma'], values=[1, 2, 3, 4, 5].

Q982. From sentence 'a a b c b a', build frequency dict.

Q983. Remove duplicates from list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] but keep order.

Q984. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q985. Group words ['eta', 'theta', 'alpha', 'beta', 'gamma'] by first letter into dict.

Q986. Create set of vowels present in word 'explanation'.

Q987. Find common items between list [1, 2, 3, 4, 5] and tuple (1, 2, 3, 4, 5).

Q988. Build list of dicts from keys ['eta', 'theta', 'alpha', 'beta', 'gamma'] with value length of key.

Q989. Sort list of dicts by key 'age'.

Q990. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] without Counter.

Q991. Map two lists to dict: keys=['theta', 'alpha', 'beta', 'gamma', 'delta'], values=[1, 2, 3, 4, 5].

Q992. From sentence 'a a b c b a', build frequency dict.

Q993. Remove duplicates from list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] but keep order.

Q994. Flatten list of tuples [(1,'a'),(2,'b')] into two lists.

Q995. Group words ['theta', 'alpha', 'beta', 'gamma', 'delta'] by first letter into dict.

Q996. Create set of vowels present in word 'explanation'.

Q997. Find common items between list [1, 2, 3, 4, 5, 6] and tuple (1, 2, 3, 4, 5, 6).

Q998. Build list of dicts from keys ['theta', 'alpha', 'beta', 'gamma', 'delta'] with value length of key.

Q999. Sort list of dicts by key 'age'.

Q1000. Find top-2 frequent nums in list [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] without Counter.

Part B — Concise Solutions

Lists

Q1 — Answer:
`a[-1]`

Q2 — Answer:
`a.reverse()`

Q3 — Answer:
`sorted(a)`

Q4 — Answer:
`a.append(1)`

Q5 — Answer:
`a.remove(1)`

Q6 — Answer:
`a[1:4]`

Q7 — Answer:
`sum(a)`

Q8 — Answer:
`list(dict.fromkeys(a))`

Q9 — Answer:
`a + b`

Q10 — Answer:
`[x for x in a if x in b]`

Q11 — Answer:
`a.count(1)`

Q12 — Answer:
`a[2] = 99`

Q13 — Answer:
`[x for x in nums if x % 2 == 0]`

Q14 — Answer:
`[x*x for x in nums]`

Q15 — Answer:
`[y for x in [[1,2],[3,4],[5]] for y in x]`

Q16 — Answer:
`[(x, x*x) for x in nums]`

Q17 — Answer:
`a[-2:] + a[:-2]`

Q18 — Answer:
`sorted(set(nums))[-2]`

Q19 — Answer:
`[x for x in [1,None,2,None,3] if x is not None]`

Q20 — Answer:
`list(zip([1, 2, 3, 4, 5, 6],['beta', 'gamma', 'delta', 'epsilon', 'zeta']))`

Q21 — Answer:
`a[-1]`

Q22 — Answer:
`a.reverse()`

Q23 — Answer:
`sorted(a)`

Q24 — Answer:
```
a.append(2)
```

Q25 — Answer:
```
a.remove(2)
```

Q26 — Answer:
```
a[1:4]
```

Q27 — Answer:
```
sum(a)
```

Q28 — Answer:
```
list(dict.fromkeys(a))
```

Q29 — Answer:
```
a + b
```

Q30 — Answer:
```
[x for x in a if x in b]
```

Q31 — Answer:
```
a.count(2)
```

Q32 — Answer:
```
a[2] = 99
```

Q33 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q34 — Answer:
```
[x*x for x in nums]
```

Q35 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q36 — Answer:
```
[(x, x*x) for x in nums]
```

Q37 — Answer:
```
a[-2:] + a[:-2]
```

Q38 — Answer:
```
sorted(set(nums))[-2]
```

Q39 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q40 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7],['gamma', 'delta', 'epsilon', 'zeta', 'eta']))
```

Q41 — Answer:
```
a[-1]
```

Q42 — Answer:
```
a.reverse()
```

Q43 — Answer:
```
sorted(a)
```

Q44 — Answer:
```
a.append(3)
```

Q45 — Answer:
```
a.remove(3)
```

Q46 — Answer:
```
a[1:4]
```

Q47 — Answer:
```
sum(a)
```

Q48 — Answer:
```
list(dict.fromkeys(a))
```

Q49 — Answer:
```
a + b
```

Q50 — Answer:
```
[x for x in a if x in b]
```

Q51 — Answer:
```
a.count(3)
```

Q52 — Answer:
```
a[2] = 99
```

Q53 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q54 — Answer:
```
[x*x for x in nums]
```

Q55 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q56 — Answer:
```
[(x, x*x) for x in nums]
```

Q57 — Answer:
```
a[-2:] + a[:-2]
```

Q58 — Answer:
```
sorted(set(nums))[-2]
```

Q59 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q60 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8],['delta', 'epsilon', 'zeta', 'eta', 'theta']))
```

Q61 — Answer:
```
a[-1]
```

Q62 — Answer:
```
a.reverse()
```

Q63 — Answer:
```
sorted(a)
```

Q64 — Answer:
```
a.append(4)
```

Q65 — Answer:
```
a.remove(4)
```

Q66 — Answer:
```
a[1:4]
```

Q67 — Answer:
```
sum(a)
```

Q68 — Answer:
```
list(dict.fromkeys(a))
```

Q69 — Answer:
```
a + b
```

Q70 — Answer:
```
[x for x in a if x in b]
```

Q71 — Answer:
```
a.count(4)
```

Q72 — Answer:
```
a[2] = 99
```

Q73 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q74 — Answer:
```
[x*x for x in nums]
```

Q75 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q76 — Answer:
```
[(x, x*x) for x in nums]
```

Q77 — Answer:
```
a[-2:] + a[:-2]
```

Q78 — Answer:
```
sorted(set(nums))[-2]
```

Q79 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q80 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8, 9],['epsilon', 'zeta', 'eta', 'theta', 'alpha']))
```

Q81 — Answer:
```
a[-1]
```

Q82 — Answer:
```
a.reverse()
```

Q83 — Answer:
```
sorted(a)
```

Q84 — Answer:
```
a.append(5)
```

Q85 — Answer:
```
a.remove(0)
```

Q86 — Answer:
```
a[1:4]
```

Q87 — Answer:
```
sum(a)
```

Q88 — Answer:
```
list(dict.fromkeys(a))
```

Q89 — Answer:
```
a + b
```

Q90 — Answer:
```
[x for x in a if x in b]
```

Q91 — Answer:
```
a.count(5)
```

Q92 — Answer:
```
a[2] = 99
```

Q93 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q94 — Answer:
```
[x*x for x in nums]
```

Q95 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q96 — Answer:
```
[(x, x*x) for x in nums]
```

Q97 — Answer:
```
a[-2:] + a[:-2]
```

Q98 — Answer:
```
sorted(set(nums))[-2]
```

Q99 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q100 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8, 9, 10],['zeta', 'eta', 'theta', 'alpha', 'beta']))
```

Q101 — Answer:
```
a[-1]
```

Q102 — Answer:
```
a.reverse()
```

Q103 — Answer:
```
sorted(a)
```

Q104 — Answer:
```
a.append(6)
```

Q105 — Answer:
```
a.remove(1)
```

Q106 — Answer:
```
a[1:4]
```

Q107 — Answer:
```
sum(a)
```

Q108 — Answer:
```
list(dict.fromkeys(a))
```

Q109 — Answer:
```
a + b
```

Q110 — Answer:
```
[x for x in a if x in b]
```

Q111 — Answer:
```
a.count(6)
```

Q112 — Answer:
```
a[2] = 99
```

Q113 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q114 — Answer:
```
[x*x for x in nums]
```

Q115 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q116 — Answer:
```
[(x, x*x) for x in nums]
```

Q117 — Answer:
```
a[-2:] + a[:-2]
```

Q118 — Answer:
```
sorted(set(nums))[-2]
```

Q119 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q120 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],['eta', 'theta', 'alpha', 'beta', 'gamma']))
```

Q121 — Answer:
```
a[-1]
```

Q122 — Answer:
```
a.reverse()
```

Q123 — Answer:
```
sorted(a)
```

Q124 — Answer:
```
a.append(7)
```

Q125 — Answer:
```
a.remove(2)
```

Q126 — Answer:
```
a[1:4]
```

Q127 — Answer:
```
sum(a)
```

Q128 — Answer:
```
list(dict.fromkeys(a))
```

Q129 — Answer:
```
a + b
```

Q130 — Answer:
```
[x for x in a if x in b]
```

Q131 — Answer:
```
a.count(0)
```

Q132 — Answer:
```
a[2] = 99
```

Q133 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q134 — Answer:
```
[x*x for x in nums]
```

Q135 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q136 — Answer:
```
[(x, x*x) for x in nums]
```

Q137 — Answer:
```
a[-2:] + a[:-2]
```

Q138 — Answer:
```
sorted(set(nums))[-2]
```

Q139 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q140 — Answer:
```
list(zip([1, 2, 3, 4, 5],['theta', 'alpha', 'beta', 'gamma', 'delta']))
```

Q141 — Answer:
```
a[-1]
```

Q142 — Answer:
```
a.reverse()
```

Q143 — Answer:
```
sorted(a)
```

Q144 — Answer:
```
a.append(8)
```

Q145 — Answer:
```
a.remove(3)
```

Q146 — Answer:
```
a[1:4]
```

Q147 — Answer:
```
sum(a)
```

Q148 — Answer:
```
list(dict.fromkeys(a))
```

Q149 — Answer:
```
a + b
```

Q150 — Answer:
```
[x for x in a if x in b]
```

Q151 — Answer:
```
a.count(1)
```

Q152 — Answer:
```
a[2] = 99
```

Q153 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q154 — Answer:
```
[x*x for x in nums]
```

Q155 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q156 — Answer:
```
[(x, x*x) for x in nums]
```

Q157 — Answer:
```
a[-2:] + a[:-2]
```

Q158 — Answer:
```
sorted(set(nums))[-2]
```

Q159 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q160 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6],['alpha', 'beta', 'gamma', 'delta', 'epsilon']))
```

Q161 — Answer:
```
a[-1]
```

Q162 — Answer:
```
a.reverse()
```

Q163 — Answer:
```
sorted(a)
```

Q164 — Answer:
```
a.append(9)
```

Q165 — Answer:
```
a.remove(4)
```

Q166 — Answer:
```
a[1:4]
```

Q167 — Answer:
```
sum(a)
```

Q168 — Answer:
```
list(dict.fromkeys(a))
```

Q169 — Answer:
```
a + b
```

Q170 — Answer:
```
[x for x in a if x in b]
```

Q171 — Answer:
```
a.count(2)
```

Q172 — Answer:
```
a[2] = 99
```

Q173 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q174 — Answer:
```
[x*x for x in nums]
```

Q175 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q176 — Answer:
```
[(x, x*x) for x in nums]
```

Q177 — Answer:
```
a[-2:] + a[:-2]
```

Q178 — Answer:
```
sorted(set(nums))[-2]
```

Q179 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q180 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7],['beta', 'gamma', 'delta', 'epsilon', 'zeta']))
```

Q181 — Answer:
```
a[-1]
```

Q182 — Answer:
```
a.reverse()
```

Q183 — Answer:
```
sorted(a)
```

Q184 — Answer:
```
a.append(0)
```

Q185 — Answer:
```
a.remove(0)
```

Q186 — Answer:
```
a[1:4]
```

Q187 — Answer:
```
sum(a)
```

Q188 — Answer:
```
list(dict.fromkeys(a))
```

Q189 — Answer:
```
a + b
```

Q190 — Answer:
```
[x for x in a if x in b]
```

Q191 — Answer:
```
a.count(3)
```

Q192 — Answer:
```
a[2] = 99
```

Q193 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q194 — Answer:
```
[x*x for x in nums]
```

Q195 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q196 — Answer:
```
[(x, x*x) for x in nums]
```

Q197 — Answer:
```
a[-2:] + a[:-2]
```

Q198 — Answer:
```
sorted(set(nums))[-2]
```

Q199 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q200 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8],['gamma', 'delta', 'epsilon', 'zeta', 'eta']))
```

Q201 — Answer:
```
a[-1]
```

Q202 — Answer:
```
a.reverse()
```

Q203 — Answer:
```
sorted(a)
```

Q204 — Answer:
```
a.append(1)
```

Q205 — Answer:
```
a.remove(1)
```

Q206 — Answer:
```
a[1:4]
```

Q207 — Answer:
```
sum(a)
```

Q208 — Answer:
```
list(dict.fromkeys(a))
```

Q209 — Answer:
```
a + b
```

Q210 — Answer:
```
[x for x in a if x in b]
```

Q211 — Answer:
```
a.count(4)
```

Q212 — Answer:
```
a[2] = 99
```

Q213 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q214 — Answer:
```
[x*x for x in nums]
```

Q215 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q216 — Answer:
```
[(x, x*x) for x in nums]
```

Q217 — Answer:
```
a[-2:] + a[:-2]
```

Q218 — Answer:
```
sorted(set(nums))[-2]
```

Q219 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q220 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8, 9],['delta', 'epsilon', 'zeta', 'eta', 'theta']))
```

Q221 — Answer:
```
a[-1]
```

Q222 — Answer:
```
a.reverse()
```

Q223 — Answer:
```
sorted(a)
```

Q224 — Answer:
```
a.append(2)
```

Q225 — Answer:
```
a.remove(2)
```

Q226 — Answer:
```
a[1:4]
```

Q227 — Answer:
```
sum(a)
```

Q228 — Answer:
```
list(dict.fromkeys(a))
```

Q229 — Answer:
```
a + b
```

Q230 — Answer:
```
[x for x in a if x in b]
```

Q231 — Answer:
```
a.count(5)
```

Q232 — Answer:
```
a[2] = 99
```

Q233 — Answer:
```
[x for x in nums if x % 2 == 0]
```

Q234 — Answer:
```
[x*x for x in nums]
```

Q235 — Answer:
```
[y for x in [[1,2],[3,4],[5]] for y in x]
```

Q236 — Answer:
```
[(x, x*x) for x in nums]
```

Q237 — Answer:
```
a[-2:] + a[:-2]
```

Q238 — Answer:
```
sorted(set(nums))[-2]
```

Q239 — Answer:
```
[x for x in [1,None,2,None,3] if x is not None]
```

Q240 — Answer:
```
list(zip([1, 2, 3, 4, 5, 6, 7, 8, 9, 10],['epsilon', 'zeta', 'eta', 'theta', 'alpha']))
```

Q241 — Answer:
```
a[-1]
```

Q242 — Answer:
```
a.reverse()
```

Q243 — Answer:
```
sorted(a)
```

Q244 — Answer:
```
a.append(3)
```

Q245 — Answer:
```
a.remove(3)
```

Q246 — Answer:
```
a[1:4]
```

Q247 — Answer:
```
sum(a)
```

Q248 — Answer:
```
list(dict.fromkeys(a))
```

Q249 — Answer:
`a + b`

Q250 — Answer:
`[x for x in a if x in b]`

Tuples

Q251 — Answer:
`t[2]`

Q252 — Answer:
`t[1:4]`

Q253 — Answer:
`t + u`

Q254 — Answer:
`a,b,c,d = t`

Q255 — Answer:
`t.count(1)`

Q256 — Answer:
`t.index(1)`

Q257 — Answer:
`tuple([1,2,3])`

Q258 — Answer:
`3 in (1,2,3)`

Q259 — Answer:
`x,y = y,x`

Q260 — Answer:
`t[1][1]`

Q261 — Answer:
`t[2]`

Q262 — Answer:
`t[1:4]`

Q263 — Answer:
`t + u`

Q264 — Answer:
`a,b,c,d = t`

Q265 — Answer:
`t.count(2)`

Q266 — Answer:
`t.index(2)`

Q267 — Answer:
`tuple([1,2,3])`

Q268 — Answer:
`3 in (1,2,3)`

Q269 — Answer:
`x,y = y,x`

Q270 — Answer:
`t[1][1]`

Q271 — Answer:
`t[2]`

Q272 — Answer:
`t[1:4]`

Q273 — Answer:
`t + u`

Q274 — Answer:
`a,b,c,d = t`

Q275 — Answer:
`t.count(0)`

Q276 — Answer:
`t.index(3)`

Q277 — Answer:
`tuple([1,2,3])`

Q278 — Answer:
`3 in (1,2,3)`

Q279 — Answer:
`x,y = y,x`

Q280 — Answer:
`t[1][1]`

Q281 — Answer:
`t[2]`

Q282 — Answer:
`t[1:4]`

Q283 — Answer:
`t + u`

Q284 — Answer:
`a,b,c,d = t`

Q285 — Answer:
`t.count(1)`

Q286 — Answer:
`t.index(4)`

Q287 — Answer:
`tuple([1,2,3])`

Q288 — Answer:
`3 in (1,2,3)`

Q289 — Answer:
`x,y = y,x`

Q290 — Answer:
`t[1][1]`

Q291 — Answer:
`t[2]`

Q292 — Answer:
`t[1:4]`

Q293 — Answer:
`t + u`

Q294 — Answer:
`a,b,c,d = t`

Q295 — Answer:
`t.count(2)`

Q296 — Answer:
`t.index(0)`

Q297 — Answer:
`tuple([1,2,3])`

Q298 — Answer:
`3 in (1,2,3)`

Q299 — Answer:
`x,y = y,x`

Q300 — Answer:
`t[1][1]`

Q301 — Answer:
`t[2]`

Q302 — Answer:
`t[1:4]`

Q303 — Answer:
`t + u`

Q304 — Answer:
`a,b,c,d = t`

Q305 — Answer:
`t.count(0)`

Q306 — Answer:
`t.index(1)`

Q307 — Answer:
`tuple([1,2,3])`

Q308 — Answer:
`3 in (1,2,3)`

Q309 — Answer:
`x,y = y,x`

Q310 — Answer:
`t[1][1]`

Q311 — Answer:
`t[2]`

Q312 — Answer:
`t[1:4]`

Q313 — Answer:
`t + u`

Q314 — Answer:
`a,b,c,d = t`

Q315 — Answer:
`t.count(1)`

Q316 — Answer:
`t.index(2)`

Q317 — Answer:
`tuple([1,2,3])`

Q318 — Answer:
`3 in (1,2,3)`

Q319 — Answer:
`x,y = y,x`

Q320 — Answer:
`t[1][1]`

Q321 — Answer:
`t[2]`

Q322 — Answer:
`t[1:4]`

Q323 — Answer:
`t + u`

Q324 — Answer:
`a,b,c,d = t`

Q325 — Answer:
`t.count(2)`

Q326 — Answer:
`t.index(3)`

Q327 — Answer:
`tuple([1,2,3])`

Q328 — Answer:
`3 in (1,2,3)`

Q329 — Answer:
`x,y = y,x`

Q330 — Answer:
`t[1][1]`

Q331 — Answer:
`t[2]`

Q332 — Answer:
`t[1:4]`

Q333 — Answer:
`t + u`

Q334 — Answer:
`a,b,c,d = t`

Q335 — Answer:
`t.count(0)`

Q336 — Answer:
`t.index(4)`

Q337 — Answer:
`tuple([1,2,3])`

Q338 — Answer:
`3 in (1,2,3)`

Q339 — Answer:
`x,y = y,x`

Q340 — Answer:
`t[1][1]`

Q341 — Answer:
`t[2]`

Q342 — Answer:
`t[1:4]`

Q343 — Answer:
`t + u`

Q344 — Answer:
`a,b,c,d = t`

Q345 — Answer:
`t.count(1)`

Q346 — Answer:
`t.index(0)`

Q347 — Answer:
`tuple([1,2,3])`

Q348 — Answer:
`3 in (1,2,3)`

Q349 — Answer:
`x,y = y,x`

Q350 — Answer:
`t[1][1]`

Q351 — Answer:
`t[2]`

Q352 — Answer:
`t[1:4]`

Q353 — Answer:
`t + u`

Q354 — Answer:
`a,b,c,d = t`

Q355 — Answer:
`t.count(2)`

Q356 — Answer:
`t.index(1)`

Q357 — Answer:
`tuple([1,2,3])`

Q358 — Answer:
`3 in (1,2,3)`

Q359 — Answer:
`x,y = y,x`

Q360 — Answer:
`t[1][1]`

Q361 — Answer:
`t[2]`

Q362 — Answer:
`t[1:4]`

Q363 — Answer:
`t + u`

Q364 — Answer:
`a,b,c,d = t`

Q365 — Answer:
`t.count(0)`

Q366 — Answer:
`t.index(2)`

Q367 — Answer:
`tuple([1,2,3])`

Q368 — Answer:
`3 in (1,2,3)`

Q369 — Answer:
`x,y = y,x`

Q370 — Answer:
`t[1][1]`

Q371 — Answer:
`t[2]`

Q372 — Answer:
`t[1:4]`

Q373 — Answer:
`t + u`

Q374 — Answer:
`a,b,c,d = t`

Q375 — Answer:
`t.count(1)`

Q376 — Answer:
`t.index(3)`

Q377 — Answer:
`tuple([1,2,3])`

Q378 — Answer:
`3 in (1,2,3)`

Q379 — Answer:
`x,y = y,x`

Q380 — Answer:
`t[1][1]`

Q381 — Answer:
`t[2]`

Q382 — Answer:
`t[1:4]`

Q383 — Answer:
`t + u`

Q384 — Answer:
`a,b,c,d = t`

Q385 — Answer:
`t.count(2)`

Q386 — Answer:
`t.index(4)`

Q387 — Answer:
`tuple([1,2,3])`

Q388 — Answer:
`3 in (1,2,3)`

Q389 — Answer:
`x,y = y,x`

Q390 — Answer:
`t[1][1]`

Q391 — Answer:
`t[2]`

Q392 — Answer:
`t[1:4]`

Q393 — Answer:
`t + u`

Q394 — Answer:
`a,b,c,d = t`

Q395 — Answer:
`t.count(0)`

Q396 — Answer:
`t.index(0)`

Q397 — Answer:
`tuple([1,2,3])`

Q398 — Answer:
`3 in (1,2,3)`

Q399 — Answer:
`x,y = y,x`

Q400 — Answer:
`t[1][1]`

Q401 — Answer:
`t[2]`

Q402 — Answer:
`t[1:4]`

Q403 — Answer:
`t + u`

Q404 — Answer:
`a,b,c,d = t`

Q405 — Answer:
`t.count(1)`

Q406 — Answer:
`t.index(1)`

Q407 — Answer:
`tuple([1,2,3])`

Q408 — Answer:
`3 in (1,2,3)`

Q409 — Answer:
`x,y = y,x`

Q410 — Answer:
`t[1][1]`

Q411 — Answer:
`t[2]`

Q412 — Answer:
`t[1:4]`

Q413 — Answer:
`t + u`

Q414 — Answer:
`a,b,c,d = t`

Q415 — Answer:
`t.count(2)`

Q416 — Answer:
`t.index(2)`

Q417 — Answer:
`tuple([1,2,3])`

Q418 — Answer:
`3 in (1,2,3)`

Q419 — Answer:
`x,y = y,x`

Q420 — Answer:
`t[1][1]`

Q421 — Answer:
`t[2]`

Q422 — Answer:
`t[1:4]`

Q423 — Answer:
`t + u`

Q424 — Answer:
`a,b,c,d = t`

Q425 — Answer:
`t.count(0)`

Q426 — Answer:
`t.index(3)`

Q427 — Answer:
`tuple([1,2,3])`

Q428 — Answer:
`3 in (1,2,3)`

Q429 — Answer:
`x,y = y,x`

Q430 — Answer:
`t[1][1]`

Q431 — Answer:
`t[2]`

Q432 — Answer:
`t[1:4]`

Q433 — Answer:
`t + u`

Q434 — Answer:
`a,b,c,d = t`

Q435 — Answer:
`t.count(1)`

Q436 — Answer:
`t.index(4)`

Q437 — Answer:
`tuple([1,2,3])`

Q438 — Answer:
`3 in (1,2,3)`

Q439 — Answer:
`x,y = y,x`

Q440 — Answer:
`t[1][1]`

Q441 — Answer:
`t[2]`

Q442 — Answer:
`t[1:4]`

Q443 — Answer:
`t + u`

Q444 — Answer:
`a,b,c,d = t`

Q445 — Answer:
`t.count(2)`

Q446 — Answer:
`t.index(0)`

Q447 — Answer:
`tuple([1,2,3])`

Q448 — Answer:
```
3 in (1,2,3)
```

Q449 — Answer:
```
x,y = y,x
```

Q450 — Answer:
```
t[1][1]
```

Sets

Q451 — Answer:
```
a | b
```

Q452 — Answer:
```
a & b
```

Q453 — Answer:
```
a - b
```

Q454 — Answer:
```
a ^ b
```

Q455 — Answer:
```
a.add(1)
```

Q456 — Answer:
```
a.discard(1)
```

Q457 — Answer:
```
a <= b
```

Q458 — Answer:
```
set([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6])
```

Q459 — Answer:
```
{x*x for x in range(1,6)}
```

Q460 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] if ([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].count(x)>1)]
```

Q461 — Answer:
```
a | b
```

Q462 — Answer:
```
a & b
```

Q463 — Answer:
```
a - b
```

Q464 — Answer:
```
a ^ b
```

Q465 — Answer:
```
a.add(2)
```

Q466 — Answer:
```
a.discard(2)
```

Q467 — Answer:
```
a <= b
```

Q468 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7])
```

Q469 — Answer:
```
{x*x for x in range(1,6)}
```

Q470 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] if ([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].c
```

Q471 — Answer:
```
a | b
```

Q472 — Answer:
```
a & b
```

Q473 — Answer:
```
a - b
```

Q474 — Answer:
```
a ^ b
```

Q475 — Answer:
```
a.add(3)
```

Q476 — Answer:
```
a.discard(3)
```

Q477 — Answer:
```
a <= b
```

Q478 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8])
```

Q479 — Answer:
```
{x*x for x in range(1,6)}
```

Q480 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] if ([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5
```

Q481 — Answer:
```
a | b
```

Q482 — Answer:
```
a & b
```

Q483 — Answer:
```
a - b
```

Q484 — Answer:
```
a ^ b
```

Q485 — Answer:
```
a.add(4)
```

Q486 — Answer:
```
a.discard(4)
```

Q487 — Answer:
```
a <= b
```

Q488 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Q489 — Answer:
```
{x*x for x in range(1,6)}
```

Q490 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] if ([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2
```

Q491 — Answer:
```
a | b
```

Q492 — Answer:
```
a & b
```

Q493 — Answer:
```
a - b
```

Q494 — Answer:
```
a ^ b
```

Q495 — Answer:
```
a.add(5)
```

Q496 — Answer:
```
a.discard(0)
```

Q497 — Answer:
```
a <= b
```

Q498 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

Q499 — Answer:
```
{x*x for x in range(1,6)}
```

Q500 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] if ([1, 2, 3, 4, 5, 6, 7, 8,
```

Q501 — Answer:
```
a | b
```

Q502 — Answer:
```
a & b
```

Q503 — Answer:
```
a - b
```

Q504 — Answer:
```
a ^ b
```

Q505 — Answer:
```
a.add(6)
```

Q506 — Answer:
```
a.discard(1)
```

Q507 — Answer:
```
a <= b
```

Q508 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
```

Q509 — Answer:
```
{x*x for x in range(1,6)}
```

Q510 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] if ([1, 2, 3, 4, 5,
```

Q511 — Answer:
```
a | b
```

Q512 — Answer:
```
a & b
```

Q513 — Answer:
```
a - b
```

Q514 — Answer:
```
a ^ b
```

Q515 — Answer:
```
a.add(7)
```

Q516 — Answer:
```
a.discard(2)
```

Q517 — Answer:
```
a <= b
```

Q518 — Answer:
```
set([1, 2, 3, 4, 5, 1, 2, 3, 4, 5])
```

Q519 — Answer:
```
{x*x for x in range(1,6)}
```

Q520 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] if ([1, 2, 3, 4, 5, 1, 2, 3, 4, 5].count(x)>1)]
```

Q521 — Answer:
```
a | b
```

Q522 — Answer:
```
a & b
```

Q523 — Answer:
```
a - b
```

Q524 — Answer:
```
a ^ b
```

Q525 — Answer:
```
a.add(8)
```

Q526 — Answer:
```
a.discard(3)
```

Q527 — Answer:
```
a <= b
```

Q528 — Answer:
```
set([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6])
```

Q529 — Answer:
```
{x*x for x in range(1,6)}
```

Q530 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] if ([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].count(x)>1)]
```

Q531 — Answer:
```
a | b
```

Q532 — Answer:
```
a & b
```

Q533 — Answer:
```
a - b
```

Q534 — Answer:
```
a ^ b
```

Q535 — Answer:
```
a.add(9)
```

Q536 — Answer:
```
a.discard(4)
```

Q537 — Answer:
```
a <= b
```

Q538 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7])
```

Q539 — Answer:
```
{x*x for x in range(1,6)}
```

Q540 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] if ([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].c
```

Q541 — Answer:
```
a | b
```

Q542 — Answer:
```
a & b
```

Q543 — Answer:
```
a - b
```

Q544 — Answer:
```
a ^ b
```

Q545 — Answer:
```
a.add(0)
```

Q546 — Answer:
```
a.discard(0)
```

Q547 — Answer:
```
a <= b
```

Q548 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8])
```

Q549 — Answer:
```
{x*x for x in range(1,6)}
```

Q550 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] if ([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5
```

Q551 — Answer:
```
a | b
```

Q552 — Answer:
```
a & b
```

Q553 — Answer:
```
a - b
```

Q554 — Answer:
```
a ^ b
```

Q555 — Answer:
```
a.add(1)
```

Q556 — Answer:
```
a.discard(1)
```

Q557 — Answer:
```
a <= b
```

Q558 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Q559 — Answer:
```
{x*x for x in range(1,6)}
```

Q560 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] if ([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2
```

Q561 — Answer:
```
a | b
```

Q562 — Answer:
```
a & b
```

Q563 — Answer:
```
a - b
```

Q564 — Answer:
```
a ^ b
```

Q565 — Answer:
```
a.add(2)
```

Q566 — Answer:
```
a.discard(2)
```

Q567 — Answer:
```
a <= b
```

Q568 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

Q569 — Answer:
```
{x*x for x in range(1,6)}
```

Q570 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] if ([1, 2, 3, 4, 5, 6, 7, 8,
```

Q571 — Answer:
```
a | b
```

Q572 — Answer:
```
a & b
```

Q573 — Answer:
```
a - b
```

Q574 — Answer:
```
a ^ b
```

Q575 — Answer:
```
a.add(3)
```

Q576 — Answer:
```
a.discard(3)
```

Q577 — Answer:
```
a <= b
```

Q578 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
```

Q579 — Answer:
```
{x*x for x in range(1,6)}
```

Q580 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] if ([1, 2, 3, 4, 5,
```

Q581 — Answer:
```
a | b
```

Q582 — Answer:
```
a & b
```

Q583 — Answer:
```
a - b
```

Q584 — Answer:
```
a ^ b
```

Q585 — Answer:
```
a.add(4)
```

Q586 — Answer:
```
a.discard(4)
```

Q587 — Answer:
```
a <= b
```

Q588 — Answer:
```
set([1, 2, 3, 4, 5, 1, 2, 3, 4, 5])
```

Q589 — Answer:
```
{x*x for x in range(1,6)}
```

Q590 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 1, 2, 3, 4, 5] if ([1, 2, 3, 4, 5, 1, 2, 3, 4, 5].count(x)>1)]
```

Q591 — Answer:
```
a | b
```

Q592 — Answer:
```
a & b
```

Q593 — Answer:
```
a - b
```

Q594 — Answer:
```
a ^ b
```

Q595 — Answer:
```
a.add(5)
```

Q596 — Answer:
```
a.discard(0)
```

Q597 — Answer:
```
a <= b
```

Q598 — Answer:
```
set([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6])
```

Q599 — Answer:
```
{x*x for x in range(1,6)}
```

Q600 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] if ([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6].count(x)>1)]
```

Q601 — Answer:
```
a | b
```

Q602 — Answer:
```
a & b
```

Q603 — Answer:
```
a - b
```

Q604 — Answer:
```
a ^ b
```

Q605 — Answer:
```
a.add(6)
```

Q606 — Answer:
```
a.discard(1)
```

Q607 — Answer:
```
a <= b
```

Q608 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7])
```

Q609 — Answer:
```
{x*x for x in range(1,6)}
```

Q610 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7] if ([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7].c
```

Q611 — Answer:
```
a | b
```

Q612 — Answer:
```
a & b
```

Q613 — Answer:
```
a - b
```

Q614 — Answer:
```
a ^ b
```

Q615 — Answer:
```
a.add(7)
```

Q616 — Answer:
```
a.discard(2)
```

Q617 — Answer:
```
a <= b
```

Q618 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8])
```

Q619 — Answer:
```
{x*x for x in range(1,6)}
```

Q620 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] if ([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5
```

Q621 — Answer:
```
a | b
```

Q622 — Answer:
```
a & b
```

Q623 — Answer:
```
a - b
```

Q624 — Answer:
```
a ^ b
```

Q625 — Answer:
```
a.add(8)
```

Q626 — Answer:
```
a.discard(3)
```

Q627 — Answer:
```
a <= b
```

Q628 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Q629 — Answer:
```
{x*x for x in range(1,6)}
```

Q630 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] if ([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2
```

Q631 — Answer:
```
a | b
```

Q632 — Answer:
```
a & b
```

Q633 — Answer:
```
a - b
```

Q634 — Answer:
```
a ^ b
```

Q635 — Answer:
```
a.add(9)
```

Q636 — Answer:
```
a.discard(4)
```

Q637 — Answer:
```
a <= b
```

Q638 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

Q639 — Answer:
```
{x*x for x in range(1,6)}
```

Q640 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] if ([1, 2, 3, 4, 5, 6, 7, 8,
```

Q641 — Answer:
```
a | b
```

Q642 — Answer:
```
a & b
```

Q643 — Answer:
```
a - b
```

Q644 — Answer:
```
a ^ b
```

Q645 — Answer:
```
a.add(0)
```

Q646 — Answer:
```
a.discard(0)
```

Q647 — Answer:
```
a <= b
```

Q648 — Answer:
```
set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
```

Q649 — Answer:
```
{x*x for x in range(1,6)}
```

Q650 — Answer:
```
[x for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] if ([1, 2, 3, 4, 5,
```

Dictionaries

Q651 — Answer:
```
d.get('k2', 0)
```

Q652 — Answer:
```
d['k1'] = 99
```

Q653 — Answer:
```
d | e
```

Q654 — Answer:
```
list(d.keys())
```

Q655 — Answer:
```
list(d.values())
```

Q656 — Answer:
```
list(d.items())
```

Q657 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q658 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q659 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q660 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q661 — Answer:
```
d.get('k2', 0)
```

Q662 — Answer:
```
d['k1'] = 99
```

Q663 — Answer:
```
d | e
```

Q664 — Answer:
```
list(d.keys())
```

Q665 — Answer:
```
list(d.values())
```

Q666 — Answer:
```
list(d.items())
```

Q667 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q668 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q669 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q670 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q671 — Answer:
```
d.get('k2', 0)
```

Q672 — Answer:
```
d['k1'] = 99
```

Q673 — Answer:
```
d | e
```

Q674 — Answer:
```
list(d.keys())
```

Q675 — Answer:
```
list(d.values())
```

Q676 — Answer:
```
list(d.items())
```

Q677 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q678 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q679 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q680 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q681 — Answer:
```
d.get('k2', 0)
```

Q682 — Answer:
```
d['k1'] = 99
```

Q683 — Answer:
```
d | e
```

Q684 — Answer:
```
list(d.keys())
```

Q685 — Answer:
```
list(d.values())
```

Q686 — Answer:
```
list(d.items())
```

Q687 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q688 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q689 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q690 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q691 — Answer:
```
d.get('k2', 0)
```

Q692 — Answer:
```
d['k1'] = 99
```

Q693 — Answer:
```
d | e
```

Q694 — Answer:
```
list(d.keys())
```

Q695 — Answer:
```
list(d.values())
```

Q696 — Answer:
```
list(d.items())
```

Q697 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q698 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q699 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q700 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q701 — Answer:
```
d.get('k2', 0)
```

Q702 — Answer:
```
d['k1'] = 99
```

Q703 — Answer:
```
d | e
```

Q704 — Answer:
```
list(d.keys())
```

Q705 — Answer:
```
list(d.values())
```

Q706 — Answer:
```
list(d.items())
```

Q707 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q708 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q709 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q710 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q711 — Answer:
```
d.get('k2', 0)
```

Q712 — Answer:
```
d['k1'] = 99
```

Q713 — Answer:
```
d | e
```

Q714 — Answer:
```
list(d.keys())
```

Q715 — Answer:
```
list(d.values())
```

Q716 — Answer:
```
list(d.items())
```

Q717 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q718 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q719 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q720 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q721 — Answer:
```
d.get('k2', 0)
```

Q722 — Answer:
```
d['k1'] = 99
```

Q723 — Answer:
```
d | e
```

Q724 — Answer:
```
list(d.keys())
```

Q725 — Answer:
```
list(d.values())
```

Q726 — Answer:
```
list(d.items())
```

Q727 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q728 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q729 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q730 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q731 — Answer:
```
d.get('k2', 0)
```

Q732 — Answer:
```
d['k1'] = 99
```

Q733 — Answer:
```
d | e
```

Q734 — Answer:
```
list(d.keys())
```

Q735 — Answer:
```
list(d.values())
```

Q736 — Answer:
```
list(d.items())
```

Q737 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q738 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q739 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q740 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q741 — Answer:
```
d.get('k2', 0)
```

Q742 — Answer:
```
d['k1'] = 99
```

Q743 — Answer:
```
d | e
```

Q744 — Answer:
```
list(d.keys())
```

Q745 — Answer:
```
list(d.values())
```

Q746 — Answer:
```
list(d.items())
```

Q747 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q748 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q749 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q750 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q751 — Answer:
```
d.get('k2', 0)
```

Q752 — Answer:
```
d['k1'] = 99
```

Q753 — Answer:
```
d | e
```

Q754 — Answer:
```
list(d.keys())
```

Q755 — Answer:
```
list(d.values())
```

Q756 — Answer:
```
list(d.items())
```

Q757 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q758 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q759 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q760 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q761 — Answer:
```
d.get('k2', 0)
```

Q762 — Answer:
```
d['k1'] = 99
```

Q763 — Answer:
```
d | e
```

Q764 — Answer:
```
list(d.keys())
```

Q765 — Answer:
```
list(d.values())
```

Q766 — Answer:
```
list(d.items())
```

Q767 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q768 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q769 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q770 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q771 — Answer:
```
d.get('k2', 0)
```

Q772 — Answer:
```
d['k1'] = 99
```

Q773 — Answer:
```
d | e
```

Q774 — Answer:
```
list(d.keys())
```

Q775 — Answer:
```
list(d.values())
```

Q776 — Answer:
```
list(d.items())
```

Q777 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q778 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q779 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q780 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q781 — Answer:
```
d.get('k2', 0)
```

Q782 — Answer:
```
d['k1'] = 99
```

Q783 — Answer:
```
d | e
```

Q784 — Answer:
```
list(d.keys())
```

Q785 — Answer:
```
list(d.values())
```

Q786 — Answer:
```
list(d.items())
```

Q787 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q788 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q789 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q790 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q791 — Answer:
```
d.get('k2', 0)
```

Q792 — Answer:
```
d['k1'] = 99
```

Q793 — Answer:
```
d | e
```

Q794 — Answer:
```
list(d.keys())
```

Q795 — Answer:
```
list(d.values())
```

Q796 — Answer:
```
list(d.items())
```

Q797 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q798 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q799 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q800 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q801 — Answer:
```
d.get('k2', 0)
```

Q802 — Answer:
```
d['k1'] = 99
```

Q803 — Answer:
```
d | e
```

Q804 — Answer:
```
list(d.keys())
```

Q805 — Answer:
```
list(d.values())
```

Q806 — Answer:
```
list(d.items())
```

Q807 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q808 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q809 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q810 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q811 — Answer:
```
d.get('k2', 0)
```

Q812 — Answer:
```
d['k1'] = 99
```

Q813 — Answer:
```
d | e
```

Q814 — Answer:
```
list(d.keys())
```

Q815 — Answer:
```
list(d.values())
```

Q816 — Answer:
```
list(d.items())
```

Q817 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q818 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q819 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q820 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q821 — Answer:
```
d.get('k2', 0)
```

Q822 — Answer:
```
d['k1'] = 99
```

Q823 — Answer:
```
d | e
```

Q824 — Answer:
```
list(d.keys())
```

Q825 — Answer:
```
list(d.values())
```

Q826 — Answer:
```
list(d.items())
```

Q827 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q828 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q829 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q830 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q831 — Answer:
```
d.get('k2', 0)
```

Q832 — Answer:
```
d['k1'] = 99
```

Q833 — Answer:
```
d | e
```

Q834 — Answer:
```
list(d.keys())
```

Q835 — Answer:
```
list(d.values())
```

Q836 — Answer:
```
list(d.items())
```

Q837 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q838 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q839 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q840 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Q841 — Answer:
```
d.get('k2', 0)
```

Q842 — Answer:
```
d['k1'] = 99
```

Q843 — Answer:
```
d | e
```

Q844 — Answer:
```
list(d.keys())
```

Q845 — Answer:
```
list(d.values())
```

Q846 — Answer:
```
list(d.items())
```

Q847 — Answer:
```
{v:k for k,v in {'a':1,'b':2}.items()}
```

Q848 — Answer:
```
f={};
for x in [1,2,1,3,2,1]: f[x]=f.get(x,0)+1;
f
```

Q849 — Answer:
```
max({'a':3,'b':7,'c':5}, key=lambda k:{'a':3,'b':7,'c':5}[k])
```

Q850 — Answer:
```
sorted({'b':2,'a':3,'c':1}.items(), key=lambda kv: kv[1])
```

Mixed/Advanced (Combination)

Q851 — Answer:
```
dict(zip(['beta', 'gamma', 'delta', 'epsilon', 'zeta'],[1, 2, 3, 4, 5]))
```

Q852 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q853 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]))
```

Q854 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q855 — Answer:
```
g={};
for s in ['beta', 'gamma', 'delta', 'epsilon', 'zeta']: g.setdefault(s[0], []).append(s); g
```

Q856 — Answer:
```
set('explanation') & set('aeiou')
```

Q857 — Answer:
```
list(set([1, 2, 3, 4, 5, 6]) & set((1, 2, 3, 4, 5, 6)))
```

Q858 — Answer:
```
[{'key':k,'len':len(k)} for k in ['beta', 'gamma', 'delta', 'epsilon', 'zeta']]
```

Q859 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q860 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=True)[:2
```

Q861 — Answer:
```
dict(zip(['gamma', 'delta', 'epsilon', 'zeta', 'eta'],[1, 2, 3, 4, 5]))
```

Q862 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q863 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7]))
```

Q864 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q865 — Answer:
```
g={};
for s in ['gamma', 'delta', 'epsilon', 'zeta', 'eta']: g.setdefault(s[0], []).append(s); g
```

Q866 — Answer:
```
set('explanation') & set('aeiou')
```

Q867 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7]) & set((1, 2, 3, 4, 5, 6, 7)))
```

Q868 — Answer:
```
[{'key':k,'len':len(k)} for k in ['gamma', 'delta', 'epsilon', 'zeta', 'eta']]
```

Q869 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q870 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=Tr
```

Q871 — Answer:
```
dict(zip(['delta', 'epsilon', 'zeta', 'eta', 'theta'],[1, 2, 3, 4, 5]))
```

Q872 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q873 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]))
```

Q874 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q875 — Answer:
```
g={};
for s in ['delta', 'epsilon', 'zeta', 'eta', 'theta']: g.setdefault(s[0], []).append(s); g
```

Q876 — Answer:
```
set('explanation') & set('aeiou')
```

Q877 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8]) & set((1, 2, 3, 4, 5, 6, 7, 8)))
```

Q878 — Answer:
```
[{'key':k,'len':len(k)} for k in ['delta', 'epsilon', 'zeta', 'eta', 'theta']]
```

Q879 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q880 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reve
```

Q881 — Answer:
```
dict(zip(['epsilon', 'zeta', 'eta', 'theta', 'alpha'],[1, 2, 3, 4, 5]))
```

Q882 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q883 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

Q884 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q885 — Answer:
```
g={};
for s in ['epsilon', 'zeta', 'eta', 'theta', 'alpha']: g.setdefault(s[0], []).append(s); g
```

Q886 — Answer:
```
set('explanation') & set('aeiou')
```

Q887 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9)))
```

Q888 — Answer:
```
[{'key':k,'len':len(k)} for k in ['epsilon', 'zeta', 'eta', 'theta', 'alpha']]
```

Q889 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q890 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9]: f[x]=f.get(x,0)+1; sorted(f, key=f.get
```

Q891 — Answer:
```
dict(zip(['zeta', 'eta', 'theta', 'alpha', 'beta'],[1, 2, 3, 4, 5]))
```

Q892 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q893 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))
```

Q894 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q895 — Answer:
```
g={};
for s in ['zeta', 'eta', 'theta', 'alpha', 'beta']: g.setdefault(s[0], []).append(s); g
```

Q896 — Answer:
```
set('explanation') & set('aeiou')
```

Q897 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9, 10)))
```

Q898 — Answer:
```
[{'key':k,'len':len(k)} for k in ['zeta', 'eta', 'theta', 'alpha', 'beta']]
```

Q899 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q900 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]: f[x]=f.get(x,0)+1; sorted(f, k
```

Q901 — Answer:
```
dict(zip(['eta', 'theta', 'alpha', 'beta', 'gamma'],[1, 2, 3, 4, 5]))
```

Q902 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q903 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]))
```

Q904 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q905 — Answer:
```
g={};
for s in ['eta', 'theta', 'alpha', 'beta', 'gamma']: g.setdefault(s[0], []).append(s); g
```

Q906 — Answer:
```
set('explanation') & set('aeiou')
```

Q907 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)))
```

Q908 — Answer:
```
[{'key':k,'len':len(k)} for k in ['eta', 'theta', 'alpha', 'beta', 'gamma']]
```

Q909 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q910 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]: f[x]=f.get(x,0)+1; sor
```

Q911 — Answer:
```
dict(zip(['theta', 'alpha', 'beta', 'gamma', 'delta'],[1, 2, 3, 4, 5]))
```

Q912 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q913 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 1, 2, 3, 4, 5]))
```

Q914 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q915 — Answer:
```
g={};
for s in ['theta', 'alpha', 'beta', 'gamma', 'delta']: g.setdefault(s[0], []).append(s); g
```

Q916 — Answer:
```
set('explanation') & set('aeiou')
```

Q917 — Answer:
```
list(set([1, 2, 3, 4, 5]) & set((1, 2, 3, 4, 5)))
```

Q918 — Answer:
```
[{'key':k,'len':len(k)} for k in ['theta', 'alpha', 'beta', 'gamma', 'delta']]
```

Q919 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q920 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=True)[:2]
```

Q921 — Answer:
```
dict(zip(['alpha', 'beta', 'gamma', 'delta', 'epsilon'],[1, 2, 3, 4, 5]))
```

Q922 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q923 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]))
```

Q924 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q925 — Answer:
```
g={};
for s in ['alpha', 'beta', 'gamma', 'delta', 'epsilon']: g.setdefault(s[0], []).append(s); g
```

Q926 — Answer:
```
set('explanation') & set('aeiou')
```

Q927 — Answer:
```
list(set([1, 2, 3, 4, 5, 6]) & set((1, 2, 3, 4, 5, 6)))
```

Q928 — Answer:
```
[{'key':k,'len':len(k)} for k in ['alpha', 'beta', 'gamma', 'delta', 'epsilon']]
```

Q929 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q930 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=True)[:2
```

Q931 — Answer:
```
dict(zip(['beta', 'gamma', 'delta', 'epsilon', 'zeta'],[1, 2, 3, 4, 5]))
```

Q932 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q933 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7]))
```

Q934 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q935 — Answer:
```
g={};
for s in ['beta', 'gamma', 'delta', 'epsilon', 'zeta']: g.setdefault(s[0], []).append(s); g
```

Q936 — Answer:
```
set('explanation') & set('aeiou')
```

Q937 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7]) & set((1, 2, 3, 4, 5, 6, 7)))
```

Q938 — Answer:
```
[{'key':k,'len':len(k)} for k in ['beta', 'gamma', 'delta', 'epsilon', 'zeta']]
```

Q939 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q940 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=Tr
```

Q941 — Answer:
```
dict(zip(['gamma', 'delta', 'epsilon', 'zeta', 'eta'],[1, 2, 3, 4, 5]))
```

Q942 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q943 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]))
```

Q944 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q945 — Answer:
```
g={};
for s in ['gamma', 'delta', 'epsilon', 'zeta', 'eta']: g.setdefault(s[0], []).append(s); g
```

Q946 — Answer:
```
set('explanation') & set('aeiou')
```

Q947 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8]) & set((1, 2, 3, 4, 5, 6, 7, 8)))
```

Q948 — Answer:
```
[{'key':k,'len':len(k)} for k in ['gamma', 'delta', 'epsilon', 'zeta', 'eta']]
```

Q949 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q950 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reve
```

Q951 — Answer:
```
dict(zip(['delta', 'epsilon', 'zeta', 'eta', 'theta'],[1, 2, 3, 4, 5]))
```

Q952 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q953 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

Q954 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q955 — Answer:
```
g={};
for s in ['delta', 'epsilon', 'zeta', 'eta', 'theta']: g.setdefault(s[0], []).append(s); g
```

Q956 — Answer:
```
set('explanation') & set('aeiou')
```

Q957 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9)))
```

Q958 — Answer:
```
[{'key':k,'len':len(k)} for k in ['delta', 'epsilon', 'zeta', 'eta', 'theta']]
```

Q959 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q960 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9]: f[x]=f.get(x,0)+1; sorted(f, key=f.get
```

Q961 — Answer:
```
dict(zip(['epsilon', 'zeta', 'eta', 'theta', 'alpha'],[1, 2, 3, 4, 5]))
```

Q962 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q963 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))
```

Q964 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q965 — Answer:
```
g={};
for s in ['epsilon', 'zeta', 'eta', 'theta', 'alpha']: g.setdefault(s[0], []).append(s); g
```

Q966 — Answer:
```
set('explanation') & set('aeiou')
```

Q967 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9, 10)))
```

Q968 — Answer:
```
[{'key':k,'len':len(k)} for k in ['epsilon', 'zeta', 'eta', 'theta', 'alpha']]
```

Q969 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q970 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]: f[x]=f.get(x,0)+1; sorted(f, k
```

Q971 — Answer:
```
dict(zip(['zeta', 'eta', 'theta', 'alpha', 'beta'],[1, 2, 3, 4, 5]))
```

Q972 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q973 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]))
```

Q974 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q975 — Answer:
```
g={};
for s in ['zeta', 'eta', 'theta', 'alpha', 'beta']: g.setdefault(s[0], []).append(s); g
```

Q976 — Answer:
```
set('explanation') & set('aeiou')
```

Q977 — Answer:
```
list(set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) & set((1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)))
```

Q978 — Answer:
```
[{'key':k,'len':len(k)} for k in ['zeta', 'eta', 'theta', 'alpha', 'beta']]
```

Q979 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q980 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]: f[x]=f.get(x,0)+1; sor
```

Q981 — Answer:
```
dict(zip(['eta', 'theta', 'alpha', 'beta', 'gamma'],[1, 2, 3, 4, 5]))
```

Q982 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q983 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 1, 2, 3, 4, 5]))
```

Q984 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q985 — Answer:
```
g={};
for s in ['eta', 'theta', 'alpha', 'beta', 'gamma']: g.setdefault(s[0], []).append(s); g
```

Q986 — Answer:
```
set('explanation') & set('aeiou')
```

Q987 — Answer:
```
list(set([1, 2, 3, 4, 5]) & set((1, 2, 3, 4, 5)))
```

Q988 — Answer:
```
[{'key':k,'len':len(k)} for k in ['eta', 'theta', 'alpha', 'beta', 'gamma']]
```

Q989 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q990 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=True)[:2]
```

Q991 — Answer:
```
dict(zip(['theta', 'alpha', 'beta', 'gamma', 'delta'],[1, 2, 3, 4, 5]))
```

Q992 — Answer:
```
f={};
for w in 'a a b c b a'.split(): f[w]=f.get(w,0)+1; f
```

Q993 — Answer:
```
list(dict.fromkeys([1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]))
```

Q994 — Answer:
```
l=[(1,'a'),(2,'b')];
xs=[x for x,_ in l]; ys=[y for _,y in l]
```

Q995 — Answer:
```
g={};
for s in ['theta', 'alpha', 'beta', 'gamma', 'delta']: g.setdefault(s[0], []).append(s); g
```

Q996 — Answer:
```
set('explanation') & set('aeiou')
```

Q997 — Answer:
```
list(set([1, 2, 3, 4, 5, 6]) & set((1, 2, 3, 4, 5, 6)))
```

Q998 — Answer:
```
[{'key':k,'len':len(k)} for k in ['theta', 'alpha', 'beta', 'gamma', 'delta']]
```

Q999 — Answer:
```
sorted([{'name':'a','age':2},{'name':'b','age':1}], key=lambda d: d['age'])
```

Q1000 — Answer:
```
f={};
for x in [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]: f[x]=f.get(x,0)+1; sorted(f, key=f.get, reverse=True)[:2
```