test_results

	code	actions	preds
0	result = 0\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult += len(city) + population\nresult = result	filter(), join(), reduce()	filter(), join(), reduce()
1	result = []\nfor item in items:\n\tif item.isActive:\n\t\tresult.append(item.property)\nresult = sum(result)	filter(), map(), sum()	filter(), map(), sum()
2	result = []\nfor a in as:\n\tfor b in bs:\n\t\tif a % 2 == b:\n\t\t\tresult.append(a * 2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
3	count = 0\nfor list in lists:\n\tfor str in list:\n\t\tcount += 1	flatMap(), count()	flatMap(), count()
4	result = 0\nfor val in data:\n\tresult += val * 5	map(), reduce()	map(), reduce()
5	count = 0\nfor num in numbers:\n\tcount += 1	count()	count()
6	result = 0\nfor entity in entities:\n\tresult = func(result, entity)	reduce()	map()
7	result = []\nfor id in ids:\n\tfor data in datas:\n\t\tresult.append((id, data.lower()))\nresult = sum([len(str(item)) for item in result])	map(), join(), sum()	map(), join(), sum()
8	sum_val = 0\nfor item in items:\n\tsum_val += item.value	reduce()	reduce()
9	result = sorted(words)	sortBy()	sortBy()
10	result = 0\nfor list in lists:\n\tfor str in list:\n\t\tresult += len(str)	flatMap(), sum()	flatMap(), sum()
11	result = []\nfor num in numbers:\n\tresult.append(num)\nresult = list(set(result))\nresult.sort()	map(), distinct(), sortBy()	map(), distinct(), sortBy()
12	result = []\nfor text in texts:\n\tfor text1 in texts1:\n\t\tif 'b' in text and 'c' not in text1:\n\t\t\tresult.append(text.upper())\nresult = ''.join(result)	join(), filter(), map(), join()	join(), filter(), map(), join()
13	result = 0\nfor x in xs:\n\tfor y in ys:\n\t\tif x != y:\n\t\t\tresult += x * y\nresult = result	filter(), join(), reduce()	filter(), join(), reduce()
14	result = 0\nfor num in numbers:\n\tif num > 10:\n\t\tresult += num	filter(), sum()	filter()
15	result = []\nfor city in cities:\n\tfor population in populations:\n\t\tresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result])	map(), join(), reduce()	map(), join(), reduce()
16	result = []\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = len(result)	map(), join(), count()	map(), join(), count()
17	product = 1\nfor val in values:\n\tproduct *= val	reduce()	map()
18	result = 0\nfor num in numbers:\n\tresult += num	sum()	sum()
19	result = []\nfor a in as:\n\tfor b in bs:\n\t\tif a % 2 == b:\n\t\t\tresult.append(a * 2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
20	result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.isValid and not obj1.isValid:\n\t\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: a*b, result)	join(), filter(), map(), reduce()	join(), filter(), map(), reduce()
21	result = []\nfor x in xs:\n\tresult.append(x3)	map()	map()
22	result = 0\nfor num in numbers:\n\tresult += num\nresult = [result for _ in range(5)]\nresult = len(result)	reduce(), flatMap(), count()	reduce(), flatMap(), count()
23	result = []\nfor item in items:\n\tfor item1 in items1:\n\t\tif item.isActive and item1.isActive:\n\t\t\tresult.append(item.property)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
24	result = []\nfor list in lists:\n\tfor num in list:\n\t\tresult.append(num)	flatMap()	flatMap()
25	result = 0\nfor list in lists:\n\tfor num in list:\n\t\tresult += num	flatMap(), sum()	flatMap(), sum()
26	result = []\nfor num in numbers:\n\tresult.append(num * 2)\nresult = result[:5]	map(), take()	map(), take()
27	result = []\nfor id in ids:\n\tfor data in datas:\n\t\tresult.append((id, data.lower()))\nresult = reduce(lambda a, b: a+b, [len(str(item)) for item in result])	join(), map(), reduce()	join(), map(), reduce()
28	result = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, val))	join()	join()
29	result = 0\nfor list in lists:\n\tfor str in list:\n\t\tresult += len(str)	flatMap(), sum()	flatMap(), sum()
30	result = []\nfor num in numbers:\n\tif num > 5:\n\t\tresult.append(num)\nresult = len(result)	filter(), count()	filter(), count()
31	result = sorted(strings, key=len)	sortBy()	count()
32	result = []\nfor entity in entities:\n\tfor entity1 in entities1:\n\t\tif entity.isActive and not entity1.isActive:\n\t\t\tresult.append(entity.property)\nresult = len(result)	join(), filter(), map(), count()	join(), filter(), map(), count()
33	result = 0\nfor key in keys:\n\tfor val in values:\n\t\tresult += key * func(val)\nresult = result	join(), map(), reduce()	join(), map(), reduce()
34	result = []\nfor num in numbers:\n\tresult.append(num * 2)\nresult = sorted(result)\nresult = [item for item in result if item in dict1]	map(), sortBy(), join()	map(), sortBy(), join()
35	result = 0\nfor obj in objs:\n\tresult = func(result, obj)	reduce()	map()
36	count = 0\nfor obj in objects:\n\tif obj.isActive:\n\t\tcount += 1	filter(), count()	filter(), count()
37	result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.isValid and not obj1.isValid:\n\t\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: a*b, result)	join(), filter(), map(), reduce()	join(), filter(), map(), reduce()
38	result = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, val))\nresult = len(result)	join(), count()	join(), count()
39	result = []\nfor list in lists:\n\tfor element in list:\n\t\tresult.append(element)\nresult.sort()	flatMap(), sortBy()	flatMap(), sortBy()
40	result = []\nfor z in zs:\n\tfor t in ts:\n\t\tif z < t:\n\t\t\tresult.append(z**2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
41	result = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort()	filter(), sortBy()	filter(), sortBy()
42	result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.isValid and obj1.isValid:\n\t\t\tresult.append(func(obj))\nresult = len(result)	join(), filter(), map(), count()	join(), filter(), map(), count()
43	result = []\nfor char in chars:\n\tresult.append(char.upper())	map()	map()
44	result = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in result]	join(), map(), sum()	map(), join(), sum()
45	result = []\nfor id in ids:\n\tfor data in datas:\n\t\tif id != data:\n\t\t\tresult.append((id, data))\nresult = len(result)	filter(), join(), count()	filter(), join(), count()
46	result = []\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult.append((city, population))\nresult = reduce(lambda a, b: a+b, [len(str(item)) for item in result])	filter(), join(), map(), reduce()	filter(), join(), map(), reduce()
47	result = []\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result])	join(), map(), join()	join(), map(), join()
48	count = 0\nfor val in values:\n\tcount += 1	count()	count()
49	result = 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result	join(), map(), reduce()	join(), map(), reduce()
50	result = []\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result])	filter(), join(), map(), reduce()	filter(), join(), map(), reduce()
51	result = []\nfor list in lists:\n\tfor obj in list:\n\t\tif obj.isActive:\n\t\t\tresult.append(obj.property)	flatMap(), filter()	flatMap(), filter()
52	result = []\nfor item in items:\n\tif item.isActive:\n\t\tresult.append(item.property)\nresult = sum(result)	filter(), map(), sum()	filter(), map(), sum()
53	result = []\nfor obj in objs:\n\tresult.append(transform(obj))	map()	map()
54	result = []\nfor num in numbers1:\n\tresult.append(num)\nfor num in numbers2:\n\tresult.append(num)\nresult = len(result)	union(), count()	union(), count()
55	result = []\nfor num in numbers:\n\tif num % 2 == 0:\n\t\tresult.append(num)\nresult = len(result)	filter(), count()	filter(), count()
56	result = []\nfor list in lists:\n\tfor obj in list:\n\t\tresult.append(obj.property)	flatMap()	flatMap()
57	intermediate = []\nfor person in people:\n\tif person['age'] > 20:\n\t\tintermediate.append(person)\nresult = ''\nfor person in intermediate:\n\tresult += person['name']	filter(), reduce()	join(), map()
58	count = 0\nfor obj in objects:\n\tcount += 1	count()	count()
59	result = []\nfor x in xs:\n\tfor y in ys:\n\t\tresult.append((x, y))	join()	join()
60	result = 0\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult += len(city) + population\nresult = result	filter(), join(), reduce()	filter(), join(), reduce()
61	result = []\nfor key, val in dict1.items():\n\tif key in dict2.keys():\n\t\tresult.append((key, (val, dict2[key])))\nresult.sort()\nresult = result[:5]\nresult = len(result)	join(), sortBy(), take(), count()	map(), join(), sortBy(), take(), count()
62	result = []\nfor entity in entities:\n\tfor entity1 in entities1:\n\t\tif entity.isActive and entity1.isNotActive:\n\t\t\tresult.append(entity.property)\nresult = len(result)	join(), filter(), map(), count()	join(), filter(), map(), count()
63	result = []\nfor y in ys:\n\tif y != 0:\n\t\tresult.append(y4)	map(), filter()	map(), filter()
64	result = []\nfor id in ids:\n\tfor data in datas:\n\t\tresult.append((id, data.lower()))\nresult = sum([len(str(item)) for item in result])	map(), join(), sum()	map(), join(), sum()
65	result = 0\nfor str in strings:\n\tif 'a' in str:\n\t\tresult += len(str)	filter(), sum()	map(), filter()
66	result = []\nfor entity in entities:\n\tresult.append(entity.property)	map()	map()
67	result = []\nfor a in as:\n\tfor b in bs:\n\t\tif a % 2 > b:\n\t\t\tresult.append(a * 2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
68	result = []\nfor element in list:\n\tif element.startswith('a'):\n\t\tresult.append(element)\nresult = list(set(result))	filter(), distinct()	filter(), distinct(), sortBy()
69	result = []\nfor entity in entities:\n\tfor entity1 in entities1:\n\t\tif not entity.isActive and entity1.isNotActive:\n\t\t\tresult.append(entity.property)\nresult = len(result)	join(), filter(), map(), count()	join(), filter(), map(), count()
70	result = 0\nfor num in numbers:\n\tresult += num\nresult = [result]	reduce(), collect()	reduce(), collect()
71	result = []\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age))\nresult = ''.join([str(item) for item in result])	join(), map()	join(), map()
72	accum = 0\nfor val in vals:\n\taccum += val	reduce()	reduce()
73	result = []\nfor name in names:\n\tfor age in ages:\n\t\tif name != age:\n\t\t\tresult.append((name, age))	filter(), join()	filter(), join()
74	result = []\nfor id in ids:\n\tfor data in datas:\n\t\tif id != data:\n\t\t\tresult.append((id, data.lower()))\nresult = sum([len(str(item)) for item in result]	filter(), join(), map(), sum()	filter(), join(), map(), sum()
75	result = []\nfor key in keys:\n\tfor val in values:\n\t\tif key == val:\n\t\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in result]	filter(), join(), map(), sum()	filter(), join(), map(), sum()
76	result = 0\nfor entity in entities:\n\tresult = func(result, entity)	reduce()	map()
77	result = []\nfor num in numbers:\n\tresult.append(num * 5)\nresult.sort(reverse=True)	map(), sortBy()	map(), sortBy()
78	result = []\nfor city in cities:\n\tfor population in populations:\n\t\tresult.append((city, population))\nresult = sum([pop for city, pop in result])	join(), map(), sum()	join(), map(), sum()
79	result = 0\nfor id in ids:\n\tfor data in datas:\n\t\tresult += id + len(data.lower())\nresult = result	join(), map(), reduce()	join(), map(), reduce()

80	result = []\nfor city in cities:\n\tfor population in populations:\n\t\tresult.append((city, population))\nresult = sum([pop for city, pop in result])	join(), map(), sum()	join(), map(), sum()
81	result = []\nfor num in numbers:\n\tif num > 0:\n\t\tresult.append(num)\nresult = result[:5]	filter(), take()	filter(), take()
82	result = []\nfor name in names:\n\tfor age in ages:\n\t\tif name != age:\n\t\t\tresult.append((name, age))\nresult = sum([age for name, age in result]	filter(), join(), map(), sum()	filter(), join(), map(), sum()
83	result = []\nfor num in numbers:\n\tresult.append(num)\nresult.sort(reverse=True)	map(), sortBy()	map(), sortBy()
84	result = []\nfor z in zs:\n\tfor t in ts:\n\t\tif z > 0 and t > 0:\n\t\t\tresult.append(z2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
85	result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif not obj.isValid and obj1.isValid:\n\t\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: ab, result)	join(), filter(), map(), reduce()	join(), filter(), map(), reduce()
86	result = []\nfor z in zs:\n\tfor t in ts:\n\t\tif z > t:\n\t\t\tresult.append(z**2)\nresult = sum(result)	join(), filter(), map(), sum()	join(), filter(), map(), sum()
87	count = 0\nfor num in numbers:\n\tif num > 10:\n\t\tcount += 1	filter(), count()	filter(), count()
88	result = []\nfor obj in objs:\n\tresult.append(transform(obj))	map()	map()
89	result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif not obj.isValid and obj1.isValid:\n\t\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: ab, result)	join(), filter(), map(), reduce()	join(), filter(), map(), reduce()
90	result = []\nfor sublist in list_of_lists:\n\tfor item in sublist:\n\t\tresult.append(item)\nresult = list(set(result))\nresult = [item for item in result if item in dict1]	flatMap(), distinct(), join()	flatMap(), distinct(), join()
91	result = 0\nfor list in lists:\n\tfor num in list:\n\t\tresult += num	flatMap(), sum()	flatMap(), sum()
92	result = 0\nfor id in ids:\n\tfor data in datas:\n\t\tif id != data:\n\t\t\tresult += id + len(data)\nresult = result	filter(), join(), reduce()	filter(), join(), reduce()
93	result = 0\nfor item in iterable:\n\tresult += item.value	sum()	sum()
94	count = 0\nfor str in strings:\n\tcount += 1	count()	count()
95	count = 0\nfor str in strings:\n\tcount += 1	count()	count()
96	result = []\nfor val in data:\n\tif val > 20:\n\t\tresult.append(val * 5)	map(), filter()	map(), filter()
97	result = []\nfor num in numbers:\n\tresult.append(num)\nresult = list(set(result))\nresult.sort()\nresult = sum(result)	map(), distinct(), sortBy(), sum	map(), distinct(), sortBy()
98	result = []\nfor element in list:\n\tresult.append(element)\nresult.sort()\nresult = result[:5]	map(), take()	map(), sortBy()
99	result = []\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))	map(), join()	map(), join()