test_results

| _ | | actions | nyada |
|---|--|--|--|
| _ | | actions | preds |
| _ | = 0\nfor city in cities:\n\tfor population in populations:\n\t\tf city != population:\n\t\t\tresult += len(city) + population\nresult = result | filter(), join(), reduce() | filter(), join(), reduce() |
| , Jouil | = [\nfor item in items:\n\tif item.isActive:\n\t\tresult.append(item.property)\nresult = sum(result) | filter(), map(), sum() | filter(), map(), sum() |
| rooult | | | |
| | = [\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() |
| count | = 0\nfor list in lists:\n\tfor str in list:\n\t\tcount += 1 | flatMap(), count() | flatMap(), count() |
| esult | = 0\nfor val in data:\n\tresult += val * 5 | map(), reduce() | map(), reduce() |
| count | = 0\nfor num in numbers:\n\tcount += 1 | count() | count() |
| | = 0\nfor entity in entities:\n\tresult = func(result, entity) | reduce() | map() |
| | | - | |
| esult | = [\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() |
| um_ | /al = 0\nfor item in items:\n\tsum_val += item.value | reduce() | reduce() |
| esult | = sorted(words) | sortBy() | sortBy() |
| esult | = 0\nfor list in lists:\n\tfor str in list:\n\t\tresult += len(str) | flatMap(), sum() | flatMap(), sum() |
| | | | |
| | = [[\nfor num in numbers:\n\tresult.append(num)\nresult = list(set(result))\nresult.sort() | map(), distinct(), sortBy() | map(), distinct(), sortBy() |
| esult | = []\nfor text in texts:\n\tfor text1 in texts1:\n\t\tif 'b' in text and 'c' not in text1:\n\t\tresult.append(text.upper())\nresult = ''.join(result) | join(), filter(), map(), join() | join(), filter(), map(), join() |
| esult | = 0\nfor x in xs:\n\tfor y in ys:\n\t\tf x != y:\n\t\tresult += x * y\nresult = result | filter(), join(), reduce() | filter(), join(), reduce() |
| esult | = 0\nfor num in numbers:\n\tif num > 10:\n\t\tresult += num | filter(), sum() | filter() |
| esult | = []\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() |
| | | | |
| esuit | = [\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() |
| rodu | ct = 1\nfor val in values:\n\tproduct *= val | reduce() | map() |
| esult | = 0\nfor num in numbers:\n\tresult += num | sum() | sum() |
| esult | = []\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() |
| | | | |
| | = [\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.isValid and not obj1.isValid:\n\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: a*b, result) | join(), filter(), map(), reduce() | |
| esult | = [[\nfor x in xs:\n\tresult.append(x3) | map() | map() |
| esult | = 0\nfor num in numbers:\n\tresult += num\nresult = [result for _ in range(5)]\nresult = len(result) | reduce(), flatMap(), count() | reduce(), flatMap(), count() |
| | = []\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() |
| | | | |
| | = [[\nfor list in lists:\n\tfor num in list:\n\t\tresult.append(num) | flatMap() | flatMap() |
| esult | = 0\nfor list in lists:\n\tfor num in list:\n\t\tresult += num | flatMap(), sum() | flatMap(), sum() |
| esult | = []\nfor num in numbers:\n\tresult.append(num * 2)\nresult = result[:5] | map(), take() | map(), take() |
| esult | = []\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() |
| | | | |
| | = [[\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, val)) | join() | join() |
| esult | = 0\nfor list in lists:\n\tfor str in list:\n\t\tresult += len(str) | flatMap(), sum() | flatMap(), sum() |
| esult | = []\nfor num in numbers:\n\tif num > 5:\n\t\tresult.append(num)\nresult = len(result) | filter(), count() | filter(), count() |
| esult | = sorted(strings, key=len) | sortBy() | count() |
| | = []\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() |
| | | | |
| esult | = 0\nfor key in keys:\n\tfor val in values:\n\t\tresult += key * func(val)\nresult = result | join(), map(), reduce() | join(), map(), reduce() |
| esult | = []\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() |
| esult | = 0\nfor obj in objs:\n\tresult = func(result, obj) | reduce() | map() |
| ount | = 0\nfor obj in objects:\n\tif obj.isActive:\n\t\tcount += 1 | filter(), count() | filter(), count() |
| | | | |
| esult | = [\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.isValid and not obj1.isValid:\n\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: a*b, result) | join(), filter(), map(), reduce() | join(), filter(), map(), reduce() |
| esult | = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, val))\nresult = len(result) | join(), count() | join(), count() |
| esult | = []\nfor list in lists:\n\tfor element in list:\n\t\tresult.append(element)\nresult.sort() | flatMap(), sortBy() | flatMap(), sortBy() |
| | | | |
| result | $= \int \int$ | ioin() filter() map() sum() | ioin() filter() man() sum() |
| | = []\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() |
| | = []\nfor z in zs:\n\tfor t in ts:\n\t\tif z < t:\n\t\tresult.append(z**2)\nresult = sum(result) = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() | join(), filter(), map(), sum() filter(), sortBy() | join(), filter(), map(), sum() filter(), sortBy() |
| result | | | |
| esult esult | = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() | filter(), sortBy() | filter(), sortBy() |
| esult esult esult | = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = []\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) = []\nfor char in chars:\n\tresult.append(char.upper()) | filter(), sortBy() join(), filter(), map(), count() map() | filter(), sortBy() join(), filter(), map(), count() map() |
| esult esult esult | = [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = [\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) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in result] | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() |
| esult esult esult esult | = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = []\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) = []\nfor char in chars:\n\tresult.append(char.upper()) = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() |
| esult esult esult esult esult | = [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = [\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) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in result] | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() |
| esult esult esult esult esult | = []\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = []\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) = []\nfor char in chars:\n\tresult.append(char.upper()) = []\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() |
| esult esult esult esult esult esult esult | = [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = [\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) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) = [\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]) = [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() |
| esult esult esult esult esult esult | [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() [\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) [\nfor char in chars:\n\tresult.append(char.upper()) [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) [\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]) [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) 0\nfor val in values:\n\tcount += 1 | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() |
| esult | = [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = [\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) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) = [\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]) = [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() |
| esult | [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() [\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) [\nfor char in chars:\n\tresult.append(char.upper()) [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) [\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]) [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) 0\nfor val in values:\n\tcount += 1 | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() |
| esult esult esult esult esult esult ount esult esult | = [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() = [\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) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) = [\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]) = [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() |
| esult esult esult ount esult esult | [\nfor num in numbers:\n\tif num > 10:\n\t\tresult.append(num)\nresult.sort() [\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) [\nfor char in chars:\n\tresult.append(char.upper()) [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) [\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult.append((id, population))\nresult = reduce(lambda a, b: a+b, [len(str(item)) for item in 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]) 0\nfor val in values:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult or y in ys:\n\t\tresult = result 0\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(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() |
| esult esult esult esult ount esult esult esult | = [\\nfor num in numbers:\n\tfn num > 10:\n\t\tresult.append(\num\\nresult.sort() = [\\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.is\alid and obj1.is\alid:\n\t\t\tresult.append(\func(obj))\nresult = len(result) = [\\nfor char in chars:\n\tresult.append(char.upper()) = [\\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) = [\\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]) = [\\nfor val in values:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor val in values:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result = 0\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]) = [\\nfor city in cities:\n\tfor obj in list:\n\t\tif obj.isActive:\n\t\t\tresult.append(obj.property) = [\\nfor list in lists:\n\tfor obj in list:\n\t\tresult.append(item.property)\nresult = sum(result) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() |
| esult esult esult ount esult | = [\nfor num in numbers:\n\tfir num > 10:\n\t\tresult.append(num)\nresult.sort() = [\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.is\alid and obj1.is\alid:\n\t\t\tresult.append(func(obj))\nresult = len(result) = [\nfor char in chars:\n\tresult.append(char.uppert)) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val))\nresult = sum([val for key, val in 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) = [\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]) = [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor val in values:\n\tfor y in ys:\n\t\tresult + = x * y2\nresult = result = 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult = x * y2\nresult = 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]) = [\nfor city in cities:\n\tfor obj in list:\n\t\tif obj.isActive:\n\t\tresult.append(obj.property) = [\nfor item in items:\n\tfor obj in list:\n\t\tresult.append(item.property)\nresult = sum(result) = [\nfor obj in objs:\n\tresult.append(transform(obj))) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() filter(), map(), sum() map() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), sum(), sum() map() |
| esult | = [\\nfor num in numbers:\n\tfn num > 10:\n\t\tresult.append(\num\\nresult.sort() = [\\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.is\alid and obj1.is\alid:\n\t\t\tresult.append(\func(obj))\nresult = len(result) = [\\nfor char in chars:\n\tresult.append(char.upper()) = [\\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val)))\nresult = sum([val for key, val in 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) = [\\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]) = [\\nfor val in values:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor val in values:\n\tfor y in ys:\n\t\tresult += x * y2\nresult = result = 0\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]) = [\\nfor city in cities:\n\tfor obj in list:\n\t\tif obj.isActive:\n\t\t\tresult.append(obj.property) = [\\nfor list in lists:\n\tfor obj in list:\n\t\tresult.append(item.property)\nresult = sum(result) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() |
| esult | = [\nfor num in numbers:\n\tfir num > 10:\n\t\tresult.append(num)\nresult.sort() = [\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif obj.is\alid and obj1.is\alid:\n\t\t\tresult.append(func(obj))\nresult = len(result) = [\nfor char in chars:\n\tresult.append(char.uppert)) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append((key, func(val))\nresult = sum([val for key, val in 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) = [\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]) = [\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1))\nresult = ''.join([str(item) for item in result]) = 0\nfor val in values:\n\tfor y in ys:\n\t\tresult + = x * y2\nresult = result = 0\nfor x in xs:\n\tfor y in ys:\n\t\tresult = x * y2\nresult = 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]) = [\nfor city in cities:\n\tfor obj in list:\n\t\tif obj.isActive:\n\t\tresult.append(obj.property) = [\nfor item in items:\n\tfor obj in list:\n\t\tresult.append(item.property)\nresult = sum(result) = [\nfor obj in objs:\n\tresult.append(transform(obj))) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() filter(), map(), sum() map() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), sum(), sum() map() |
| esult | = [\nfor num in numbers:\n\tifn num > 10:\n\t\tresult.append(num)\nresult.sort() = [\nfor obj in objects:\n\tfor obj in objects1:\n\t\tif obj.is\valid and obj1.is\valid:\n\t\t\tresult.append(func(obj))\nresult = len(result) = [\nfor char in chars:\n\tresult.append(char.upper()) = [\nfor key in keys:\n\tfor val in values:\n\t\tresult.append([key, func(val)))\nresult = sum([val for key, val in result] = [\nfor id in ids:\n\tfor val in values:\n\t\tif id != data:\n\t\t\tresult.append([id, data)\n\t\t\tresult.append([id, data))\nresult = len(result) = [\nfor city in cities:\n\tfor population in populations:\n\t\tif city != population:\n\t\t\tresult.append((id, population))\nresult = reduce(lambda a, b: a+b, [len(str(item)) for item in 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]) = 0\nfor val in values:\n\tfor y in ys:\n\t\tresult.append(\notations:\n\t\t\tfor ity! = population:\n\t\t\tresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result]) = [\nfor city in cities:\n\tfor obj in list:\n\tfor obj in objects:\n\tfor objects:\n | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), sum() map() union(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), sum() map() union(), count() |
| esult | = [\nfor num in numbers:\n\tfr num > 10:\n\t\tresult.append(num)\nresult.sort() = [\nfor obj in objects:\n\tfor obj in objects:\n\tfor obj in objects:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in interesult.append(interesult) = [\nfor obji in objis:\n\tfor obji in interesult:\n\tfor obji in interesu | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() map() union(), count() filter(), count() filter(), count() filter(), filter() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), count() |
| esult | [\nfor num in numbers:\n\tifn um > 10:\n\t\tresult.append(num)\nresult.sort() [\nfor obj in objects:\n\tifn obj in objects:\n\tifn obj-is\s\alid and obj1.is\s\alid:\n\tit\tresult.append(func(obj))\nresult = len(result) [\nfor char in chars:\n\tresult.append(char.upper()) [\nfor key in keys:\n\tfor val in values:\n\tit\tif id = data:\n\tit\tresult.append((key, func(val))\nresult = sum([val for key, val in result]) [\nfor id in ids:\n\tifn data in datas:\n\tit\tif id = data:\n\tit\tresult.append((id, data))\nresult = len(result) [\nfor city in cities:\n\tfor population in populations:\n\tit\tresult.append((name, age + 1))\nresult = ".join([str(item) for item in result]) [\nfor name in names:\n\tifn age in ages:\n\tit\tresult.append((name, age + 1))\nresult = ".join([str(item) for item in result]) [\nfor val in values:\n\tifn population in populations:\n\tit\tif city != population:\n\ti\tiftresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result]) [\nfor city in cities:\n\tfor population in populations:\n\tift city != population:\n\tit\tresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result]) [\nfor item in lists:\n\tfor obj in list:\n\tift obj.isActive:\n\ti\tresult.append((obj.property)) [\nfor item in items:\n\tift item.isActive:\n\tit\tresult.append((item.property)\nresult = sum(result) [\nfor num in numbers:\n\tift result.append(num)\nfor num in numbers:\n\tresult.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\titresult.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() join(), map() |
| esult | = [\nfor num in numbers:\n\tfr num > 10:\n\t\tresult.append(num)\nresult.sort() = [\nfor obj in objects:\n\tfor obj in objects:\n\tfor obj in objects:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in objs:\n\tfor obji in in objects:\n\tfor obji in objs:\n\tfor obji in interesult.append(interesult) = [\nfor obji in objis:\n\tfor obji in interesult:\n\tfor obji in interesu | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), sum() map() union(), count() filter(), count() filter(), count() filter(), filter() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), count() |
| esult | [\nfor num in numbers:\n\tifn um > 10:\n\t\tresult.append(num)\nresult.sort() [\nfor obj in objects:\n\tifn obj in objects:\n\tifn obj-is\s\alid and obj1.is\s\alid:\n\tit\tresult.append(func(obj))\nresult = len(result) [\nfor char in chars:\n\tresult.append(char.upper()) [\nfor key in keys:\n\tfor val in values:\n\tit\tif id = data:\n\tit\tresult.append((key, func(val))\nresult = sum([val for key, val in result]) [\nfor id in ids:\n\tifn data in datas:\n\tit\tif id = data:\n\tit\tresult.append((id, data))\nresult = len(result) [\nfor city in cities:\n\tfor population in populations:\n\tit\tresult.append((name, age + 1))\nresult = ".join([str(item) for item in result]) [\nfor name in names:\n\tifn age in ages:\n\tit\tresult.append((name, age + 1))\nresult = ".join([str(item) for item in result]) [\nfor val in values:\n\tifn population in populations:\n\tit\tif city != population:\n\ti\tiftresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result]) [\nfor city in cities:\n\tfor population in populations:\n\tift city != population:\n\tit\tresult.append((city, population * 2))\nresult = reduce(lambda a, b: a+b, [pop for city, pop in result]) [\nfor item in lists:\n\tfor obj in list:\n\tift obj.isActive:\n\ti\tresult.append((obj.property)) [\nfor item in items:\n\tift item.isActive:\n\tit\tresult.append((item.property)\nresult = sum(result) [\nfor num in numbers:\n\tift result.append(num)\nfor num in numbers:\n\tresult.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\titresult.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) [\nfor num in numbers:\n\tift ro obj in list:\n\tift result.append(num)\nresult = len(result) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() join(), map() |
| esult | [\nfor num in numbers:\n\tfr num > 10:\n\tfresult.append(\num\nresult.sort() [\nfor obj in objects:\n\tfor obj 1 in objects:\n\tfresult.append(char.upper()) [\nfor char in chars:\n\tfor val in values:\n\tfresult.append(key, func(val))\nresult = sum(val for key, val in result) [\nfor key in keys:\n\tfor val in values:\n\tfresult.append(key, func(val))\nresult = sum(val for key, val in result) [\nfor id in ids:\n\tfor data in datas:\n\tfresult.append(key, func(val))\nresult = len(result) [\nfor city in cities:\n\tfor population in populations:\n\tfresult.append((id, data)\nresult = ".join([str(item) for item in result]) [\nfor name in names:\n\tfor age in ages:\n\tfresult.append((name, age + 1))\nresult = ".join([str(item) for item in result]) [\nfor val in values:\n\tfor yin ys:\n\tfresult.append((id) + a x * y2\nresult = result [\nfor city in cities:\n\tfor yin ys:\n\tfresult.append(id) + a x * y2\nresult = result [\nfor city in cities:\n\tfor obj in list:\n\tfresult + a x * y2\nresult = result [\nfor item in items:\n\tfor obj in list:\n\tfresult.append(intem.property)\nresult = sum(result) [\nfor item in items:\n\tfresult.append(item.property)\nresult = sum(result) [\nfor item in items:\n\tfresult.append(item.property)\nresult = sum(result) [\nfor num in numbers:\n\tfresult.append(item.property)\nresult = sum(result) [\nfor num in numbers:\n\tfresult.append(item.property)\nresult = len(result) [\nfor num in numbers:\n\tfresult.append(item.property)\nresult = len(result) [\nfor num in numbers:\n\tfresult.append(item.property)\nresult = len(result) [\nfor num in numbers:\n\tfresult.append(inum)\nresult = len(result) [\nfor person in people:\n\tfresult.append(inum)\nresult = len(result) [\nfor person in people:\n\tfresult.append(inum)\nresult = len(result) [\nfor person in people:\n\tfresult.append(inum)\nresult = \nfor person in intermediate:\n\tresult.append(inum)\nresult = \nfor person in intermediate:\n\tresult.append(inum)\nresult = \nfor pe | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), reduce() count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() flatMap() join(), map() count() |
| esult | Epinfor num in numbers:\n\tif num > 10\:\n\thresult.append(num)\nresult.sort() Epinfor obj in objects:\n\tifor obj in in objects:\n\tifor obj in objects:\n\tifor obji in objects:\n\tifor obj in objects:\n\tifor obj in list:\n\tifor obji in objects:\n\tifor obji in objects:\n\tifor obj in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in list:\n\ | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() filter(), join(), reduce() |
| esult | Eynfor num in numbers:\n\tifn num > 10.\n\t\tresult.append(num)\nresult.sort() Eynfor obj in objects:\n\t\for obj in objects:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. inobj. in objects:\n\t\for obj. in objects:\n\t\for obj | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() count() join(), map() map(), join(), sortBy(), take(), |
| esult | Epinfor num in numbers:\n\tif num > 10\:\n\thresult.append(num)\nresult.sort() Epinfor obj in objects:\n\tifor obj in in objects:\n\tifor obj in objects:\n\tifor obji in objects:\n\tifor obj in objects:\n\tifor obj in list:\n\tifor obji in objects:\n\tifor obji in objects:\n\tifor obj in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in objects:\n\tifor obji in list:\n\tifor obji in list:\n\ | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() filter(), join(), reduce() |
| esult | Eynfor num in numbers:\n\tifn num > 10.\n\t\tresult.append(num)\nresult.sort() Eynfor obj in objects:\n\t\for obj in objects:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. is\valid:\n\t\for obj. inobj. in objects:\n\t\for obj. in objects:\n\t\for obj | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() count() join(), map() map(), join(), sortBy(), take(), |
| esult | Enfort num in numbers:/ntif num > 10.\nt\tresult.append(num)\nresult.sort() Enfor obj in objects:/ntifor obj 1 in objects:/ntifor objects: | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), map(), sum() map() union(), count() filter(), count() flatMap() join(), map() count() join() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() |
| esult | Enfort num in numbers.\n\tifn num > 10\n\t\titresult.append(num)\nresult.sortl) Enfort obj in objects.\n\tifn obj1 in objects1\n\titrit obj.is\valid.and obj1.is\valid.and \titresult.append(func(obj))\nresult = len(result) Enfort key in keys:\n\titro val in values\n\titresult.append(key, func(val))\nresult = sum(val for key, val in result) Enfort key in keys:\n\titro val in values\n\titresult.append(key, func(val))\nresult = sum(val for key, val in result) Enfort val in inds:\n\titro data in datas.\n\titrit id != data.\n\titresult.append(id, data)\nresult = len(result) Enfort or population in populations.\n\titresult.append(id, data)\nresult = ".join(striftem) for item in result)\n\titresult.append(id) Enfort or all in values\n\titro age in ages\n\titresult.append(imme, age + 1)\n\nresult = ".join(striftem) for item in result) Enfort is in inters\n\tifr population in populations\n\titrit id != population.\n\titresult.append(id) Enfort is in lists\n\titro population in populations\n\titresult.append(id)-property) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = len(result) Enfort is in lists\n\titro bij in bijs\n\titresult.append(imm.property)\nresult = len(result) Enfort is in lists\n\titro bij in bijs\n\titresult.append(imm.property)\nresult = len(result) Enfort in in numbers\n\titresult.append(imm.property)\nresult = len(result) Enfort in in in numbers\n\titresult.append(imm.property)\nresult = len(result) Enfort in in intersult.append(imm.property)\nresult = len(result) Enfort in inters\n\titresult.append(imm.property)\nresult = len(result) Enfort in inters\n\titresult.append(imm.property)\nresult = len(re | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() count() join(), map() count() join(), filter(), map(), count() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), join(), sum() |
| esult | ### Information numbers/n/tifror obj1 in objects1/n/t/tresult.append(num)/nresult.sort() ### Information numbers/n/tifror obj1 in objects1/n/t/tifrobj.is/valid and obj1.is/valid/n/t/t/tresult.append(func(obj))/nresult = len(result) ### Information of har in chars/n/tresult.append(fice, func/val))/nresult = sum(func for key, val in result) ### Information in idis/n/t/for data in datas/n/t/tresult.append(fice, func/val))/nresult = len(result) ### Information in idis/n/t/for data in datas/n/t/tresult.append(fice, func/val))/nresult = len(result) ### Information in interes/n/t/for age in ages/n/t/tresult.append(fice) into the interesult interesult in names/n/t/for age in ages/n/t/tresult.append(fice) into the interesult interesult into the interesult interesult interesult into the interesult interesult interesult interesunt | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() map(), join(), sum() filter(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), rount() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), join(), sum() map(), filter() map(), filter() map(), filter() map(), filter() |
| esult | Enfort num in numbers.\n\tifn num > 10\n\t\titresult.append(num)\nresult.sortl) Enfort obj in objects.\n\tifn obj1 in objects1\n\titrit obj.is\valid.and obj1.is\valid.and \titresult.append(func(obj))\nresult = len(result) Enfort key in keys:\n\titro val in values\n\titresult.append(key, func(val))\nresult = sum(val for key, val in result) Enfort key in keys:\n\titro val in values\n\titresult.append(key, func(val))\nresult = sum(val for key, val in result) Enfort val in inds:\n\titro data in datas.\n\titrit id != data.\n\titresult.append(id, data)\nresult = len(result) Enfort or population in populations.\n\titresult.append(id, data)\nresult = ".join(striftem) for item in result)\n\titresult.append(id) Enfort or all in values\n\titro age in ages\n\titresult.append(imme, age + 1)\n\nresult = ".join(striftem) for item in result) Enfort is in inters\n\tifr population in populations\n\titrit id != population.\n\titresult.append(id) Enfort is in lists\n\titro population in populations\n\titresult.append(id)-property) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = sum(result) Enfort is in lists\n\titro population in populations\n\titresult.append(imm.property)\nresult = len(result) Enfort is in lists\n\titro bij in bijs\n\titresult.append(imm.property)\nresult = len(result) Enfort is in lists\n\titro bij in bijs\n\titresult.append(imm.property)\nresult = len(result) Enfort in in numbers\n\titresult.append(imm.property)\nresult = len(result) Enfort in in in numbers\n\titresult.append(imm.property)\nresult = len(result) Enfort in in intersult.append(imm.property)\nresult = len(result) Enfort in inters\n\titresult.append(imm.property)\nresult = len(result) Enfort in inters\n\titresult.append(imm.property)\nresult = len(re | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join(), map() count() join(), filter(), map(), count() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), join(), sum() |
| esult | ### Information numbers/n/tifror obj1 in objects1/n/t/tresult.append(num)/nresult.sort() ### Information numbers/n/tifror obj1 in objects1/n/t/tifrobj.is/valid and obj1.is/valid/n/t/t/tresult.append(func(obj))/nresult = len(result) ### Information of har in chars/n/tresult.append(fice, func/val))/nresult = sum(func for key, val in result) ### Information in idis/n/t/for data in datas/n/t/tresult.append(fice, func/val))/nresult = len(result) ### Information in idis/n/t/for data in datas/n/t/tresult.append(fice, func/val))/nresult = len(result) ### Information in interes/n/t/for age in ages/n/t/tresult.append(fice) into the interesult interesult in names/n/t/for age in ages/n/t/tresult.append(fice) into the interesult interesult into the interesult interesult interesult into the interesult interesult interesult interesunt | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() map(), join(), sum() filter(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), map() count() join(), map() count() join() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), join(), sum() map(), filter() map(), filter() map(), filter() map(), filter() |
| esult | = [Infor num in numbers-knitif num > 10\ninthresult.append(num)\nresult.sort() = [Infor obj in objects\ninthresult.append(charuppent) = [Infor char in chars\nintresult.append(charuppent) = [Infor char in chars\nintresult.append(charuppent) = [Infor key in keys\nintfor vai in values\ninthresult.append(key, func\vai)\ninthresult.esum(vai for key, vai in result) = [Infor key in keys\nintfor vai in values\ninthresult.append(key, func\vai)\ninthresult.esum(vai for key, vai in result) = [Infor city in cities\nintfor data in datas\ninthresult.append(key, func\vai)\ninthresult.esum(vai for key, vai in result) = [Infor name in names\nintfor appulation in populations\ninthresult.append((inty, population)\ninthresult) = [Infor name in names\nintfor ape in ages\ninthresult.append((inty, population)\ninthresult) = [Infor name in names\nintfor ape in ages\ninthresult.append((inty, population)\ninthresult) = [Infor name in names\ninthresult\ninthresult.esum\ninthresult.append(inty)\ninthresult.esum\ninthresult\ninthresult.append(inty)\ninthresult.append(inty, population\ninthresult\ni | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), filter() map(), join(), sum() filter(), sum() map() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map() |
| esult | = [[Infor num in numbers:/nitif num > 10.\nititresult.append(num)\nresult.append[tunc(obj])\nresult = len(result) = [[Infor obj in objects:\nititro objects:\niti | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() flatMap() join(), map() count() join(), map() count() join(), filter(), map(), take(), join(), filter(), map(), count() map(), join(), sortBy(), take(), join(), filter() map(), join(), sum() map(), filter() map(), join(), sum() map(), filter() map(), filter(), map(), sum() filter(), distinct(), sortBy() |
| esult | ### Cyrlor num in numbers 'n'tif num > 10.\n'\thresult append(inun')\nresult append(inun')\nresult = len(result) ### Cyrlor char in chars-'n\n'teresult.append(charuppend) ### Cyrlor char in chars-'n\n'teresult.append(charuppend) ### Cyrlor in chars-'n\n'teresult.append(charuppend) ### Cyrlor in in in its shintfor data in datas-'n'\n'tif id != data-'n\n'thresult.append(id, data)\n'nesult = len(result) ### Cyrlor id in ids's\n'tif or data in datas-'n'\n'tif id != data-'n\n'thresult.append(id, data)\n'nesult = len(result) ### Cyrlor id in ids's\n'tif or openiation in populations-'n'\n'thresult.append(id, data)\n'thresult.append(id, y. population)\n'thresult = reduce(ambda a, b: a+b, [len(stri)tem) for lem in result) ### Cyrlor rain in annes-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor rain in annes-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor is in in instal-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor is in in instal-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = encire in result) ### Cyrlor is in in instal-'n'\n'tor oppulation in populations-'n'\n'thresult.append(inthresu | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sum() map(), filter() map(), filter() map() join(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() |
| esult | = [[Infor num in numbers:/nitif num > 10.\nititresult.append(num)\nresult.append[tunc(obj])\nresult = len(result) = [[Infor obj in objects:\nititro objects:\niti | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() flatMap(), filter() filter(), count() filter(), count() flatMap() join(), map() count() join(), map() count() join(), filter(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sum() map(), filter() map(), filter() map(), join(), sum() filter(), distinct(), sortBy() |
| esult | ### Cyrlor num in numbers 'n'tif num > 10.\n'\thresult append(inun')\nresult append(inun')\nresult = len(result) ### Cyrlor char in chars-'n\n'teresult.append(charuppend) ### Cyrlor char in chars-'n\n'teresult.append(charuppend) ### Cyrlor in chars-'n\n'teresult.append(charuppend) ### Cyrlor in in in its shintfor data in datas-'n'\n'tif id != data-'n\n'thresult.append(id, data)\n'nesult = len(result) ### Cyrlor id in ids's\n'tif or data in datas-'n'\n'tif id != data-'n\n'thresult.append(id, data)\n'nesult = len(result) ### Cyrlor id in ids's\n'tif or openiation in populations-'n'\n'thresult.append(id, data)\n'thresult.append(id, y. population)\n'thresult = reduce(ambda a, b: a+b, [len(stri)tem) for lem in result) ### Cyrlor rain in annes-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor rain in annes-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor is in in instal-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = ".join(stri(tem) for item in result) ### Cyrlor is in in instal-'n'\n'tor age in ages-'n'\n'thresult.append(iname, age + 1)\n'nesult = encire in result) ### Cyrlor is in in instal-'n'\n'tor oppulation in populations-'n'\n'thresult.append(inthresu | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), join(), sum() map(), filter() map(), filter() map() join(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() |
| esult | ### Control of the Co | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() join(), filter(), map(), count() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter() map(), join(), sum() map(), filter() map(), filter() map(), filter() map(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() |
| esult | = [Infor num in numbers/nitif num > 10.\nthresult.append[num]\nresult.sort[] = [Infor num in numbers/nitif num > 10.\nthresult.append[num]\nresult | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter() map(), join(), sum() map(), join(), sum() map(), filter() map(), join(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() |
| esult | "Unfor num in numbers/nitif num > 10/nnt/tresult.append/num/nresult.sort() | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter() map(), join(), sum() map(), filter() map(), filter() map(), join(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join() |
| esult | = [Infor num in numbers/nitif num > 10.\nthresult.append[num]\nresult.sort[] = [Infor num in numbers/nitif num > 10.\nthresult.append[num]\nresult | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), join() count() join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter() map(), join(), sum() map(), join(), sum() map(), filter() map(), join(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() |
| esult | "Unfor num in numbers/nitif num > 10/nnt/tresult.append/num/nresult.sort() | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join() |
| esult | Pinfor num in numbers/intif num > 10/in/thresult.append(numi)/nesult soft) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), filter() map(), join(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sum() map(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() |
| esult | Enforce from in minithers/shift num > 10 shift/tresult.append(jum)/inesult.append(jum) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), join(), sum() filter(), sum() map(), join(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sum() map(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() |
| esult | Control on In numbers in rid num > 10 in thickets in which do juil was and obj. Livial and o | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), reduce() count() join(), sortBy(), take(), count() map(), filter(), map(), count() map(), filter(), map(), sum() filter(), sum() map() join(), filter(), map(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() reduce() map(), sortBy() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), join(), filter(), map(), count() map(), filter() map(), join(), sortBy() join(), filter(), map(), count() map(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() map() map(), sortBy() |
| result | Enforce from in minithers/shift num > 10 shift/tresult.append(jum)/inesult.append(jum) | filter(), sortBy() join(), filter(), map(), count() map() join(), map(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), reduce() count() join() filter(), join(), reduce() join(), sortBy(), take(), count() map(), join(), sum() filter(), sum() map(), join(), sum() filter(), distinct() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() filter(), join(), map(), sum() | filter(), sortBy() join(), filter(), map(), count() map() map(), join(), sum() filter(), join(), count() filter(), join(), map(), reduce() join(), map(), reduce() filter(), join(), map(), reduce() filter(), join(), map(), reduce() filter(), map(), sum() map() union(), count() filter(), count() filter(), join(), reduce() map(), join(), sortBy(), take(), orthogonal () join(), filter(), map(), count() map(), filter() map(), join(), sum() map(), filter(), map(), sum() filter(), distinct(), sortBy() join(), filter(), map(), count() reduce(), collect() join(), map() reduce() filter(), join(), map(), sum() filter(), join(), map(), sum() |

| 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() |
| 34 | $result = [\nfor z in zs:\n\tfor t in ts:\n\tfi z > 0 and t > 0:\n\tft esult.append(z2)\nresult = sum(result)] $ | join(), filter(), map(), sum() | join(), filter(), map(), sum() |
| 35 | result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif not obj.isValid and obj1.isValid:\n\t\tresult.append(func(obj))\nresult = reduce(lambda a, b: ab, result) | join(), filter(), map(), reduce() | join(), filter(), map(), reduce() |
| 36 | $result = [\n zs:\n t in ts:\n t in t in ts:\n t in ts:\n t in t in t in ts:\n t in t in ts:\n t in t$ | join(), filter(), map(), sum() | join(), filter(), map(), sum() |
| 37 | 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() |
| 39 | result = []\nfor obj in objects:\n\tfor obj1 in objects1:\n\t\tif not obj.isValid and obj1.isValid:\n\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\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(), sur | map(), distinct(), sortBy() |
| 8 | result = []\nfor element in list:\n\tresult.append(element)\nresult.sort()\nresult = result[:5] | map(), take() | map(), sortBy() |
| 9 | result = []\nfor name in names:\n\tfor age in ages:\n\t\tresult.append((name, age + 1)) | map(), join() | map(), join() |
| - | | | |