

```
# 1
users = [{"name": "Amit", "age": 22}, {"name": "Priya", "age": 29},
{"name": "Raj", "age": 32}, {"name": "Meena", "age": 19}]
older_users = list(map(lambda u: u["name"], filter(lambda x: x["age"]
> 25, users)))
print(older_users)
```

```
['Priya', 'Raj']
```

```
# 2
data = ["hello", " ", "", "python ", " dev ", ""]
cleaned = list(map(lambda x: x.strip(), filter(lambda s: s.strip(),
data)))
print(cleaned)
```

```
['hello', 'python', 'dev']
```

```
# 3
emails = ["test@example.com", "invalid", " user@domain.com", "",
"hello"]
valid = list(map(lambda e: e.strip().upper(), filter(lambda e: "@" in
e and e.strip(), emails)))
print(valid)
```

```
['TEST@EXAMPLE.COM', 'USER@DOMAIN.COM']
```

```
# 4
data_dict = {"a": 10, "b": 5, "c": 15, "d": 3}
filtered = dict(filter(lambda kv: kv[1] > 7, data_dict.items()))
print(filtered)
```

```
{'a': 10, 'c': 15}
```

```
# 5
students = [{"name": "Ravi", "marks": 85}, {"name": "Riya", "marks":
40}, {"name": "Neha", "marks": 76}]
passed = list(map(lambda s: s["name"], filter(lambda s: s["marks"] >=
50, students)))
print(passed)
```

```
['Ravi', 'Neha']
```

```
# 6
nums = list(range(1, 31))
result = list(filter(lambda x: x % 2 == 0 and x % 3 == 0, nums))
print(result)
```

```
[6, 12, 18, 24, 30]
```

```
#7
data = ["Hello", None, "", " ", "World", "Python "]
```

```

cleaned = list(filter(lambda x: x and x.strip(), data))
print(cleaned)

['Hello', 'World', 'Python ']

# 8
users = ["alice", "bob", "john", "admin", "eve"]
blacklist = {"admin", "eve"}
filtered = list(filter(lambda u: u not in blacklist, users))
print(filtered)

['alice', 'bob', 'john']

# 9
nums = [1,2,3,4,5,6,7,8,9,10]
evens = list(filter(lambda x: x % 2 == 0, nums))
print(evens)

[2, 4, 6, 8, 10]

# 10
words = ["apple", "banana", "avocado", "grape", "apricot", "berry"]
starts_a = list(filter(lambda w: w.startswith('a'), words))
print(starts_a)

['apple', 'avocado', 'apricot']

# 11
numbers = [-5, -2, 0, 3, 7, -1, 10]
positive = list(filter(lambda n: n >= 0, numbers))
print(positive)

[0, 3, 7, 10]

# 12
nums = [2,4,6,8,10]
cubes = list(map(lambda x: x**3, filter(lambda x: x > 5, nums)))
print(cubes)

[216, 512, 1000]

# 13
people = [("Amit",17),("Priya",22),("Ravi",30),("Meena",16)]
adults = list(filter(lambda p: p[1] > 18, people))
print(adults)

[('Priya', 22), ('Ravi', 30)]

# 14
words = ["madam","apple","level","hello","noon"]
palindromes = list(filter(lambda w: w == w[::-1], words))
print(palindromes)

```

```

['madam', 'level', 'noon']

# 15
nums = [1,2,4,5,9,10,16,20]
perfect_squares = list(filter(lambda x: int(x**0.5)**2 == x, nums))
print(perfect_squares)

[1, 4, 9, 16]

# 16
words = ["python","", "map", " ", "reduce", ""]
filtered = list(filter(lambda w: w.strip(), words))
print(filtered)

['python', 'map', 'reduce']

# 17
words = ["python", "java", "developer", "ai", "filter", "map"]
long_words = list(filter(lambda w: len(w) > 5, words))
print(long_words)

['python', 'developer', 'filter']

# 18
users = [{"name": "Alice", "status": "active"},
{"name": "Bob", "status": "inactive"},
{"name": "Charlie", "status": "active"}]
active = list(filter(lambda u: u["status"] == "active", users))
print(active)

[{'name': 'Alice', 'status': 'active'}, {'name': 'Charlie', 'status': 'active'}]

# 19
employees = [{"name": "Amit", "salary": 45000},
{"name": "Priya", "salary": 60000}, {"name": "Raj", "salary": 75000}]
rich = list(map(lambda e: e["name"], filter(lambda e: e["salary"] > 50000, employees)))
print(rich)

['Priya', 'Raj']

# 20
numbers = ["9876543210", "9123456789", "9988776655"]
masked = list(map(lambda n: "*" * 6 + n[-4:], numbers))
print(masked)

['*****3210', '*****6789', '*****6655']

# 21
files = ["data.csv", "image.jpeg", "report.pdf", "main.py"]

```

```

extensions = list(map(lambda f: f.split(".")[1], files))
print(extensions)

['csv', 'jpeg', 'pdf', 'py']

# 22
prices_usd = [10, 15.5, 100, 75.25]
prices_inr = list(map(lambda p: round(p*83.20, 2), prices_usd))
print(prices_inr)

[832.0, 1289.6, 8320.0, 6260.8]

# 23
names = ["R@hul", "N!kita", "Ami#t", "P&riya"]
clean_names = list(map(lambda n: ''.join(ch for ch in n if
ch.isalpha()), names))
print(clean_names)

['Rhul', 'Nkita', 'Amit', 'Priya']

# 24
prices = [200, 450, 1000, 150]
gst_prices = list(map(lambda p: p + p*0.05, prices))
print(gst_prices)

[210.0, 472.5, 1050.0, 157.5]

# 25
names = ["Amit Sharma", "Neha Rajput", "Ravi Kumar"]
emails = list(map(lambda n: n.lower().replace(" ", ".") + "@gmail.com",
names))
print(emails)

['amit.sharma@gmail.com', 'neha.rajput@gmail.com',
'ravi.kumar@gmail.com']

# 26
timestamps = ["2024-06-15 14:30:00", "2024-12-01 09:45:10", "2025-01-10
23:59:59"]
dates = list(map(lambda t: t.split()[0], timestamps))
print(dates)

['2024-06-15', '2024-12-01', '2025-01-10']

# 27
tags = ["python", "devlife", "code", "bugfix"]
hashtags = list(map(lambda t: "#" + t, tags))
print(hashtags)

['#python', '#devlife', '#code', '#bugfix']

```

```

# 28
binaries = ["1010", "111", "1001", "1100"]
decimals = list(map(lambda b: int(b, 2), binaries))
print(decimals)

[10, 7, 9, 12]

# 29
names = ["Amit Kumar Singh", "Riya Sharma", "Neha Raj"]
initials = list(map(lambda n: ''.join(x[0] for x in n.split()),
names))
print(initials)

['AKS', 'RS', 'NR']

# 30
nums = [1, 2, 3, 4, 5]
squares = list(map(lambda x: x**2, nums))
print(squares)

[1, 4, 9, 16, 25]

# 31
words = ["python", "map", "filter", "reduce"]
uppercased = list(map(lambda w: w.upper(), words))
print(uppercased)

['PYTHON', 'MAP', 'FILTER', 'REDUCE']

# 32
sentence = "Python makes coding easy"
words = sentence.split()
lengths = list(map(len, words))
print(lengths)

[6, 5, 6, 4]

# 33
nums = [10, 20, 30, 40]
prefixed = list(map(lambda x: f"Num_{x}", nums))
print(prefixed)

['Num_10', 'Num_20', 'Num_30', 'Num_40']

# 34
temps_c = [0, 25, 37, 100]
temps_f = list(map(lambda c: (c*9/5)+32, temps_c))
print(temps_f)

[32.0, 77.0, 98.6, 212.0]

```

```

# 35
words = ["hello world", "python code", "filter map"]
replaced = list(map(lambda s: s.replace(" ", "-"), words))
print(replaced)

['hello-world', 'python-code', 'filter-map']

# 36
nums = [2, 4, 5, 0, 10]
reciprocals = list(map(lambda x: round(1/x, 2), filter(lambda x: x != 0,
nums)))
print(reciprocals)

[0.5, 0.25, 0.2, 0.1]

# 37
sentence = "Python is powerful"
words = sentence.split()
reversed_words = list(map(lambda w: w[::-1], words))
print(reversed_words)

['nohtyP', 'si', 'lufrewop']

# 38
words = ["python", "coding", "rocks"]
capitalized = list(map(lambda w: w.capitalize(), words))
print(capitalized)

['Python', 'Coding', 'Rocks']

# 39
pairs = [(1, "a"), (2, "b"), (3, "c")]
first_elements = list(map(lambda p: p[0], pairs))
print(first_elements)

[1, 2, 3]

# 40
products = [("Book", 200), ("Laptop", 50000), ("Pen", 20)]
discounted = list(map(lambda p: (p[0], round(p[1]*0.9, 2)), products))
print(discounted)

[('Book', 180.0), ('Laptop', 45000.0), ('Pen', 18.0)]

# 41
emails = ["test@example.com", "user@domain.com"]
obfuscated = list(map(lambda e: "*"*(e.find("@")) + e[e.find("@"):],
emails))
print(obfuscated)

['*****@example.com', '*****@domain.com']

```

```

# 42
emails = ["test@example.com", "hello@gmail.com", "world@yahoo.in"]
domains = list(map(lambda e: e.split("@")[1], emails))
print(domains)

['example.com', 'gmail.com', 'yahoo.in']

#43

from functools import reduce

nums = [3, 4, 5]
factorials = list(map(lambda n: reduce(lambda x, y: x * y, range(1, n + 1)), nums))
print(factorials)

[6, 24, 120]

# 44
nums = [1,2,3,4,5]
total_sum = reduce(lambda a,b: a+b, nums)
print(total_sum)

15

# 45
nums = [10,45,32,99,5]
max_num = reduce(lambda a,b: a if a>b else b, nums)
print(max_num)

99

# 46
nums = [2,3,4,5]
product = reduce(lambda a,b: a*b, nums)
print(product)

120

# 47
words = ["Python", "is", "fun"]
sentence = reduce(lambda a,b: a + " " + b, words)
print(sentence)

Python is fun

# 48
words = ["apple", "banana", "watermelon", "grape"]
longest = reduce(lambda a,b: a if len(a)>len(b) else b, words)
print(longest)

watermelon

```

```
# 49
```

```
words = ["Python","map","reduce"]  
total_chars = reduce(lambda a,b: a + len(b), words, 0)  
print(total_chars)
```

```
15
```

```
# 50
```

```
dicts = [{"a":1},{"b":2},{"c":3}]  
merged = reduce(lambda a,b: {**a, **b}, dicts)  
print(merged)
```

```
{'a': 1, 'b': 2, 'c': 3}
```

```
# 51
```

```
transactions = ["+200","-100","+50","-30"]  
balance = reduce(lambda a,b: a + int(b), transactions, 0)  
print(balance)
```

```
120
```