
Answers to Practice Problems — Lesson 2

Solutions

1) Even/Odd

```
def even_or_odd(n):
    if n % 2 == 0:
        print("Even")
    else:
        print("Odd")

# Tests
even_or_odd(7)
even_or_odd(22)
```

Output

```
Odd
Even
```

2) Budget check for prices

```
budget = 20
prices = [8, 12, 5, 15, 4]

for p in prices:
    if p <= budget:
        print(p, ": buy")
    else:
        print(p, ": skip")
```

Output

```
8 : buy
12 : buy
5 : buy
15 : buy
4 : buy
```

3) Transport rules (raining + distance)

```
def mode(raining, distance_miles):
    if raining:
        return "Bus"
    else:
        if distance_miles < 1:
            return "Walk"
        elif distance_miles < 2.5:
            return "Bike"
        else:
            return "Bus"

# Examples
print(mode(True, 0.8))
print(mode(False, 0.6))
print(mode(False, 1.7))
print(mode(False, 3.1))
```

Output

```
Bus
Walk
Bike
Bus
```

4) Numbers from 5 to 15 (inclusive)

```
for i in range(5, 16):
    print(i)
```

Output

```
5
6
7
8
9
10
11
12
13
14
15
```

5) Average temperature

```
temps = [59, 63, 71, 68, 64]
total = 0
count = 0
for t in temps:
    total += t
    count += 1
avg = total / count
print("Average:", avg)
```

Output

```
Average: 65.0
```

6) Mode for each distance

```
miles = [0.6, 1.1, 2.3, 3.2]
for d in miles:
    if d < 1:
        print(d, "-> Walk")
    elif d < 2.5:
        print(d, "-> Bike")
    else:
        print(d, "-> Bus")
```

Output

```
0.6 -> Walk
1.1 -> Bike
2.3 -> Bike
3.2 -> Bus
```

7) Index of fastest route

```
times = [14, 12, 17, 9, 13]
best_time = float('inf')
best_idx = -1

for i, t in enumerate(times):
    if t < best_time:
        best_time = t
        best_idx = i

print("Fastest index:", best_idx, "time:", best_time)
```

Output

```
Fastest index: 3 time: 9
```

8) Index of highest rating (first if tie)

```
ratings = [3, 5, 4, 5, 2]
best_val = -float('inf')
best_idx = -1

for i, r in enumerate(ratings):
    if r > best_val:
        best_val = r
        best_idx = i

print("Best index:", best_idx, "value:", best_val)
```

Output

```
Best index: 1 value: 5
```

9) Most expensive item you can afford

```
prices = [5, 7, 3, 6, 9]
budget = 8
best = -1

for p in prices:
    if p <= budget and p > best:
        best = p

print("Most expensive within budget:", best)
```

Output

```
Most expensive within budget: 7
```

10) (Challenge) Best pair under budget

```
items = [(3,4), (4,5), (6,6), (2,2)] # (price, fun)
budget = 10

best_fun = -1
best_pair = None

for i in range(len(items)):
    for j in range(i+1, len(items)):
```

```
price = items[i][0] + items[j][0]
fun = items[i][1] + items[j][1]
if price <= budget and fun > best_fun:
    best_fun = fun
    best_pair = (items[i], items[j], price, fun)

print("Best pair:", best_pair[0], "+", best_pair[1])
print("Total price:", best_pair[2], "Total fun:", best_pair[3])
```

Output

```
Best pair: (4, 5) + (6, 6)
Total price: 10 Total fun: 11
```