DEADLINE: 14th of July 2025

Question: Extract and Analyze List Slices

Description: Given a list of tuples, where each tuple contains information about a product (product name, price, and quantity), write a program that extracts the product names and prints the names of products with prices above a certain threshold using list slicing and if statements.

The threshold should be set within your code.

```
'''
# Example Input:
products = [
    ("Laptop", 1200, 5),
    ("Smartphone", 700, 10),
    ("Headphones", 150, 15),
    ("Monitor", 300, 7)
]
threshold = 500

# Expected Output:
Products above the price threshold: ['Laptop', 'Smartphone']
'''
```

```
'\n# Example Input:\nproducts = [\n    ("Laptop", 1200, 5),\n    ("Smartphone", 700, 10),\n    ("Headphones", 150, 15),\n
("Monitor", 300, 7)\n]\nthreshold = 500\n\n# Expected Output:\nProducts above the price threshold: [\'Laptop\',
\'Smartphone\']\n'
```

```
products = [
    ("Laptop", 1200, 5),
    ("Smartphone", 700, 10),
    ("Headphones", 150, 15),
    ("Monitor", 300, 7)
]
threshold = 500
valid_products = []
for product in products:
    if product[1] > threshold:
        print(f"the products {product[0]} is above the threshold {threshold} with {product[1] - threshold}")
        valid_products.append(product[0])

print(f"\n\nproducts above the price threshold: {valid_products}")
```

```
the products Laptop is above the threshold 500 with 700
the products Smartphone is above the threshold 500 with 200


products above the price threshold: ['Laptop', 'Smartphone']
```

Question: Update Dictionary with Tuple Data

Description: You have a dictionary where keys are employee IDs and values are their current salaries. You also have a list of tuples where each tuple contains an employee ID and a percentage increment to their salary. Write code to update the salaries in the dictionary based on the increment provided in the list of tuples.

Use if statements to ensure that only existing employees in the dictionary are updated.

```
'''
# Example Input:
salaries = {
    101: 50000,
    102: 60000,
    103: 55000
}
increments = [
    (101, 10),  # 10% increment
    (104, 5),   # 5% increment, not in salaries
    (102, 15)
]

# Expected Output:
Updated salaries: {101: 55000, 102: 69000, 103: 55000}
'''
```

```
salaries = {
    101: 50000,
    102: 60000,
    103: 55000
}
increments = [
    (101, 10),  # 10% increment
    (104, 5),   # 5% increment, not in salaries
    (102, 15)
]

for key in salaries: #but this way is slow as it has a time complecity of O(n^2) so we can do it in a more optimized way:
    for item in increments:
        if key == item[0]:
            salaries[key] = salaries[key] + salaries[key] * (item[1]/100)

print(salaries)
```

```
{101: 55000.0, 102: 69000.0, 103: 55000}
```

but this way is slow as it has a time complexity of O(n^2) so we can do it in a more optimized way:

```
salaries = {
    101: 50000,
    102: 60000,
    103: 55000
}
increments = [
    (101, 10),  # 10% increment
    (104, 5),   # 5% increment, not in salaries
    (102, 15)
]

for id,percent in increments:
    if id in salaries:
        salaries[id] = salaries[id] + salaries[id] * (percent/100)
print(f"Updated salaries: {salaries}")
```

```
Updated salaries: {101: 55000.0, 102: 69000.0, 103: 55000}
```

now the time complexity of this code is optimized and its O(N) only instead of O(n^2) so we did optimize it

```
Start coding or generate with AI.
```