

```

regions = {
    "India": 0.20,
    "China": 0.25,
    "Vietnam": 0.26,
    "EU/USA": 0.10,
    "Other": 0.19
}

purchased_goods = 4_503_000

supplier_data = []
for region, pct in regions.items():
    region_total = purchased_goods * pct
    for i in range(1, 11):
        emissions = region_total * (0.3 / i)
        supplier_data.append({
            "Region": region,
            "Factory": f"{region}_Factory_{i}",
            "Emissions": round(emissions, 2)
        })

print("Available regions:", ", ".join(regions.keys()))
region = input("Enter region: ").strip()
factories = sorted(
    [f for f in supplier_data if f["Region"] == region],
    key=lambda x: x["Emissions"],
    reverse=True
)

if not factories:
    print("Region not found.")
    exit()

total = sum(f["Emissions"] for f in factories)
cumulative = 0
top = []
for f in factories:
    cumulative += f["Emissions"]
    top.append(f)
    if cumulative / total >= 0.80:
        break

print(f"\nTop {len(top)} factories in {region} contributing ~80% emissions:")
for f in top:
    print(f"    Бҗӱ {f['Factory']}    Бҗӱ {f['Emissions']} tCO₂,,e")

factory_name = input("\nSelect one factory from above: ").strip()
factory = next((f for f in top if f["Factory"] == factory_name), None)
if not factory:
    print("Factory not in top list.")
    exit()

print(f"\nRoot Cause for {factory_name}: Coal-based boilers (common in Tier-2 dyeing)")
em = factory["Emissions"]
options = [
    ("Switch to Biomass Boiler", round(em * 0.8)),
    ("Install Solar Thermal", round(em * 1.2)),
    ("Buy Renewable Energy Certificates (RECs)", round(em * 0.6))
]

print("\nCleaner methods & estimated costs:")
for method, cost in options:
    print(f"    {method}: USD {cost:,}")

```