```
# 1. Check user-id and password for login
def login(user_id, password):
    default_id = "Sree_BCA"
    default_pass = "Python@123"
    if user_id == default_id and password == default_pass:
        print("Home-page")
    else:
        print("Invalid-Credentials")
login("Sree_BCA", "Python@123")
```

```
Home-page
```

```
# 2. Create Ticket class with senior citizen discount
class Ticket:
    def __init__(self, **passengers):
        self.passengers = passengers
        self.price = 500

    def display_details(self):
        grand_total = 0
        for name, age in self.passengers.items():
            fare = self.price
            if age >= 60:
                fare = fare * 0.9
            print(f"{name}: {fare}")
            grand_total += fare
        print(f"Total: {grand_total}")

obj = Ticket(person_A=21, person_B=24, person_C=65)
obj.display_details()
```

```
person_A: 500
person_B: 500
person_C: 450.0
Total: 1450.0
```

```
# 3. Ask for password until it is strong
def get_password():
    while True:
        p = input("Enter password: ")
        if any(c.isalpha() for c in p) and any(c.isdigit() for c in p) and any(
            print("Strong password set!")
            break
        print("Weak password!")
```

```
# 4. Calculate the median value of a list
def find_median(lst):
    lst.sort()
    n = len(lst)
    mid = n // 2
    if n % 2 == 0:
        return (lst[mid - 1] + lst[mid]) / 2
```

```
    return lst[mid]
print(find_median([1, 2, 3, 4, 5]))
```

```
3
```

```
# 5. Convert 4-bit binary string to integer
def binary_to_int(binary_str):
    return int(binary_str, 2)
print(binary_to_int("1011"))
```

```
11
```

```
# 6. Use map() to find cubes of list numbers
nums = [1, 2, 3, 4]
cubes = list(map(lambda x: x**3, nums))
print(cubes)
```

```
[1, 8, 27, 64]
```

```
# 7. Add inputs to list until user says 'stop'
def collect():
    elements = []
    while True:
        val = input("Enter: ")
        if val.lower() == 'stop': break
        elements.append(val)
    print(elements)
```

```
# 8. Print only 3-digit numbers using continue
nums = [5, 12, 150, 8, 300, 45, 999]
for x in nums:
    if len(str(x)) != 3:
        continue
    print(x)
```

```
150
300
999
```

```
# 9. Remove strings that start with vowels
def remove_vowels(lst):
    return [s for s in lst if s[0].lower() not in 'aeiou']

print(remove_vowels(["apple", "banana", "orange", "grape"]))
```

```
['banana', 'grape']
```

```
# 10. Transpose a 2D matrix
matrix = [[1, 2], [3, 4], [5, 6]]
transpose = [[matrix[j][i] for j in range(len(matrix))] for i in range(len(mat
print(transpose)
```

```
[[1, 3, 5], [2, 4, 6]]
```