```python
# problem 1

import random
import string


def access_code_function(number_of_codes, length_of_code):
    letter = string.ascii_uppercase
    digit = string.digits
    codes = []

    for x in range(number_of_codes):
        code = (random.choice(letter)) + "".join(random.choice(digit) for y in range(length_of_code))
        codes.append(code)
    return codes


access_code = access_code_function(5, 5)
print(access_code)



# problem 2
start = int(input("Enter starting index: "))
end = int(input("Enter ending index: "))

square = {x: x**2 for x in range(start, end) if x % 2 == 0}
print(square)



# problem 3

def attendance_status():
    n = int(input("Enter the total number of attendees: "))
    attendees = list(map(int, input("Enter the attendee IDs, separated by spaces: ").split()))
    m = int(input("Enter the total number of attendees who attended the conference: "))
    attended = list(
        map(int, input("Enter the IDs of attendees who attended the conference, separated by spaces: ").split()))

    attendance = {}

    for attendee in attendees:
        if attendee in attended:
            attendance[attendee] = 'Attended'
        else:
            attendance[attendee] = 'Did not attend'
    total_attendance = {'Attended': 0, 'Did not attend': 0}

    for status in attendance.values():
        total_attendance[status] += 1

    print(attendance)

    print(total_attendance)


attendance_status()



# problem 4

lst = ["1", "2", "3", "4", "5"]

new_list = [(x, int(x)**2) for x in lst]
print(new_list)



# problem 5

numbers = list(range(101))


def is_palindrome(n):
    return str(n) == str(n)[::-1]


palindromes = list(filter(is_palindrome, numbers))

print(palindromes)

```

```python


# problem 6

import random

random_nums = [random.randint(0,100) for x in range(50)]

unique_nums = list(set(random_nums))

duplicates = []
for num in random_nums:
    print(num,end=" ")
    if num in unique_nums:
        unique_nums.remove(num)
    else:
        duplicates.append(num)

print("\nDuplicate numbers are:", duplicates)




# problem 7

heights = {'Child': [30, 40, 35, 45, 30], 'Teenage': [50, 60, 55, 65, 60], 'Adult': [85, 90, 92, 88, 82]}

average_heights = {age_group: sum(heights) / len(heights) for age_group, heights in heights.items()}

print(average_heights)




# problem 8

celsius_temp = [0, 10, 20, 30, 40, 50]


def convert_fahrenheit_temp(temp):
    fahrenheit_temp = (temp * (9 / 5)) + 32
    return fahrenheit_temp


temp_in_fahrenheit = list(map(convert_fahrenheit_temp, celsius_temp))
print(temp_in_fahrenheit)
```