Question 1

Given an integer array nums of 2n integers, group these integers into n pairs (a1, b1), (a2, b2),..., (an, bn) such that the sum of min(ai, bi) for all i is maximized. Return the maximized sum.

```
def arrayPairSum(nums):
    nums.sort()  # Sort the array in ascending order
    max_sum = 0
    for i in range(0, len(nums), 2):
        max_sum += min(nums[i], nums[i+1])
    return max_sum
```

Question 2

Alice has n candies, where the ith candy is of type candyType[i]. Alice noticed that she started to gain weight, so she visited a doctor. The doctor advised Alice to only eat n / 2 of the candies she has (n is always even). Alice likes her candies very much, and she wants to eat the maximum number of different types of candies while still following the doctor's advice. Given the integer array candyType of length n, return the maximum number of different types of candies she can eat if she only eats n / 2 of them.

```
def maxCandies(candyType):
    unique_candies = set()
    for candy in candyType:
        unique_candies.add(candy)
        if len(unique_candies) == len(candyType) // 2:
            break
    return min(len(unique_candies), len(candyType) // 2)
```

Question 3

We define a harmonious array as an array where the difference between its maximum value and its minimum value is exactly 1. Given an integer array nums, return the length of its longest harmonious subsequence among all its possible subsequences. A subsequence of an array is a sequence that can be derived from the array by deleting some or no elements without changing the order of the remaining elements.

```
from collections import Counter
```

```
def findLHS(nums):
    frequency_map = Counter(nums)
    max_length = 0
    for num in set(nums):
        if num + 1 in frequency_map:
            max_length = max(max_length, frequency_map[num] + frequency_map[num + 1])
    return max_length
```

Question 4

You have a long flowerbed in which some of the plots are planted, and some are not. However, flowers cannot be planted in adjacent plots. Given an integer array flowerbed containing 0's and 1's, where 0 means empty and 1 means not empty, and an integer n, return true if n new flowers can be planted in the flowerbed without violating the no-adjacent-flowers rule and false otherwise.

```
def canPlaceFlowers(flowerbed, n):
    count = 0
    length = len(flowerbed)
    for i in range(length):
        if flowerbed[i] == 0 and (i == 0 or flowerbed[i-1] == 0) and (i == length-1 or flowerbed[i+1] == 0):
            count += 1
            flowerbed[i] = 1
            if count == n:
                return True
    return False
```

Question 5

Given an integer array nums, find three numbers whose product is maximum and return the maximum product.

```
def maximumProduct(nums):
    nums.sort()  # Sort the array in ascending order
    max_product1 = nums[-1] * nums[-2] * nums[-3]
    max_product2 = nums[0] * nums[1] * nums[-1]
    return max(max_product1, max_product2)
```

Question 6

Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1. You must write an algorithm with O(log n) runtime complexity.

```
def search(nums, target):
    left, right = 0, len(nums) - 1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            return mid
        elif nums[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

Question 7

An array is monotonic if it is either monotone increasing or monotone decreasing. An array nums is monotone increasing if for all i <= j, nums[i] <= nums[j]. An array nums is monotone decreasing if for all i <= j, nums[i] >= nums[j]. Given an integer array nums, return true if the given array is monotonic, or false otherwise.

```
def isMonotonic(nums):
    isIncreasing = True
    isDecreasing = True
    for i in range(1, len(nums)):
        if nums[i] > nums[i - 1]:
            isDecreasing = False
        elif nums[i] < nums[i - 1]:
            isIncreasing = False
        if not isIncreasing and not isDecreasing:
            return False
    return True
```

Question 8

You are given an integer array nums and an integer k. In one operation, you can choose any index i where 0 <= i < nums.length and change nums[i] to nums[i] + x where x is an integer from the range [-k, k]. You can apply this operation at most once for each index i. The score of nums is the difference between the maximum and minimum elements in nums. Return the minimum score of nums after applying the mentioned operation at most once for each index in it.

```
def minimumScore(nums, k):
    min_val = float('inf')
    max_val = float('-inf')
    for num in nums:
        min_val = min(min_val, num)
        max_val = max(max_val, num)
    if min_val >= max_val - 2 * k:
        return max_val - k
    else:
        return min_val + k
```