Assignment 4

Question 1 Given three integer arrays arr1, arr2 and arr3 **sorted** in **strictly increasing** order, return a sorted array of **only** the integers that appeared in **all** three arrays.

Example 1:

Input: arr1 = [1,2,3,4,5], arr2 = [1,2,5,7,9], arr3 = [1,3,4,5,8]

Output: [1,5]

Explanation: Only 1 and 5 appeared in the three arrays.

Ans =

```
class Solution(object):


    def makeArrayIncreasing(self, arr1, arr2):


        arr2 = sorted(arr2)

        dict_pre = {0: -float("inf")}


        for num in arr1:

            dict_cur =
collections.defaultdict(lambda: float("inf"))

            for n_swap in dict_pre:

                if num > dict_pre[n_swap]:

                    dict_cur[n_swap] =
min(dict_cur[n_swap], num)
```

```
                loc = bisect.bisect(arr2,
dict_pre[n_swap])
                if loc < len(arr2):
                    dict_cur[n_swap+1] =
min(dict_cur[n_swap+1], arr2[loc])


            if not dict_cur:
                return -1
            dict_pre = dict_cur


        return min(dict_pre.keys())
```

Question 3 Given a 2D integer array matrix, return *the* ***transpose*** *of* matrix.

The **transpose** of a matrix is the matrix flipped over its main diagonal, switching the matrix's row and column indices.

Example 1:

Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]

Output: [[1,4,7],[2,5,8],[3,6,9]]

Ans =

```
class Solution {
public:
```

```
    vector<vector<int>> transpose(vector<vector<int>>& matrix) {
        vector<vector<int>> v(matrix[0].size(), vector<int>(matrix.size()));
        for(int i=0;i<matrix.size();i++)
        {
            for(int j=0;j<matrix[0].size();j++)
            {
                v[j][i]=matrix[i][j];
            }
        }
        return v;
    }
};
```

Question 5

You have n coins and you want to build a staircase with these coins. The staircase consists of k rows where the ith row has exactly i coins. The last row of the staircase may be incomplete.

Given the integer n, return *the number of complete rows of the staircase you will build*.

Ans =

```
class Solution {
public:
```

```
    int arrangeCoins(int n)
    {
        return (-1+sqrt(1.0+4.0*2.0*n))/2;
    }
};
```

Question 6

Given an integer array nums sorted in non-decreasing order, return *an array of the squares of each number sorted in non-decreasing order.*

Example 1:

Input: nums = [-4,-1,0,3,10]

Output: [0,1,9,16,100]

Explanation: After squaring, the array becomes [16,1,0,9,100]. After sorting, it becomes [0,1,9,16,100]

ANS =

```
class Solution {
public:
    bool checkPossibility(vector<int>& nums) {
        int cnt = 0;
        for(int i = 1; i < nums.size(); i++){
            if(nums[i] < nums[i-1]){
                if(++cnt > 1) return false;
                if(i == 1 || nums[i-2] <= nums[i]) nums[i-1] =
nums[i];
```

```
                else nums[i] = nums[i-1];
            }
        }
        return true;
    }
};
```

Question 7

You are given an m x n matrix M initialized with all 0's and an array of operations ops, where ops[i] = [ai, bi] means M[x][y] should be incremented by one for all 0 <= x < ai and 0 <= y < bi.

Count and return *the number of maximum integers in the matrix after performing all the operations*

ANS =

```
class Solution {
public:
    vector<vector<int>> updateMatrix(vector<vector<int>>& mat) {
            int n=mat.size();
        int m=mat[0].size();
        int t=m+n;
        int top,left;
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
```

```
                if(!mat[i][j])continue;
                left=t;top=t;
                if(i>0)top=mat[i-1][j];
                if(j>0)left=mat[i][j-1];
                mat[i][j]=min(top,left)+1;
            }
        }
        for(int i=n-1;i>=0;i--){
            for(int j=m-1;j>=0;j--){
                if(!mat[i][j])continue;
                left=t;top=t;
                if(i+1<n)top=mat[i+1][j];
                if(j+1<m)left=mat[i][j+1];
                mat[i][j] = min(mat[i][j], min(left,top)
+ 1);
            }
        }
        return mat;
    }
};
```

Question 8

Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn].

Return the array in the form [x1,y1,x2,y2,...,xn,yn].

Example 1:

Input: nums = [2,5,1,3,4,7], n = 3

Output: [2,3,5,4,1,7]

Explanation: Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is [2,3,5,4,1,7].

ANS =

```
class Solution {
public:
    int consecutiveNumbersSum(int n) {
        int count = 0;
        for(int i = 2 ; i < n ; i++){
            int sum_1 = i*(i+1)/2;
            if(sum_1 > n)
                break;
            if((n-sum_1)%i == 0)
                count++;
        }
        return count+1;
    }
};
```