Vectors
(Assignment Solutions)

Question 1 :

```cpp
vector<int> findErrorNums(vector<int>& nums) {

    vector<int> ans;

    int n = nums.size();

    vector<bool> isPresent(n+1, false);


    for(int i=0; i<n; i++) {

        if(isPresent[nums[i]]) {

            ans.push_back(nums[i]);

        }


        isPresent[nums[i]] = true;

    }


    for(int i=1; i<isPresent.size(); i++) {

        if(!isPresent[i]) {

            ans.push_back(i);

            break;

        }

    }


    return ans;

}
```

Note - This solution is brute force & more optimized approaches for the same Qs exist that use Bit Manipulation.

Question 2 :

```cpp
int maxArea(vector<int>& height) {

    int left = 0;
    int right = height.size() - 1;
    int maxArea = 0;


    while (left < right) {
        int currentArea = min(height[left], height[right]) * (right -
left);
        maxArea = max(maxArea, currentArea);


        if (height[left] < height[right]) {
            left++;
        } else {
            right--;
        }
    }


    return maxArea;
}
```

Question 3 :

```cpp
vector<vector<int>> threeSum(vector<int>& nums) {

    vector<vector<int>> ans;
    int n = nums.size();


    sort(nums.begin() , nums.end());


    for(int i=0; i<n; i++){
        if(i>0 && nums[i] == nums[i-1])  {
            continue;
        }


        int j = i+1;
        int k= n-1;
```

```cpp
        while(j < k){
            int sum = nums[i] + nums[j] + nums[k];
            if(sum < 0){
                j++;
            }
            else if(sum > 0){
                k--;
            }
            else{
                vector<int> temp = {nums[i] , nums[j] , nums[k]};
                ans.push_back(temp);
                j++;
                k--;

                while(j<k && nums[j] == nums[j-1]) {
                    j++;
                }

                while(j<k && nums[k] == nums[k+1]) {
                    k--;
                }
            }
        }
    }
    return ans;
}
```