1 https://leetcode.com/problems/concatenation-of-array/description/?envType=list&envId=pmtl1c6j

Solution:

#include <bits/stdc++.h>

using namespace std;

class Solution

{

public:

  vector<int> getConcatenation(vector<int> &nums)

  {

    vector<int> a;

    for (int i = 0; i < nums.size(); i++)

    {

      a.push\_back(nums[i]);

    }

    for (int i = 0; i < nums.size(); i++)

    {

      a.push\_back(nums[i]);

    }

    return a;

  }

};

int main()

{

  cout << "Enter Size of Vector nums: " << endl;

  int size;

  cin >> size;

  vector<int> nums(size);

  cout << "Enter elements for Vector nums: " << endl;

  for (int i = 0; i < size; i++)

  {

    cin >> nums[i];

  }

  Solution solu;

  vector<int> result = solu.getConcatenation(nums);

  cout << "Concatenated Array: ";

  for (int num : result)

  {

    cout << num << " ";

  }

  cout << endl;

  return 0;

}

2 <https://leetcode.com/problems/build-array-from-permutation/description/?envType=list&envId=pmtl1c6j>

#include<bits/stdc++.h>

using namespace std;

class Solution{

  public:

  vector<int> buildArray(vector<int> &nums)

  {

    int n = nums.size();

    vector<int> a(n);

    for(int i=0;i<nums.size();i++)

    {

      a[i] = nums[nums[i]];

    }

    return a;

  }

};

int main()

{

  cout<<"Enter Size: ";

  int size;

  cin>>size;

  cout<<"Enter Elements: ";

  vector<int> nums(size);

  for(int i=0;i<size;i++)

  {

    cin>>nums[i];

  }

  Solution solu;

  vector<int> result = solu.buildArray(nums);

  cout<<"Build Array: "<<endl;

  for(int num: result)

  {

    cout<<num<<" ";

  }

  cout<<endl;

  return 0;

}

3 <https://leetcode.com/problems/count-square-sum-triples/description/?envType=list&envId=pmtl1c6j>

#include <bits/stdc++.h>

using namespace std;

class Solution

{

public:

  int countTriples(int n)

  {

    int count = 0;

    for (int a = 1; a <= n; a++)

    {

      for (int b = a; b <= n; b++)

      {

        int c\_square = a \* a + b \* b;

        int c = sqrt(c\_square);

        if (c <= n && c \* c == c\_square)

        {

          // counting two times, because if a^2 + b^2 = c^2, then b^2 + a^2 = c^2

          count += 2;

        }

      }

    }

    return count;

  }

};

int main()

{

  int n;

  cin >> n;

  Solution solu;

  int count = solu.countTriples(n);

  cout << count;

  return 0;

}

4