**ADDITIVE SEQUENCING**

#include<bits/stdc++.h>

using namespace std;

bool additive(string a, int last, int current,int last2)

{

    int len = a.size();

    if(len==0 && current==3)

        return true;

    if(len==0)

        return false;

    bool op = false;

    if(current==1)

    {

        for(int i=1;i<a.size();i++)

        {

            stringstream s(a.substr(0,i));

            int n;

            s>>n;

            op = op || additive(a.substr(i,len-i),n,2,last2);

        }

        return op;

    }

    if(current==2)

    {

        bool op= false;

        for(int i=1;i<a.size();i++)

        {

            stringstream s(a.substr(0,i));

            int n;

            s>>n;

            int temp = last + n;

            op = op || additive(a.substr(i,len-i),temp,3,n);

        }

        return op;

    }

    if(current==3)

    {

        bool op = false;

        for(int i=1;i<=a.size();i++)

        {

            stringstream s(a.substr(0,i));

            int n;

            s>>n;

            if(n==last)

            {

                int temp = last2 + n;

                op = op || additive(a.substr(i,len-i),temp,3,n);

            }

        }

        return op;

    }

}

bool isAdditiveSequence(string n)

{

    return additive(n,0,1,0);

}

int main()

{

    string s="1235813";

    cout << isAdditiveSequence(s) << endl;

}

**TRACING:**

**HANDSHAKES PROBLEM**

package javapractice;

import java.util.Scanner;

public class HandShakes {

static int count(int N) {

if (N % 2 == 1) {

return 0;

} else if (N == 0) {

return 1;

}

int res = 0;

for (int i = 0; i < N; i += 2) {

res += count(i) \* count(N - 2 - i);

}

return res;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int tests = sc.nextInt();

while (tests-- > 0) {

int x = sc.nextInt();

System.out.println(count(x));

}

}

}

**TRACING:**