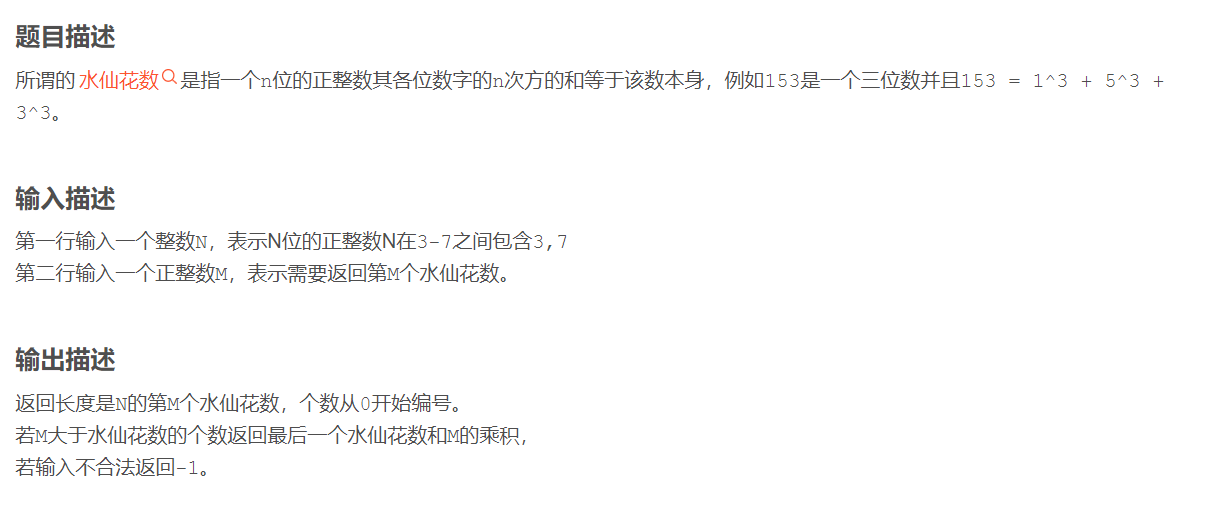
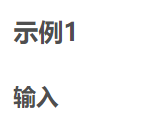
# **E卷-水仙花数[100分]（ Java | Python3 | C++ | C语言 | JsNode | Go ）**



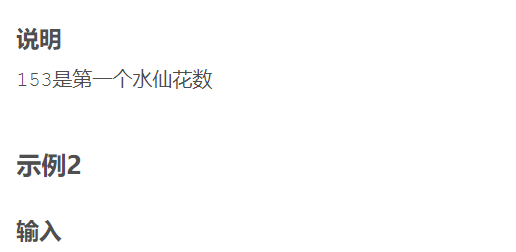


3

0



153



9

1



-1



import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Main {

// 判断一个数字是否是水仙花数

public static boolean ok(int num, int n) {

int x = num;

int res = 0;

while (x != 0) {

int y = x % 10; // 获取最后一位数字

res += Math.pow(y, n); // 计算该数字的n次方，并累加到res

x /= 10; // 去掉最后一位数字

}

return res == num; // 判断累加结果是否等于原数字

}

// 主解决方案函数

public static int solution(int n, int m) {

// 检查输入的合法性

if (!(3 <= n && n <= 7) || m < 0) {

return -1;

}

// 找到所有n位的水仙花数

int start = (int)Math.pow(10, n - 1); // 范围的起始点

int end = (int)Math.pow(10, n); // 范围的终止点

List<Integer> a = new ArrayList<>(); // 用于存储找到的水仙花数

for (int num = start; num < end; num++) {

if (ok(num, n)) {

a.add(num); // 如果num是水仙花数，则加入列表

}

}

// 检查找到的水仙花数的数量是否足够

if (m >= a.size()) {

if (!a.isEmpty()) {

return a.get(a.size() - 1) \* m; // 如果不够，返回最后一个水仙花数乘以m

} else {

return -1; // 如果没有找到满足条件的水仙花数，返回-1

}

} else {

return a.get(m); // 返回第m个水仙花数

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt(); // 读取n

int m = sc.nextInt(); // 读取m

System.out.println(solution(n, m)); // 输出结果

}

}



def ok(num, n):

x = num

res = 0

while x:

y = x % 10

res += int(y\*\*n)

x //= 10

return res == num

def solution(n, m):

# 检查输入的合法性

if not (3 <= n <= 7) or m < 0:

return -1

# 找到所有n位的水仙花数

start = 10\*\*(n-1)

end = 10\*\*n

a = []

for num in range(start, end):

if ok(num, n):

a.append(num)

# 如果找到的水仙花数不够m，则返回最后一个水仙花数乘以m

if m >= len(a):

if a:

return a[-1] \* m

else:

return -1

else:

return a[m]

n = int(input())

m = int(input())

# 输出

print(solution(n, m))



#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

// 判断一个数字是否是水仙花数的函数

bool ok(int num, int n) {

int x = num;

int res = 0;

while (x != 0) {

int y = x % 10; // 获取最后一位数字

res += pow(y, n); // 计算该数字的n次方，并累加到res

x /= 10; // 去掉最后一位数字

}

return res == num; // 判断累加结果是否等于原数字

}

// 主解决方案函数

int solution(int n, int m) {

// 检查输入的合法性

if (!(3 <= n && n <= 7) || m < 0) {

return -1;

}

// 找到所有n位的水仙花数

int start = pow(10, n - 1); // 范围的起始点

int end = pow(10, n); // 范围的终止点

vector<int> a; // 用于存储找到的水仙花数

for (int num = start; num < end; num++) {

if (ok(num, n)) {

a.push\_back(num); // 如果num是水仙花数，则加入列表

}

}

// 检查找到的水仙花数的数量是否足够

if (m >= a.size()) {

if (!a.empty()) {

return a.back() \* m; // 如果不够，返回最后一个水仙花数乘以m

} else {

return -1; // 如果没有找到满足条件的水仙花数，返回-1

}

} else {

return a[m]; // 返回第m个水仙花数

}

}

int main() {

int n, m;

cin >> n >> m; // 读取n和m

cout << solution(n, m) << endl; // 输出结果

return 0;

}



#include <stdio.h>

#include <math.h>

// 判断一个数字是否是水仙花数的函数

int ok(int num, int n) {

int x = num;

int res = 0;

while (x != 0) {

int y = x % 10; // 获取最后一位数字

res += pow(y, n); // 计算该数字的n次方，并累加到res

x /= 10; // 去掉最后一位数字

}

return res == num; // 判断累加结果是否等于原数字

}

// 主解决方案函数

int solution(int n, int m) {

// 检查输入的合法性

if (!(3 <= n && n <= 7) || m < 0) {

return -1;

}

// 找到所有n位的水仙花数

int start = pow(10, n - 1); // 范围的起始点

int end = pow(10, n); // 范围的终止点

int a[1000]; // 用于存储找到的水仙花数

int count = 0; // 记录找到的水仙花数的数量

for (int num = start; num < end; num++) {

if (ok(num, n)) {

a[count++] = num; // 如果num是水仙花数，则加入列表

}

}

// 检查找到的水仙花数的数量是否足够

if (m >= count) {

if (count > 0) {

return a[count - 1] \* m; // 如果不够，返回最后一个水仙花数乘以m

} else {

return -1; // 如果没有找到满足条件的水仙花数，返回-1

}

} else {

return a[m]; // 返回第m个水仙花数

}

}

int main() {

int n, m;

scanf("%d %d", &n, &m); // 读取n和m

printf("%d\n", solution(n, m)); // 输出结果

return 0;

}



const readline = require("readline");

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout,

});

// 判断一个数字是否是水仙花数的函数

const ok = (num, n) => {

let x = num;

let res = 0;

while (x > 0) {

const y = x % 10; // 获取最后一位数字

res += Math.pow(y, n); // 计算该数字的n次方，并累加到res

x = Math.floor(x / 10); // 去掉最后一位数字

}

return res === num; // 判断累加结果是否等于原数字

};

// 主解决方案函数

const solution = (n, m) => {

// 检查输入的合法性

if (!(3 <= n && n <= 7) || m < 0) {

return -1;

}

const start = Math.pow(10, n - 1); // 范围的起始点

const end = Math.pow(10, n); // 范围的终止点

const a = []; // 用于存储找到的水仙花数

for (let num = start; num < end; num++) {

if (ok(num, n)) {

a.push(num); // 如果num是水仙花数，则加入列表

}

}

// 检查找到的水仙花数的数量是否足够

if (m >= a.length) {

if (a.length > 0) {

return a[a.length - 1] \* m; // 如果不够，返回最后一个水仙花数乘以m

} else {

return -1; // 如果没有找到满足条件的水仙花数，返回-1

}

} else {

return a[m]; // 返回第m个水仙花数

}

};

rl.question("", (input1) => {

rl.question("", (input2) => {

const n = parseInt(input1.trim()); // 读取n

const m = parseInt(input2.trim()); // 读取m

console.log(solution(n, m)); // 输出结果

rl.close();

});

});



package main

import (

"fmt"

"math"

)

// 判断一个数字是否是水仙花数的函数

func ok(num, n int) bool {

x := num

res := 0

for x != 0 {

y := x % 10 // 获取最后一位数字

res += int(math.Pow(float64(y), float64(n))) // 计算该数字的n次方，并累加到res

x /= 10 // 去掉最后一位数字

}

return res == num // 判断累加结果是否等于原数字

}

// 主解决方案函数

func solution(n, m int) int {

// 检查输入的合法性

if !(3 <= n && n <= 7) || m < 0 {

return -1

}

start := int(math.Pow(10, float64(n-1))) // 范围的起始点

end := int(math.Pow(10, float64(n))) // 范围的终止点

var a []int // 用于存储找到的水仙花数

for num := start; num < end; num++ {

if ok(num, n) {

a = append(a, num) // 如果num是水仙花数，则加入列表

}

}

// 检查找到的水仙花数的数量是否足够

if m >= len(a) {

if len(a) > 0 {

return a[len(a)-1] \* m // 如果不够，返回最后一个水仙花数乘以m

} else {

return -1 // 如果没有找到满足条件的水仙花数，返回-1

}

} else {

return a[m] // 返回第m个水仙花数

}

}

func main() {

var n, m int

fmt.Scanln(&n) // 读取n

fmt.Scanln(&m) // 读取m

fmt.Println(solution(n, m)) // 输出结果

}