1

Числа Фибоначчи

2

Мячик на лесенке

3

Платная лестница

4

Калькулятор с восстановлением ответа

5

Ход коня

6

Маршрут черепашки

7

Взрывоопасность

8

Кубики

Задача 1.

Лёгкая

**Числа Фибоначчи**

Последовательность Фибоначчи определена следующим образом: φ0=1, φ1=1, φn=φn−1+φn−2 при n>1. Начало ряда Фибоначчи выглядит следующим образом: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, … Напишите функцию function phi (n:longint): longint, которая по данному натуральному n возвращает φn.

ВХОДНЫЕ ДАННЫЕ

Вводится одно число n.

ВЫХОДНЫЕ ДАННЫЕ

Необходимо вывести  значение φn.

ПРИМЕРЫ

|  |  |
| --- | --- |
| **ВВОД** | **ВЫВОД** |
| 3 | 3 |

Начало формы

Задача решена верно

OK. Your score is = 100, 25/25 tests passed

Решение задачи

Обсудим, как должна быть устроена искомая функция. Если n = 0 или n = 1, то функция должна вернуть 1. Если n > 1, то с помощью цикла for можно последовательно вычислить φ2, .., φn, при этом не обязательно их все хранить: пусть f1=φ0, f2=φ1, f=φ2. Тогда f получается как сумма f1 и f2. Если теперь записать f1:=f2, f2:=f и опять вычислить сумму f1 и f2, то будет найдено φ3 и так далее. Полное решение на языке Pascal может выглядеть так:

Конец формы

Задача 2.

Вводная

**Мячик на лесенке**

На вершине лесенки, содержащей N ступенек, находится мячик, который начинает прыгать по ним вниз, к основанию. Мячик может прыгнуть на следующую ступеньку, на ступеньку через одну или через 2. (То есть, если мячик лежит на 8-ой ступеньке, то он может переместиться на 5-ую, 6-ую или 7-ую.) Определить число всевозможных "маршрутов" мячика с вершины на землю.

ВХОДНЫЕ ДАННЫЕ

Вводится одно число 0<N<31.

ВЫХОДНЫЕ ДАННЫЕ

Выведите одно число — количество маршрутов.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 4 | 7 |

Начало формы

Задача решена верно

OK. Your score is = 100, 8/8 tests passed

Решение задачи

Требуется динамическое программирование. dp[i] - количество способов добраться до i-той ступеньки. dp[i] = dp[i + 1] + dp[i + 2] + dp[i + 3]

Конец формы

Задача 3.

Вводная

**Платная лестница**

Мальчик подошел к платной лестнице. Чтобы наступить на любую ступеньку, нужно заплатить указанную на ней сумму. Мальчик умеет перешагивать на следующую ступеньку, либо перепрыгивать через ступеньку. Требуется узнать, какая наименьшая сумма понадобится мальчику, чтобы добраться до верхней ступеньки. Изначально мальчик стоит перед лестницей.

ВХОДНЫЕ ДАННЫЕ

В первой строке входного файла вводится одно натуральное число N<100 — количество ступенек.  
В следующей строке вводятся N натуральных чисел, не превосходящих 100 — стоимость каждой ступеньки (снизу вверх).

ВЫХОДНЫЕ ДАННЫЕ

Выведите одно число — наименьшую возможную стоимость прохода по лесенке.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3  1 3 1 | 2 |

Начало формы

Задача решена верно

OK. Your score is = 100, 14/14 tests passed

Решение задачи

Потребуется динамическое программирование. dp[i] - стоимость, чтобы добраться до i-той ступеньки. dp[i] = min(dp[i - 1], dp[i - 2]) + cost[i]

Задача 4.

Сложная

**Калькулятор с восстановлением ответа**

**Ограничение по времени работы программы: 5 секунд**

Эта задача аналогична предыдущей задаче «Калькулятор», но в этой задаче требуется также восстановить ответ.

Исполнитель «Калькулятор» может с заданным числом X выполнить одну из трех операций и получить новое число. Возможные операции:

* Прибавить к числу X единицу.
* Умножить число X на 2.
* Умножить число X на 3.

При помощи наименьшего числа операций получите из числа 1 заданное число N.

### ВХОДНЫЕ ДАННЫЕ

Программа получает на вход одно число N, не превосходящее 106.

### ВЫХОДНЫЕ ДАННЫЕ

Сначала выведите необходимое количество операций k.

Во второй строке выведите k+1 число, последовательно получающиеся при выполнении операций. Первое из них должно быть равно 1, а последнее N. Если решений несколько, выведите любое из них.

### ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 1 | 0 1 |
| 5 | 3 1 3 4 5 |
| 962340 | 17 1 3 9 27 54 55 165 495 1485 4455 8910 17820 17821 53463 160389 160390 481170 962340 |

Задача 5.

Лёгкая

**Ход коня**

**Ограничение по времени работы программы: 1 секунда**

Дана прямоугольная доска N×M (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. При этом конь может ходить ТОЛЬКО на две клетки вниз и на одну клетку вправо, либо на две клетки вправо и на одну клетку вниз (см. рисунок).

Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

### ВХОДНЫЕ ДАННЫЕ

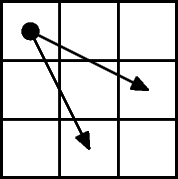
Программа получает на вход два натуральных числа N и M (1⩽N,M⩽50), записанных в одной строке.

### ВЫХОДНЫЕ ДАННЫЕ

Выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

### ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 2 | 1 |
| 31 34 | 293930 |
| 1 1 | 1 |



Начало формы

Задача решена верно

OK. Your score is = 100, 21/21 tests passed

Решение задачи

Решим задачу методом динамического программирования. Целевая функция — F(i,j) — количество путей коня из начальной клетки в клетку (i,j). В клетку (i,j) можно прийти из клеток (i−1,j−2) и (i−2,j−1), поэтому F(i,j)=F(i−1,j−2)+F(i−2,j−1). Начальное значение - F(n,m)=1. В качестве ответа необходимо вывести значение F(n,m).

Для обработки крайних клеток доски удобно ввести каемочку — строку и столбец с номером 0. При этом ширина каемочки должна быть две клетки, так как при вычислении значения функции приходится вычитать 2 из номера строки и номера столбца. В качестве второй строки (столбца) каемочки будет использоваться строка и столбец с номером 1 — они все заполнены нулями, кроме начальной клетки.

**Подсказка**

Используйте динамическое программирование. Целевая функция — F(i,j) — количество путей коня из начальной клетки в клетку (i,j).

**Подсказка**

В клетку (i,j) можно прийти из клеток (i−1,j−2) и (i−2,j−1), поэтому F(i,j)=F(i−1,j−2)+F(i−2,j−1).

**Подсказка**

Начальное значение — F(1,1)=1. В качестве ответа необходимо вывести значение F(n,m). Для обработки крайних клеток доски удобно ввести каемочку — строку и столбец с номером 0. При этом ширина каемочки должна быть две клетки, так как при вычислении значения функции приходится вычитать 2 из номера строки и номера столбца. В качестве второй строки (столбца) каемочки будет использоваться строка и столбец с номером 1 — они все заполнены нулями, кроме начальной клетки.

Задача 6.

Сложная

**Маршрут черепашки**

**Ограничение по времени работы программы: 1 секунда**

В левом верхнем углу прямоугольной таблицы размером N×M находится черепашка. В каждой клетке таблицы записано некоторое число. Черепашка может перемещаться вправо или вниз (не по диагонали), при этом маршрут черепашки заканчивается в правом нижнем углу таблицы.

Подсчитаем сумму чисел, записанных в клетках, через которую проползла черепашка (включая начальную и конечную клетку). Найдите наибольшее возможное значение этой суммы и маршрут, на котором достигается эта сумма.

### ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных записаны два натуральных числа N и M, не превосходящих 100 — размеры таблицы. Далее идет N строк, каждая из которых содержит M чисел, разделенных пробелами — описание таблицы. Все числа в клетках таблицы целые и могут принимать значения от 0 до 100.

### ВЫХОДНЫЕ ДАННЫЕ

Первая строка выходных данных содержит максимальную возможную сумму, вторая – маршрут, на котором достигается эта сумма. Маршрут выводится в виде последовательности, которая должна содержать N−1 букву D, означающую передвижение вниз и M−1 букву R, означающую передвижение направо. Если таких последовательностей несколько, необходимо вывести ровно одну (любую) из них.

### ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 4 1 1 2 1 2 2 1 1 2 1 2 1 | 9 D R D R R |

Начало формы

Задача решена верно

OK. Your score is = 100, 17/17 tests passed

Решение задачи

Решил задачу методом динамического программирования. Целевая функция — F(i,j) — максимальное значение суммы на пути из угловой клетки в клетку (i,j).

В клетку (i,j) можно прийти из клеток (i−1,j) и (i,j−1), поэтому F(i,j)=max(F(i−1,j),F(i,j−1))+A(i,j) (где A - данная таблица с числами). После заполнения таблицы F необходимо восстановить ответ.

Для восстановления ответа будем идти от конечной клетки в начальную, выбирая ту из двух клеток, которые могли быть предшественниками данной клетки, в которой значение F больше.

Задача 7.

Лёгкая

**Взрывоопасность**

При переработке радиоактивных материалов образуются отходы трех видов — особо опасные (тип A), неопасные (тип B) и совсем не опасные (тип C). Для их хранения используются одинаковые контейнеры. После помещения отходов в контейнеры последние укладываются вертикальной стопкой. Стопка считается взрывоопасной, если в ней подряд идет более одного контейнера типа A. Стопка считается безопасной, если она не является взрывоопасной. Для заданного количества контейнеров N определить число безопасных стопок.

### ВХОДНЫЕ ДАННЫЕ

Одно число 1<N<20.

### ВЫХОДНЫЕ ДАННЫЕ

Одно число — количество безопасных вариантов формирования стопки.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 2 | 8 |

Начало формы

Задача решена верно

OK. Your score is = 100, 10/10 tests passed

Решение задачи

Требуется динамическое программирование. Пусть коробка типа A имеет номер 0, типа B имеет номер 1, типа C имеет номер 2. dp[i][j] - количество стопок высоты i с последней коробкой типа j.

Тогда dp[i][0] = dp[i - 1][1] + dp[i - 1][2]

dp[i][1] = dp[i][2] = dp[i - 1][0] + dp[i - 1][1] + dp[i - 1][2]

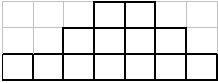
Задача 8.

Лёгкая

**Кубики**

**Ограничение по времени работы: 1 секунда**

Родители подарили мальчику Пете очень много одинаковых кубиков. Наиболее интересным сооружением из кубиков Петя счел двусторонние лесенки.



В основании (нижнем ряду) такой лесенки расположено $N$ кубиков, а каждый следующий ряд кубиков укладывается на предыдущий так, что один кубик укладывается ровно на один нижестоящий кубик, а по крайней мере на самый правый и самый левый кубики предыдущего ряда новые кубики не кладутся (чтобы получилась ступенька).

Петя поручил старшему брату подсчитать, сколько можно построить различных лесенок, состоящих из ровно $K$ рядов кубиков, в основании которых лежит ровно $N$ кубиков. При этом, если одну лесенку можно получить из другой путем зеркального отображения, то они все равно считаются различными.

### ВХОДНЫЕ ДАННЫЕ

Вводятся два числа N и K (1⩽N⩽100, 1⩽K⩽100).

### ВЫХОДНЫЕ ДАННЫЕ

Выведите одно число – количество различных лесенок. Гарантируется, что правильный ответ не будет превышать 1018.

### ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 10 4 | 84 |

Начало формы

Загрузите исходный код программы

**Подсказка**

Задача решается двумерным динамическим программированием.

f(n,k) — количество пирамидок высоты k с основанием n.

Начальное значение: f(n,1)=1.

**Подсказка**

Рекуррентное соотношение — пусть требуется найти значение f(n,k).

Основание пирамидки состоит из n кубиков, пронумеруем их числами от 1 до n. Теперь переберем основание второго ряда. Пусть номер левого кубика в основании второго ряда l, а номер правого кубика — r. Тогда двумя циклами перебираем значения: 2⩽l⩽r⩽n−1.

К значению f(n,k) нужно добавить величину f(r−l+1,k−1) — количество пирамидок с основанием от l до r высоты k−1.

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы