

**КЛАСС\_11\_ ВАРИАНТ\_1\_**

**ШИФР ­­[w34nba6k6pgf88n]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **∑** | **Подпись** |
|  |  |  |  |  |  |

|  |
| --- |
| **Задание 1** |
| В генеалогическом древе у каждого человека, кроме родоначальника,  есть ровно один родитель.  Для каждого элемента дерева определите число всех его потомков (не  считая его самого). | |
| **Код решения:** | |
| def height(man):  if man not in p\_tree:  return 0  else:  return 1 + height(p\_tree[man])  p\_tree = {}  n = int(input())  for i in range(n - 1):  child, parent = input().split()  p\_tree[child] = parent  heights = {}  for man in set(p\_tree.keys()).union(set(p\_tree.values())):  heights[man] = height(man)  for key, value in sorted(heights.items()):  print(key, value) | |
| **Пример работы программы:** | |
|  | |
| **Комментарии (при необходимости):** | |
| Решение этой задачи было реализовано на языке программирования Python. | |

|  |
| --- |
| **Задание 2** |
| Даны целые числа 0, 1, 2, 3, … , 63. Получить квадратную матрицу  (таблицу) 8 на 8, элементами которой являются указанные числа,  расположенные в ней по схемам, приведённым ниже. | |
| **Код решения:** | |
| n = int(input())  for i1 in ['', '', '+']:  for i2 in ['', '-', '+']:  for i3 in ['', '-', '+']:  for i4 in ['', '-', '+']:  for i5 in ['', '-', '+']:  for i6 in ['', '-', '+']:  for i7 in ['', '-', '+']:  for i8 in ['', '-', '+']:  s = '1' + i8 + '2' + i7 + '3' + i6 + '4' + i5 + '5' + i4 + '6' + i3 + '7' + i2 + '8' + i1 + '9'  if eval(s) == n:  print(s)  exit()  print('Нет решения') | |
| **Пример работы программы:** | |
|  | |
| **Комментарии (при необходимости):** | |
| Решение этой задачи было реализовано на языке программирования Python. | |

|  |
| --- |
| **Задание 3** |
| Дано натуральное число m. Вставить между некоторыми цифрами  1, 2, 3, 4, 5, 6, 7, 8, 9, записанными именно в таком порядке, знаки «+» и «-»  так, чтобы значением получившегося выражения было число m. Например,  если m = 122, то подойдёт следующая расстановка знаков: 12+34-5-6+78+9.  Если требуемая расстановка знаков невозможна, то вывести «Нет решения». | |
| **Код решения:** | |
| n = int(input())  for i1 in ['', '', '+']:  for i2 in ['', '-', '+']:  for i3 in ['', '-', '+']:  for i4 in ['', '-', '+']:  for i5 in ['', '-', '+']:  for i6 in ['', '-', '+']:  for i7 in ['', '-', '+']:  for i8 in ['', '-', '+']:  s = '1' + i8 + '2' + i7 + '3' + i6 + '4' + i5 + '5' + i4 + '6' + i3 + '7' + i2 + '8' + i1 + '9'  if eval(s) == n:  print(s)  exit()  print('Нет решения') | |
| **Пример работы программы:** | |
|  | |
| **Комментарии (при необходимости):** | |
| Решение этой задачи было реализовано на языке программирования Python.  Тест №1 выводит другой вариант решения. | |

|  |
| --- |
| **Задание 4** |
| В левом верхнем углу прямоугольной таблицы размером N×M  находится черепашка. В каждой клетке таблицы записано некоторое число.  Черепашка может перемещаться вправо или вниз, 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. | |
| **Код решения:** | |
| n, m = map(int, input().split())  p = [list(map(int, input().split())) for i in range(n)]  a = [[0] \* m for i in range(n)]  b = [[''] \* m for i in range(n)]  a[0][0] = p[0][0]  for j in range(1, m):  a[0][j] = a[0][j - 1] + p[0][j]  b[0][j] = b[0][j - 1] + 'R '  for i in range(1, n):  a[i][0] = a[i - 1][0] + p[i][0]  b[i][0] = b[i - 1][0] + 'D '  for i in range(1, n):  for j in range(1, m):  if a[i][j - 1] > a[i - 1][j]:  a[i][j] = a[i][j - 1] + p[i][j]  b[i][j] = b[i][j - 1] + 'R '  else:  a[i][j] = a[i - 1][j] + p[i][j]  b[i][j] = b[i - 1][j] + 'D '  print(a[-1][-1])  print(b[-1][-1]) | |
| **Пример работы программы:** | |
|  | |
| **Комментарии (при необходимости):** | |
| Решение этой задачи было реализовано на языке программирования Python. | |