**Отчет**

Выполнил студент Шакирзянов Руслан 4214

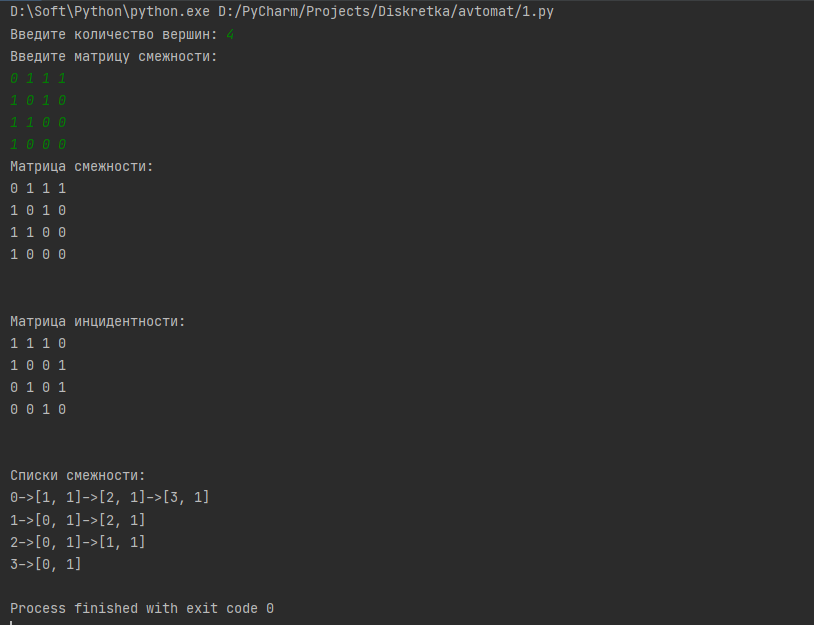
**Задача**

|  |  |  |
| --- | --- | --- |
| Наименование программы | Вход | Выход |
| Преставление графов в ЭВМ | Одна из форм представления графов (матрица смежности) | Все три формы представления графов |

**Процедурное решение задачи**

Получаем матрицу смежности на вход. Для преобразования матрицы смежности в матрицу инцидентности проходим по матрице смежности, при неравном 0 значении элемента матрицы смежности создаем столбец матрицы инцидентности со значениями 1 и -1, если это кратное ребро то копируем столбец в доп. список, по завершению добавляем кратные ребра в основную матрицу и проверяем есть ли противоположно направленные ребра и при наличие объединяем их в одно двунаправленное ребро. Выводим матрицу инцидентности. Для преобразования матрицы смежности в списки смежности проходим по матрице смежности и для каждой вершины создаем список вершин смежных с ней.

**Таблица значений**



**Текст программы**

def SmejToInced(vertex\_num, smejnost\_matrix):  
 incedentnost\_matrix = [[] for i in range(vertex\_num)] # создание пустой матрици инцидентности  
 krat\_edge\_list = [[] for i in range(vertex\_num)] # пустой список кратных верин для дальнейших проверок  
 krat\_edge\_i = 0  
 for i in range(vertex\_num):  
 for j in range(vertex\_num): # проход по матрице смежности  
 if smejnost\_matrix[i][j] != 0: # поиск смежной вершины  
 for l in range(vertex\_num): # если найдена такая вершина заполняем столбец в матрице инцедентности  
 if l == i:  
 incedentnost\_matrix[l].append(1) # добавляем в матрицу инцедентности 1 если это инцедентная вершина  
 if smejnost\_matrix[i][j] == 2: # если это не кратное ребро  
 krat\_edge\_list[krat\_edge\_i].append(1) # записываем в список кратных ребер  
 elif l == j:  
 incedentnost\_matrix[l].append(-1) # добавляем в матрицу инцедентности 1 если это инцедентная вершина  
 if smejnost\_matrix[i][j] == 2: # если это не кратное ребро  
 krat\_edge\_list[krat\_edge\_i].append(-1) # записываем в список кратных ребер  
 else:  
 incedentnost\_matrix[l].append(0) # иначе добавляем 0  
 if smejnost\_matrix[i][j] == 2: # если это не кратное ребро  
 krat\_edge\_list[krat\_edge\_i].append(0) # записываем в список кратных ребер  
 krat\_edge\_i += 1 if smejnost\_matrix[i][j] == 2 else 0 # если записалось кратное ребро увеличиваем итератор списка кратных ребер  
 incedentnost\_matrix = list(zip(\*incedentnost\_matrix)) # транспонирование матрицы для удаления повторений  
 for i in krat\_edge\_list:  
 incedentnost\_matrix.append(tuple(i)) # добавление кратных ребер  
 i = 0  
 while i < len(incedentnost\_matrix) - 1:  
 t = []  
 for j in incedentnost\_matrix[i]: # создание обратноориентированного ребра для дальнейшей проверки  
 t.append(-j)  
 if tuple(t) in incedentnost\_matrix: # проверка на то что ребро у ориентированного графа двунаправленное  
 del (incedentnost\_matrix[incedentnost\_matrix.index(tuple(t))]) # удаление столбца с ребром в одном раправлении  
 incedentnost\_matrix[i] = tuple(  
 abs(x) for x in incedentnost\_matrix[i]) # перезапись столбца с двунаправленным ребром  
 else:  
 i += 1  
 incedentnost\_matrix = list(zip(\*filter(None, incedentnost\_matrix))) # транспонирование матрицы обратно  
 print("\n\nМатрица инцидентности:\n" + '\n'.join(' '.join(str(j) for j in incedentnost\_matrix[i]) for i in range(len(incedentnost\_matrix)))) # вывод машкицы инцедентности  
  
  
def SmejToSmejList(vertex\_num, smejnost\_matrix):  
 smejnost\_lists = [[i] for i in range(vertex\_num)] # создание списка смежности с номерами вершин  
 for i in range(len(smejnost\_matrix)):  
 for j in range(len(smejnost\_matrix[i])): # проход по матрице смежности  
 if smejnost\_matrix[i][j] != 0: # если найдена смежная вершина  
 smejnost\_lists[i].append([j, smejnost\_matrix[i][j] if smejnost\_matrix[i][j] != 2 else 1]) # добавить в список смежности пару в формате [вершина, вес]  
 print("\n\nСписки смежности:\n" + '\n'.join('->'.join(str(j) for j in smejnost\_lists[i]) for i in range(len(smejnost\_lists)))) # вывод списка смежности  
  
  
def main():  
 vertex\_num = int(input("Введите количество вершин: ")) # запрос навведенение количества вершин  
 print("Введите матрицу смежности:")  
 smejnost\_matrix = [[int(i) for i in input().split()] for j in range(vertex\_num)] # ввод матрицы смежности с клавиатуры  
  
 print("Матрица смежности:\n" + '\n'.join(' '.join(str(j) for j in smejnost\_matrix[i]) for i in range(vertex\_num))) # вывод матрицы смежности  
  
 SmejToInced(vertex\_num, smejnost\_matrix)  
 SmejToSmejList(vertex\_num, smejnost\_matrix)  
  
  
main()