**НИУ «МЭИ»**

**Лабораторная работа № 7**

Нисходящее проектирование: программы с использованием

подпрограмм и модулей

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил: \_\_Кобзев Н.И.\_\_

Группа: \_\_\_Э-07-20\_\_\_

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ф.И.О./

Москва, 2020 г.

1. **Формулировка исходной задачи:**

Дана целочисленная матрица А (n, m). Если в чётной строке матрицы есть хотя бы один элемент, равный заданному числу, то в следующей нечётной строке найти минимальный элемент среди тех, значение которых попадает на отрезок [C, D]. В противном случае найти в ней номер первого элемента, значение которого больше Р.

1. **Уточненная постановка задачи:**

Вводится целочисленная матрица a (n, m). Если в четной строке матрицы есть хотя бы один элемент, равный num, то в следующей нечётно строке найти минимальный элемент сред тех, значение которых попадает на отрезок в диапазон c ≤ число ≤ d. В противном случае найти в ней номер первого элемента, значение которого больше p.

1. **Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Имя, смысл | Тип | Структура | Диапазон |
| Входные данные | n, m – количество строк и столбцов | Целый | Простые переменные | [2, 10] |
| a – матрица | Целый | Двумерный массив | (-∞, +∞) |
| num – заданное число | Целый | Простая переменная | (-∞, +∞) |
| c, d – границы отрезка | Целый | Простые переменные | (-∞, +∞) |
| p – число | Целый | Простая переменная | (-∞, +∞) |
| Промежуточные данные | num – число, возвращаемое функцией – условием | Целый | Простая переменная | (-∞, +∞) |
| min – минимальный элемент в первом условии | Целый | Простая переменная | (-∞, +∞) |
| row – строка в массиве | Целый | Одномерный массив | (-∞, +∞) |
| elem – элемент цикле | целый | Простая переменная | (-∞, +∞) |
| i – счетчик цикла | Целый | Простая переменная | [2, 10] |
| Выходные данные | num – конечное число | Целый | Простая переменная | (-∞, +∞) |

1. **Блок-схемы:**

* **Головной модуль:**

for i, row in enumerate(array)

num2 = min\_in\_the\_segment(array[i + 1], c, d, p)

**+**

**+**

**\_**

**\_**

Вывод num2

Выход из цикла

i % 2 = 0 and i < len(array) – 1 and even\_row\_in\_matrix(row, num)

element\_position\_bigger\_than\_p(array[i+1], p) is not None

num2 = element\_position\_bigger\_than\_p(array[i+1], p)

**+**

Вывод num2

Выход из цикла

**\_**

matrix\_output()

read\_data\_from\_file()

* **Подпрограмма ввода данных:**

Возврат a, NUM, C, D, P

Ввод n, m, a, NUM, C, D, P

Открытие файла

Чтение командной строки

* **Подпрограмма вывода матрицы:**

**\_**

**\_**

**+**

**+**

Ввод col

for col in row

for row in array

* **Подпрограмма нахождения четного числа в строке матрицы:**

**\_**

**\_**

Возврат False

**+**

Возврат True

**+**

Elem = n

for elem in row

* **Подпрограмма нахождения минимального элемента, принадлежащего отрезку [C, D]:**

**+**

**+**

num = min = elem

for elem in row

**\_**

**\_**

**\_**

Возврат num

**+**

num is not None

elem < min and c ≤ elem ≤ d

num = None; min = d + 1

* **Подпрограмма нахождения числа, большего, чем P:**

**+**

elem > p

**+**

**\_**

**\_**

Возврат i

for i, elem in enumerate(row)

**5.Тесты:**

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  1 2 3 4 5  6 7 8 9 10  11 12 13 14 15  16 17 18 19 20  8  0 10  6 | Позиция элемента большего, чем P, равна: 7 | Проверка невыполнения первого условия на структурированных данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  1 2 3 4 5  6 7 8 9 10  11 12 13 14 15  16 17 18 19 20  5  10 11  6 | Элемент, значение которого попадает в отрезок [10, 11], равен: 10 | Проверка выполнения первого условия на структурированных данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  20 43 1 15 23  15 14 11 8 46  50 28 13 58 74  1 3 7 -1 -23  5  10 11  6 | Позиция элемента большего, чем P, равна: 6 | Проверка невыполнения первого условия на неструктурированных данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  20 43 1 15 23  15 14 11 8 46  50 28 13 58 74  1 3 7 -1 -23  1  8 46  6 | Элемент, значение которого попадает в отрезок [8, 46], равен: 8 | Проверка выполнения первого условия на неструктурированных данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  1 1 1 1 1  1 1 1 1 1  1 1 1 1 1  1 1 1 1 1  1  1 2  5 | Элемент, значение которого попадает в отрезок [1, 2], равен: 1 | Проверка выполнения первого условия на одинаковых данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 4 5  1 1 1 1 1  1 1 1 1 1  1 1 1 1 1  1 1 1 1 1  2  1 2  0 | Элемент, значение которого попадает в отрезок [1, 2], равен: 1 | Проверка невыполнения первого условия на одинаковых данных |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 2 2  1 2  3 4  2  3 4  6 | Элемент, значение которого попадает в отрезок [3, 4], равен: 3 | Проверка выполнения первого условия на матрице с минимально допустимой размерностью |

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Смысл теста |
| 2 2  1 2  3 4  0  3 4  3 | Позиция элемента большего, чем P, равна: 6 | Проверка нвыполнения первого условия на матрице с минимально допустимой размерностью |

1. **Код программы:**

import sys  
  
  
def read\_data\_from\_file():  
 name\_file = sys.argv[1] # Чтение с командной строки  
  
 file = open(name\_file, 'r') # Открытие файла  
  
 row, col = map(int, file.readline().split()) # Ввод количества строк и столбцов  
  
 print("Количество строк равно: {}\nКоличество столбцов равно: {}".format(row, col))  
  
 a = [[] for i in range(row)] # инициализация массива  
  
 for i in range(row): # Ввод массива  
 for elem in file.readline().split():  
 a[i].append(int(elem))  
  
 num = int(file.readline()) # Ввод заданного числа  
  
 print("Заданное число равно: {}".format(num))  
  
 c, d = map(int, file.readline().split()) # Ввод границ отрезка  
  
 print("Отрезок = [{}, {}]".format(c, d))  
  
 p = int(file.readline()) # Ввод числа P  
 print("Число P равно: {}".format(p))  
  
 return a, num, c, d, p, row, col # Возврат значений в головной модуль  
  
  
def matrix\_output(a): # Подпрограмма вывода матрицы  
 for row in a:  
 for col in row:  
 print("%3d" % col, end=' ')  
 print()  
  
  
def even\_row\_in\_matrix(row, n): # Подпрограмма нахождения элемента равного num  
 for elem in row:  
 if elem == n:  
 return True  
 return False  
  
  
def min\_in\_the\_segment(row, c, d, p): # Подпрограмма проверки условия [C, D]  
 num = None  
 min = d + 1  
 for elem in row:  
 if elem < min and c <= elem <= d:  
 num = min = elem  
 if num is not None:  
 return num  
  
  
def element\_position\_bigger\_than\_p(row, p): # Подпрограмма нахождения числа большего P  
 for i, elem in enumerate(row):  
 if elem > p:  
 return i  
  
  
array, num, c, d, p, row, col = read\_data\_from\_file() # Присваивание констант  
  
print('Веденная матрица:')  
matrix\_output(array) # Вывод матрицы  
  
for i, row in enumerate(array): # Основной цикл  
 if i % 2 == 0 and even\_row\_in\_matrix(row, num) and i < len(array) - 1: # Условие проверки выполнения первого условия  
 num2 = min\_in\_the\_segment(array[i + 1], c, d, p) # Присваивание значения подпрограммы [C, D]  
 print("Элемент, значение которого попадает в отрезок [{}, {}], равен: {}\n".format(c, d, num2)) # Вывод выходных данных  
 break # Выход из цикла  
 elif element\_position\_bigger\_than\_p(array[i + 1], p) is not None: # Если условие не выполняется  
 num2 = element\_position\_bigger\_than\_p(array[i + 1], p) # Присваивания значения подпрограммы > P  
 print("Позиция элемента большего, чем P, равна: {}\n".format(num2 + (i + 1) \* col + 1)) # Вывод выходных данных  
 break # Выход из цикла