**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 | [-1000, 1000] |
| num – заданное число | Целый | Простая переменная | [-1000, 1000] |
| c, d – границы отрезка | Целый | Простые переменные | [-1000, 1000] |
| p – число | Целый | Простая переменная | [-1000, 1000] |
| Промежуточные данные | i – счетчик цикла | Целый | Простая переменная | [0, row] |
| j – счетчик цикла | Целый | Простая переменная | [0, col] |
| min – минимальное число в функции | Целый | Простая переменная | [-1000, 1000] |
| num – число возвращаемое функцией | Целый | Одномерный массив | [-1000, 1000] |
| Выходные данные | num2 – конечное число | Целый | Простая переменная | [-1000, 1000] |

1. **Блок-схемы:**

* **Головной модуль:**

Вх. array

Чтение переменных с файла

Вывод матрицы

Вых. array, num, c, d, p, row, col

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

if func\_bigger\_p !=-1001

Вх. array[i + 1], p

Вх. array[i + 1], p

Вых. num2

Вх. array[i + 1], c, d

Вх. array[i], num

Вых. число или None

Вых. num2

Вых. true или false

if i < row and поиск элемента равного заданному

for i = 0; i < row – 1; +2

num2 = проверка принадлежности к отрезку

**+**

**+**

**\_**

**\_**

Вывод num2

Выход из цикла

num2 = func\_bigger\_p

**+**

Вывод num2

Выход из цикла

**\_**

* **Подпрограмма ввода данных (read\_data\_from\_file):**

Ввод row, col, a, num, c, d, p

Открытие файла

Чтение командной строки

Возврат a, num, c, d, p, row, col

* **Подпрограмма вывода матрицы (matrix\_output):**

for j = 0; i < row; +1

**\_**

**\_**

for i = 0; i < row; +1

Ввод a[i][j]

**+**

**+**

* **Подпрограмма нахождения числа в четной строке матрицы (even\_row\_in\_matrix):**

check = False

**\_**

for i = 0; i < row; +1

line[i] = n

**+**

**+**

**\_**

check = False

check = True

Выход из цикла

* **Подпрограмма нахождения минимального элемента, принадлежащего отрезку [C, D] (min\_in\_the\_segment):**

**+**

**+**

for i = 0; i < row; +1

if line[i] < min and c ≤ line[i] ≤ d

num = min = line[i]

**\_**

**\_**

num = -1001; min = d + 1

* **Подпрограмма нахождения числа, большего, чем P (element\_position\_bigger\_than\_p):**

**\_**

if line[i] > p

**+**

**+**

**\_**

Возврат i

for i = 0; i < row; +1

**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 = [0] \* col  
 a[len(a) - 1] = [0] \* col  
  
 for i in range(row):  
 a[i] = [0] \* col  
  
 for i in range(row): # Ввод массива  
 line = file.readline().split()  
 for j in range(col):  
 a[i][j] = int(line[j])  
  
 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 i in range(row):  
 for j in range(col):  
 print("%3d" % a[i][j], end=' ')  
 print()  
  
  
def even\_row\_in\_matrix(line, n): # Подпрограмма нахождения элемента равного num  
 check = False  
 for i in range(row):  
 if line[i] == n:  
 check = True  
 break  
 else:  
 check = False  
 return check  
  
  
def min\_in\_the\_segment(line, c, d): # Подпрограмма проверки условия [C, D]  
 num = -1001  
 min = d + 1  
 for i in range(row):  
 if line[i] < min and c <= line[i] <= d:  
 num = min = line[i]  
 return num  
  
  
def element\_position\_bigger\_than\_p(line, p): # Подпрограмма нахождения числа большего P  
 check = False  
 for i in range(row):  
 if line[i] > p:  
 check = True  
 break  
 if check:  
 return i  
  
  
array, num, c, d, p, row, col = read\_data\_from\_file() # Присваивание констант  
  
print('Веденная матрица:')  
matrix\_output(array) # Вывод матрицы  
  
check = False  
  
for i in range(0, row - 1, 2): # Основной цикл  
 func\_bigger\_p = element\_position\_bigger\_than\_p(array[i + 1], p)  
 if even\_row\_in\_matrix(array[i], num) and i < row: # Условие проверки выполнения первого условия  
 num2 = min\_in\_the\_segment(array[i + 1], c, d) # Присваивание значения подпрограммы [C, D]  
 print("Элемент, значение которого попадает в отрезок [{}, {}], равен: {}\n".format(c, d, num2)) # Вывод выходных данных  
 check = True  
 break # Выход из цикла  
 elif func\_bigger\_p != -1001: # Если условие не выполняется  
 num2 = func\_bigger\_p # Присваивания значения подпрограммы > P  
 print("Позиция элемента большего, чем P, равна: {}\n".format(num2 + (i + 1) \* col + 1)) # Вывод выходных данных  
 check = True  
 break # Выход из цикла  
 else:  
 check = False  
  
if not check:  
 print("Ни одно из условий ни разу не выполнилось!\n")