**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] |
| flg – логическая переменная |  | Простая переменная | Истина или ложь |
| Выходные данные | num2 – конечное число | Целый | Простая переменная | [-1000, 1000] |

1. **Блок-схемы:**

* **Головной модуль:**

Чтение переменных с файла

Вывод матрицы

Вых. array, num, c, d, p, row, col

i = i + 1

check = True

Вывод num2

flg

Вых. flg, num2

Вх. array[i], P

Нахождение индекса первого элемента, значение которого больше p.

Вх. array[i], C, D

check = True

Вывод num2

flg

Вых. num2

Нахождение значения минимального элемента в нечетной строке, принадлежащего [C, D]

**\_**

**+**

**\_**

**+**

Вх. array[i], num

Вых. flg

Проверка на наличие заданного числа в четной строке

!check and i < row - 2

check = False; i = 0

Вх. array

* **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):**

flg = False; i = 0

!flg

**\_**

line[i] = n

**+**

**+**

flg = True

i = i + 1

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

**+**

for i = 0; i < row; +1

**\_**

min = line[i]

**+**

if line[i] < min

**\_**

**+**

min = line[i]; flg = True

**+**

**\_**

!flag

if c ≤ line[i] ≤ d

min = line[0]; flg = False

**\_**

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

i = i + 1

**\_**

**+**

**\_**

**+**

flg = True

line[i] > p

!flg

flg = False; i = 0

**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 |
| i = 0 |
| while not check: |
| if line[i] == n: |
| check = True |
| i += 1 |
| return check |
|  |
|  |
| def min\_in\_the\_segment(line, c, d): # Подпрограмма проверки условия [C, D] |
| check = False |
| min = line[0] |
| for i in range(row): |
| if c <= line[i] <= d: |
| if not check: |
| min = line[i] |
| flg = True |
| elif line[i] < min: |
| min = line[i] |
| return check, min |
|  |
|  |
| def element\_position\_bigger\_than\_p(line, p): # Подпрограмма нахождения числа большего P |
| check = False |
| i = 0 |
| while not check: |
| if line[i] > p: |
| check = True |
| i += 1 |
|  |
| return check, i |
|  |
|  |
| array, num, c, d, p, row, col = read\_data\_from\_file() # Присваивание констант |
|  |
| print('Веденная матрица:') |
| matrix\_output(array) # Вывод матрицы |
|  |
| check = False |
| i = 0 |
|  |
| while not check and i < row - 2: |
| flg\_p, func\_bigger\_p = element\_position\_bigger\_than\_p(array[i + 1], p) |
| flg\_seg, func\_min\_in\_seg = min\_in\_the\_segment(array[i + 1], c, d) # Присваивание значения подпрограммы [C, D] |
| if even\_row\_in\_matrix(array[i], num) and flg\_seg: # 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 |
| elif flg: |
| num2 = func\_bigger\_p |
| print("Позиция элемента большего, чем P, равна: {}\n".format(num2 + (i + 1) \* col + 1)) # Вывод выходных данных |
| check = True |
| i += 2 |
|  |
| if not check: |
| print("Ни одно из условий ни разу не выполнилось!\n") |