**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 программы:**

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. **Таблица данных:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Класс | Имя, смысл | Тип | Структура | Диапазон | Формат |
| Входные данные | n, m – количество строк и столбцов | Целый | Простые переменные |  |  |
| 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 |  |  |
| str – место выхода функции – условия | Строка | Простая строка |
| row – строка в массиве | Целый | Одномерный массив |
| i – счетчик цикла | Целый | Простая переменная |
| Выходные данные | num – конечное число | Целый | Простая переменная |  |  |

1. **Блок-схемы:**

* **Головной модуль:**

matrix\_output()

read\_data\_from\_file()

**\_**

**\_**

**\_**

**\_**

**+**

**+**

**+**

**+**

Вывод num

str = ‘pos’

Вывод num

str = ‘pos’

Выход из цикла

num, str = min\_in\_the\_segment(array[i], C, D, P)

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

for i, row in enumerate(array)

Str = ‘’, num = 0

* **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] и нахождения первого элемента большего P.**

Возврат num, ‘num’

Возврат pos, ‘pos’

**\_**

**+**

num is None

**\_**

**+**

**\_**

**\_**

**+**

**+**

num = min = elem

elem < min and c ≤ min ≤ d

pos = i

elem > p and pos = -1

num = None; min = d + 1; pos = -1

for i, elem in enumerate(row)

1. **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 | Позиция элемента большего, чем P, равна: 6 | Проверка невыполнения первого условия на одинаковых данных |

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))  
  
 # row, col = 4, 5  
 # a = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15], [16, 17, 18, 19, 20]]  
 # num = 5  
 # c, d = 10, 11  
 # p = 6  
  
 return a, num, c, d, p  
  
  
def matrix\_output(a):  
 for row in a:  
 for col in row:  
 print("%3d" % col, end=' ')  
 print()  
  
  
def even\_row\_in\_matrix(row, n):  
 for elem in row:  
 if elem == n:  
 return True  
 return False  
  
  
def min\_in\_the\_segment(row, c, d, p):  
 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):  
 for i, elem in enumerate(row):  
 if elem > p:  
 return i  
  
  
array, NUM, C, D, P = read\_data\_from\_file()  
  
print('Веденная матрица:')  
matrix\_output(array)  
  
str = ''  
num = 0  
  
for i, row in enumerate(array):  
 if i % 2 == 0 and even\_row\_in\_matrix(row, NUM) and i < len(array) - 1:  
 print("Элемент, значение которого попадает в отрезок [{}, {}], равен: {}\n".format(C, D, min\_in\_the\_segment(array[i + 1], C, D, P)))  
 break  
 elif element\_position\_bigger\_than\_p(array[i + 1], P) is not None:  
 print("Позиция элемента большего, чем P, равна: {}\n".format(element\_position\_bigger\_than\_p(array[i + 1], P) + (i + 1) \* 5 + 1))  
 break