

# Отчет по лабораторной работе № 2 по курсу “Фундаментальная информатика”

Студент группы М80-101Б-21 Ершова Станислава Григорьевича, № по списку 8

Контакты e-mail, telegram: stas.ershov57@gmail.com ,  
@stas\_orel

Работа выполнена: «5» мая 2022г.

Преподаватель: каф. 806 Севастьянов Виктор Сергеевич

Отчет сдан «5» мая 2022 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

**1. Тема: Разреженные матрицы**

**2. Цель работы:**

**3. Задание (вариант № 7):** Найти строку с минимальным числом 0-ых элементов, вывести её номер и сумму элементов. Если таких строк несколько, то обработать все.

**4. Оборудование (студента):**

Процессор *Intel Pentium N4200 1.1 ГГц* с ОП 8 Гб, SSD 128 Гб. Монитор *1920x1080*

**5. Программное обеспечение (студента):**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *20.04.3 LTS*

интерпретатор команд: *bash* версия *5.0.17*

Система программирования -- версия --, редактор текстов *nano* версия *4.8*

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

**6. Идея, метод, алгоритм:**

**7. Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

MATRIX.H:

```
#ifndef MATRIX_H
#define MATRIX_H
```

```
#include "stdbool.h"
```

```
typedef struct Matrix {
    int CIP[100];
    int PI[100];
    int YE[100];
} Matrix;
```

```
void create_matrix(Matrix * m);
void input_matrix(Matrix * mat, int* n, int *m);
void print_matrix(Matrix *mat, int n, int m);
void function_matrix(Matrix * mat, int n, int m);
```

```
#endif
```

MATRIX.C:

```
#include "stdlib.h"
#include "stdio.h"
#include "math.h"
```

```
#include "matrix.h"
```

```
void create_matrix(Matrix* m) {
    m = malloc(sizeof(Matrix));
}
```

```
void input_matrix(Matrix * mat, int *n, int *m) {
    FILE * input;
    input = fopen("input.mai", "r");
    if (input == NULL) {
        printf("Невозможно открыть файл\n");
        exit(1);
    }
```

```
    fscanf(input, "%d %d", n, m);
    int num;
    int cnt = 0;
```

```
    for (int i = 0; i != *n; ++i) {
        int flag = 0;
        for (int j = 0; j != *m; ++j) {
            fscanf(input, "%d", &num);
            if (num != 0) {
                if (flag == 0) {
                    flag = 1;
                    mat->CIP[i] = j + 1;
                }
                mat->PI[cnt] = j + 1;
                mat->YE[cnt] = num;
                cnt++;
            }
        }
        if (flag == 0)
            mat->CIP[i] = 0;
    }
}
```

```
    mat->PI[cnt] = 0;
    fclose(input);
}
```

```
void print_matrix(Matrix *mat, int n, int m) {
    int cnt = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            if (mat->PI[cnt] == j + 1) {
                int flag = 0;
                if (j + 1 == m) {
                    flag = 1;
                } else if (mat->PI[cnt] == mat->CIP[i + 1]) {
                    int k = cnt + 1;
                    int prevPI = mat->PI[cnt];
                    while (mat->PI[k] != 0) {
                        if (mat->PI[k] == mat->PI[cnt]) {
                            flag = 1;
                            break;
                        } else if (mat->PI[k] < prevPI) {
                            break;
                        }
                    }
                    k++;
                }

                } else
                    flag = 1;
                if (flag > 0) {
                    printf("%d ", mat->YE[cnt]);
                    cnt++;
                } else
                    printf("0 ");
            }
        } else {
            printf("0 ");
        }
    }
}
```

```

    }
    printf("\n");
}
}

```

```

void function_matrix(Matrix * mat, int n, int m) {
    int n_nums[100];
    int n_cnt = 0;
    int max_len = 0;
    int n_sum[100];
    int cnt = 0;
    for (int i = 0; i < n; ++i) {
        int local_len = 0;
        int local_sum = 0;
        for (int j = 0; j < m; ++j) {
            if(mat->PI[cnt] == j + 1) {
                int flag = 0;
                if (j + 1 == m) {
                    flag = 1;
                } else if (mat->PI[cnt] == mat->CIP[i + 1]) {
                    int k = cnt + 1;
                    int prevPI = mat->PI[cnt];
                    while(mat->PI[k] != 0) {
                        if (mat->PI[k] == mat->PI[cnt]) {
                            flag = 1;
                            break;
                        } else if (mat->PI[k] < prevPI) {
                            break;
                        }
                        k++;
                    }
                } else
                    flag = 1;
                if (flag > 0) {
                    local_len++;
                    local_sum += mat->YE[cnt];
                    cnt++;
                }
            }
        }
        if (local_len > max_len) {
            n_cnt = 1;
            max_len = local_len;
            n_sum[n_cnt - 1] = local_sum;
            n_nums[n_cnt - 1] = i + 1;
        } else if (local_len == max_len) {
            n_cnt++;
            n_sum[n_cnt - 1] = local_sum;
            n_nums[n_cnt - 1] = i + 1;
        }
    }

    printf("Количество строк с минимальным числом 0-ых элементов: %d\n", n_cnt);
    for (int i = 0; i < n_cnt; ++i)
        printf("Строка №%d: сумма = %d\n", n_nums[i], n_sum[i]);
}

```

MAIN.C:

```
#include <stdio.h>
```

```
#include "matrix.h"
```

```

int main() {
    int n;
    int m;
    Matrix* mat;
    create_matrix(mat);

    input_matrix(mat, &n, &m);
    printf("%d %d\n", n, m);
    print_matrix(mat, n, m);
}

```

```
function_matrix(mat, n, m);  
}
```

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

**10. Замечания автора** по существу работы

**11. Выводы:** Прокачал свои умственные способности, подумал.

Подпись студента

---