

**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»  
(НГТУ)**

Институт радиоэлектроники и информационных технологий  
Кафедра «Информационные радиосистемы»

**Контрольная работа по дисциплине  
«Информационные технологии»**

Направление подготовки: 11.03.01 Радиотехника  
код и наименование направления подготовки

Выполнил:

**Студент гр. 24-Рз** **Иванов И.И.**  
(группа) (подпись)

Проверил:

доцент кафедры ИРС \_\_\_\_\_ Балашова Д.М.  
(подпись)

Оценка: \_\_\_\_\_

Дата: «\_\_» \_\_\_\_\_ 2024 г.

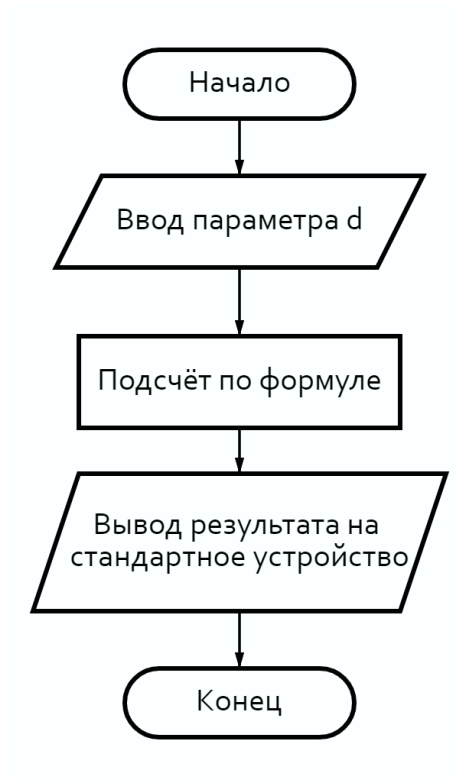
Нижний Новгород, 2024

## ЗАДАНИЕ 1

### ВАРИАНТ 12

Преобразовать значение длины из дюймов в метры по формуле  $m=0.0254*d$ . Параметр  $d$  должен вводиться с клавиатуры.

#### Блок-схема алгоритма



#### Листинг программного кода

```
#include <stdio.h>

int main(){

    int d = 0;
    long double m = 0.0;

    printf("Input an inch length to convert into meters:\n");
    scanf("%d", &d);

    m = 0.0254 * d;

    printf("Result is: %Lf\n", m);

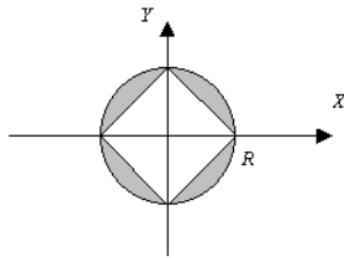
    return 0;

}
```

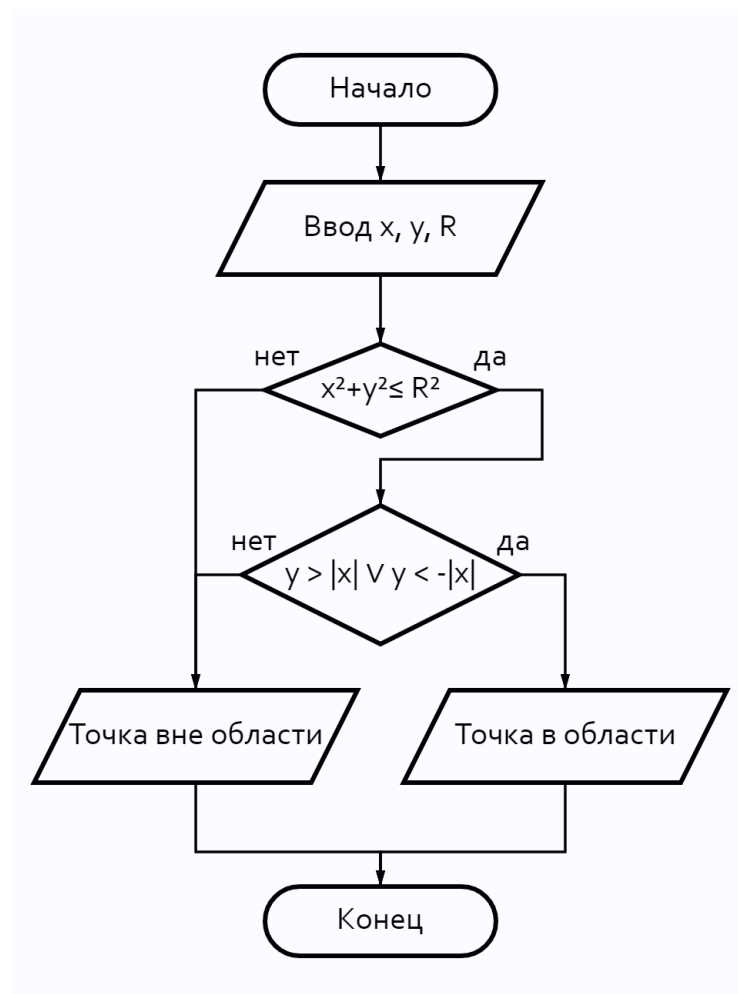
## ЗАДАНИЕ 2. ЧАСТЬ 1

### ВАРИАНТ 24

Проверить попала ли точка с координатами  $P(x, y)$  в заштрихованную область:



### Блок-схема алгоритма



### Листинг программного кода

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

int main() {

    double x, y, R;
```

```
printf("Enter the coords for the dot (x, y): ");
scanf("%lf %lf", &x, &y);
printf("Enter the radius R: ");
scanf("%lf", &R);

if (x * x + y * y <= R * R) {
    if ((y > fabs(x)) || (y < -fabs(x))) {
        printf("Dot (%.2lf, %.2lf) is inside the region.\n", x, y);
    }
    else {
        printf("Dot (%.2lf, %.2lf) is outside the region.\n", x, y);
    }
}
else {
    printf("Dot (%.2lf, %.2lf) is outside the region.\n", x, y);
}

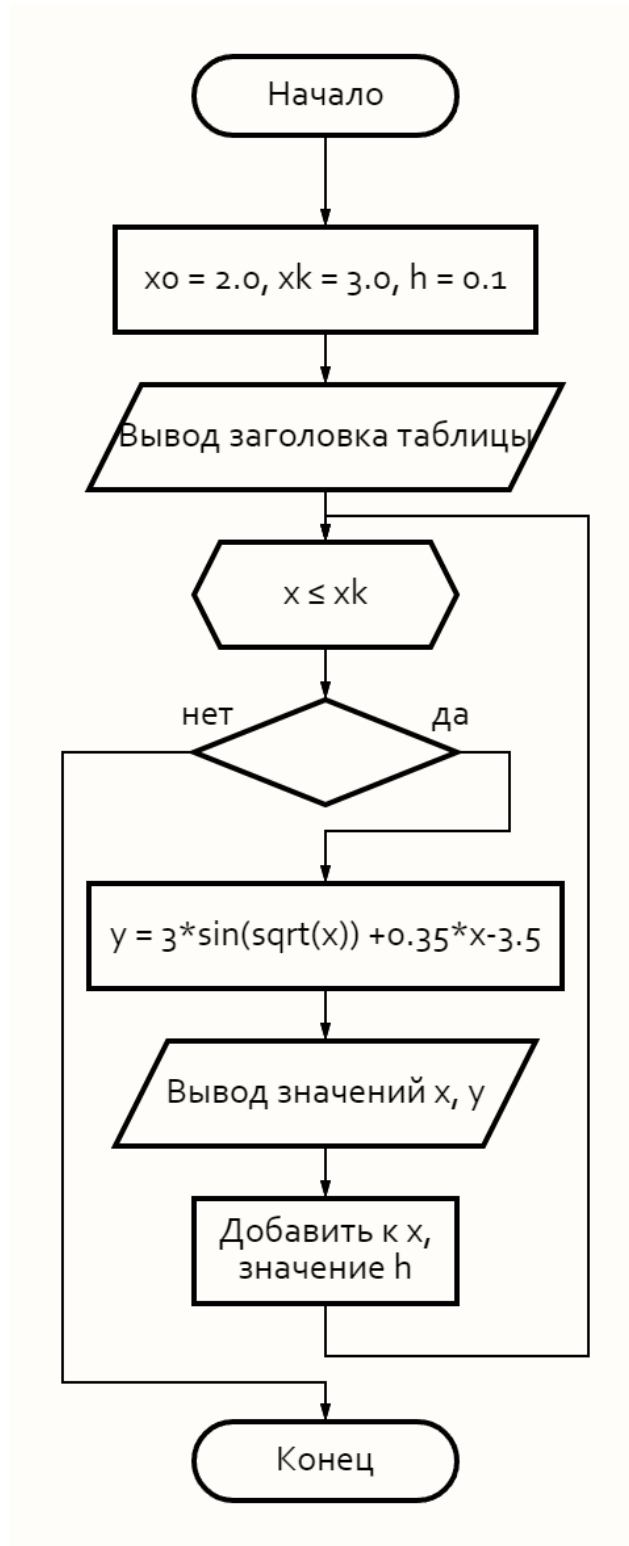
return 0;
}
```

## ЗАДАНИЕ 2. ЧАСТЬ 2

### ВАРИАНТ 1

Построить таблицы функции  $y = 3 * \sin(\sqrt{x}) + 0.35 * x - 3.8$  Пусть  $x_0 = 2$  (начальное значение);  $x_k = 3$  (конечное значение);  $h = 0.1$  (шаг изменения  $x$ ).

#### Блок-схема алгоритма



### Листинг программного кода

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

int main() {

    double x0 = 2.0, xk = 3.0, h = 0.1;
    double x, y;

    printf("x\t  y\n");
    printf("-----\n");

    for (x = x0; x <= xk; x += h) {
        y = 3 * sin(sqrt(x)) + 0.35 * x - 3.8;
        printf("%.2f\t %.4f\n", x, y);
    }

    return 0;
}
```

### ЗАДАНИЕ 3

#### ВАРИАНТ 6

$$\text{Формула: } r = \frac{\left(\sum_{i=1}^n x_i\right)^2 - \frac{1}{n} \cdot \sum_{i=3}^n x_i}{\sqrt{(n-2) \cdot \sum_{i=4}^n x_i}}, \quad n=8;$$

$$\text{функция: } x_i = b \cdot \frac{\lg(7.2 + i)}{2} + \sqrt{i + a + b},$$

где  $a=4, b=2$ .

Расчет сумм в формуле и расчет  $x_i$  должны быть оформлены в виде отдельных функций.

#### Листинг программного кода

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

double sum_squares(int n) {
    double sum = 0.0;
    for (int i = 1; i <= n; i++) {
        sum += i * i;
    }
    return sum;
}

double sum_xi(int n, double *x) {
    double sum = 0.0;
    for (int i = 3; i <= n; i++) {
        sum += x[i - 1];
    }
    return sum;
}

double calculate_xi(int i, int a, int b) {
    return b * log10(7.2 + i) / 2.0 + sqrt(i + a + b);
}

int main() {

    int n = 8;
    int a = 4, b = 2;
    double sum_sq = sum_squares(n);
    double x[n];
    for (int i = 1; i <= n; i++) {
        x[i - 1] = calculate_xi(i, a, b);
    }
    double sum_x = sum_xi(n, x);
    double r = (sum_sq - sum_x) / sqrt((n - 2) * sum_x);
    printf("r = %.2f\n", r);
    return 0;
}
```

## ЗАДАНИЕ 4 - 1

### ВАРИАНТ 4

Удалить из строки s1 все символы, входящие в строку s2. Строки должны вводиться с клавиатуры.

#### Листинг программного кода

Файл 4a-header.h

```
#ifndef STRING_UTILS_H
#define STRING_UTILS_H

#include <stdio.h>
#include <string.h>

void do_input(char s1[], char s2[]);
void do_process(char s1[], char s2[]);
void do_output(char s1[]);

#endif // STRING_UTILS_H
```

Файл 4a-main.c

```
#include "4a-header.h"

int main() {

    char s1[100], s2[100];
    do_input(s1, s2);
    do_process(s1, s2);
    do_output(s1);

    return 0;
}
```

Файл 4a-input.c

```
#include "4a-header.h"

void do_input(char s1[], char s2[]) {

    printf("Enter the first string (s1): ");
    fgets(s1, 100, stdin);
    s1[strcspn(s1, "\n")] = '\0';

    printf("Enter the second string (s2): ");
    fgets(s2, 100, stdin);
    s2[strcspn(s2, "\n")] = '\0';

}
```



Файл 4a-process.c

```
#include "4a-header.h"

void do_process(char s1[], char s2[]) {

    int char_map[256] = {0};
    for (int i = 0; s2[i] != '\0'; i++) {
        char_map[(unsigned char)s2[i]] = 1;
    }

    int new_index = 0;
    for (int i = 0; s1[i] != '\0'; i++) {
        if (!char_map[(unsigned char)s1[i]]) {
            s1[new_index++] = s1[i];
        }
    }
    s1[new_index] = '\0';
}
```

Файл 4a-output.c

```
#include <stdio.h>

void do_output(char s1[]) {

    printf("Processed string (s1): %s\n", s1);
}
```

## ЗАДАНИЕ 4 - 2

### ВАРИАНТ 4

Удалить из строки s1 все символы, входящие в строку s2. Строки должны вводиться с клавиатуры.

#### Листинг программного кода

Файл 4p-header.h

```
#ifndef STRING_UTILS_H
#define STRING_UTILS_H

#include <stdio.h>
#include <string.h>

void do_input(char *source_string, char *remove_string);
void do_process(char *source_string, char *remove_string);
void do_output(char *source_string);

#endif // STRING_UTILS_H
```

Файл 4p-main.c

```
#include "4p-header.h"

int main() {

    char source_string[100], remove_string[100];

    do_input(source_string, remove_string);
    do_process(source_string, remove_string);
    do_output(source_string);

    return 0;
}
```

Файл 4p-input.c

```
#include "4p-header.h"

void do_input(char *source_string, char *remove_string) {

    printf("Enter the source string: ");
    fgets(source_string, 100, stdin);
    source_string[strcspn(source_string, "\n")] = '\0';

    printf("Enter the string with characters to remove: ");
    fgets(remove_string, 100, stdin);
    remove_string[strcspn(remove_string, "\n")] = '\0';

}
```

Файл 4p-process.c

```
#include "4p-header.h"

void do_process(char *source_string, char *remove_string) {

    int character_flags[256] = {0};
    char *read_ptr = remove_string;

    while (*read_ptr) {
        character_flags[(unsigned char)*read_ptr] = 1;
        read_ptr++;
    }

    char *read_source = source_string;
    char *write_source = source_string;

    while (*read_source) {
        if (!character_flags[(unsigned char)*read_source]) {
            *write_source = *read_source;
            write_source++;
        }
        read_source++;
    }
    *write_source = '\0';
}
```

Файл 4p-output.c

```
#include "4p-header.h"

void do_output(char *source_string) {

    printf("Processed string: %s\n", source_string);

}
```

## ЗАДАНИЕ 5 - 1

### ВАРИАНТ 5

В таблице собраны сведения об успеваемости студентов одной группы по некоторой дисциплине. Информация об отдельном студенте включает в себя ФИО студента и отметки (сдано или не сдано) по каждой из 4 лабораторных работ. Выдать список студентов, имеющих задолженности, и подсчитать общее число таких студентов.

#### Листинг программного кода

Файл 5a-header.h

```
#ifndef STUDENT_H
#define STUDENT_H

#include <stdio.h>
#include <string.h>

#define MAX_STUDENTS 100
#define LABS 4

typedef struct {
    char name[50];
    int marks[LABS];
} Student;

void do_input(Student students[], int *totalStudents);
void do_process(Student students[], int totalStudents, int
students_with_debts[], int *debtCount);
void do_output(Student students[], int students_with_debts[], int
debtCount);

#endif
```

Файл 5a-main.c

```
#include "5a-header.h"

int main() {

    int totalStudents;
    Student students[MAX_STUDENTS];
    int students_with_debts[MAX_STUDENTS];
    int debtCount = 0;

    do_input(students, &totalStudents);
    do_process(students, totalStudents, students_with_debts, &debtCount);
    do_output(students, students_with_debts, debtCount);

    return 0;
}
```

Файл 5a-input.c

```
#include "5a-header.h"

void do_input(Student students[], int *totalStudents) {
    printf("Enter the total number of students: ");
    scanf("%d", totalStudents);
    getchar();

    for (int i = 0; i < *totalStudents; i++) {
        printf("Enter name for student %d: ", i + 1);
        fgets(students[i].name, sizeof(students[i].name), stdin);
        students[i].name[strcspn(students[i].name, "\n")] = '\0';

        printf("Enter marks for LAB 4 (0 = fail, 1 = pass) for %s:\n",
students[i].name);
        for (int j = 0; j < LABS; j++) {
            printf("Lab %d: ", j + 1);
            scanf("%d", &students[i].marks[j]);
        }
        getchar();
    }
}
```

Файл 5a-process.c

```
#include "5a-header.h"

void do_process(Student students[], int totalStudents, int
students_with_debts[], int *debtCount) {

    *debtCount = 0;

    for (int i = 0; i < totalStudents; i++) {

        int hasDebt = 0;

        for (int j = 0; j < LABS; j++) {
            if (students[i].marks[j] == 0) {
                hasDebt = 1;
                break;
            }
        }
        if (hasDebt) {
            students_with_debts[*debtCount] = i;
            (*debtCount)++;
        }
    }
}
```

```
#include "5a-header.h"
```

```
void do_output(Student students[], int students_with_debts[], int  
debtCount) {
```

```
    if (debtCount > 0) {  
        printf("\nStudents with debts:\n");  
        for (int i = 0; i < debtCount; i++) {  
            printf("%s\n", students[students_with_debts[i]].name);  
        }  
        printf("\nTotal number of students with debts: %d\n", debtCount);  
    } else {  
        printf("\nNo students have debts.\n");  
    }  
}
```

```
}
```

## ЗАДАНИЕ 5 - 2

### ВАРИАНТ 5

В таблице собраны сведения об успеваемости студентов одной группы по некоторой дисциплине. Информация об отдельном студенте включает в себя ФИО студента и отметки (сдано или не сдано) по каждой из 4 лабораторных работ. Выдать список студентов, имеющих задолженности, и подсчитать общее число таких студентов.

#### Листинг программного кода

Файл 5p-header.h

```
#ifndef STUDENT_H
#define STUDENT_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define LABS 4

typedef struct {
    char name[50];
    int marks[LABS];
} Student;

void do_input(Student *students, int totalStudents);
void do_process(Student *students, int totalStudents, int
*students_with_debts, int *debtCount);
void do_output(Student *students, int *students_with_debts, int debtCount);

#endif
```

Файл 5p-main.c

```
#include "5p-header.h"

int main() {

    int totalStudents;
    int debtCount = 0;
    printf("Enter the total number of students: ");
    scanf("%d", &totalStudents);
    getchar();

    Student *students = (Student *)malloc(totalStudents * sizeof(Student));
    int *students_with_debts = (int *)malloc(totalStudents * sizeof(int));

    if (students == NULL || students_with_debts == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }
}
```

```

do_input(students, totalStudents);
do_process(students, totalStudents, students_with_debts, &debtCount);
do_output(students, students_with_debts, debtCount);

free(students);
free(students_with_debts);

return 0;
}

```

#### Файл 5p-input.c

```

#include "5p-header.h"

void do_input(Student *students, int totalStudents) {
    for (int i = 0; i < totalStudents; i++) {
        printf("Enter name for student %d: ", i + 1);
        fgets(students[i].name, sizeof(students[i].name), stdin);
        students[i].name[strcspn(students[i].name, "\n")] = '\0';

        printf("Enter marks for LAB 4 (0 = fail, 1 = pass) for %s:\n",
students[i].name);
        for (int j = 0; j < LABS; j++) {
            printf("Lab %d: ", j + 1);
            scanf("%d", &students[i].marks[j]);
        }
        getchar();
    }
}

```

#### Файл 5p-process.c

```

#include "5p-header.h"

void do_process(Student *students, int totalStudents, int
*students_with_debts, int *debtCount) {
    *debtCount = 0;
    for (int i = 0; i < totalStudents; i++) {
        int hasDebt = 0;
        for (int j = 0; j < LABS; j++) {
            if (students[i].marks[j] == 0) {
                hasDebt = 1;
                break;
            }
        }
        if (hasDebt) {
            students_with_debts[*debtCount] = i;
            (*debtCount)++;
        }
    }
}

```



```
#include "5p-header.h"

void do_output(Student *students, int *students_with_debts, int debtCount)
{
    if (debtCount > 0) {
        printf("\nStudents with debts:\n");
        for (int i = 0; i < debtCount; i++) {
            printf("%s\n", students[students_with_debts[i]].name);
        }
        printf("\nTotal number of students with debts: %d\n", debtCount);
    } else {
        printf("\nNo students have debts.\n");
    }
}
```