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.

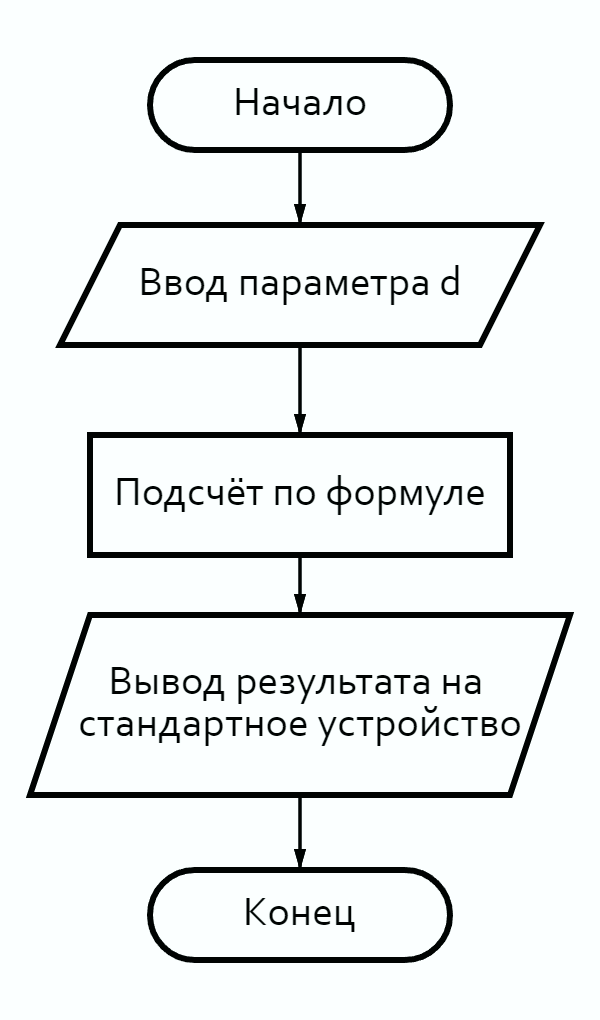
Нижний Новгород, 2024

**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);

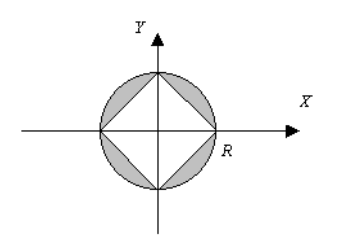
    return 0;

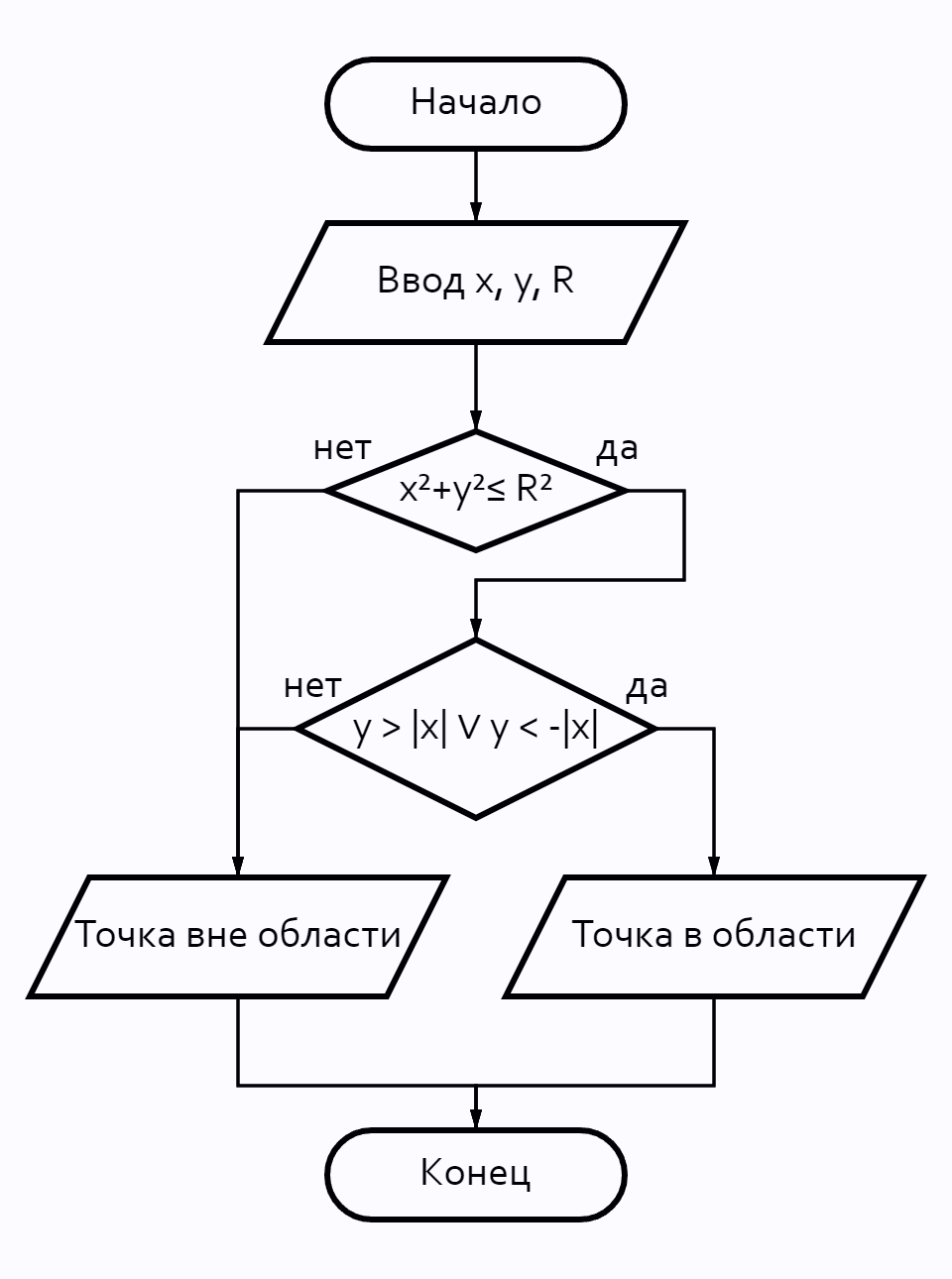
}

**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;

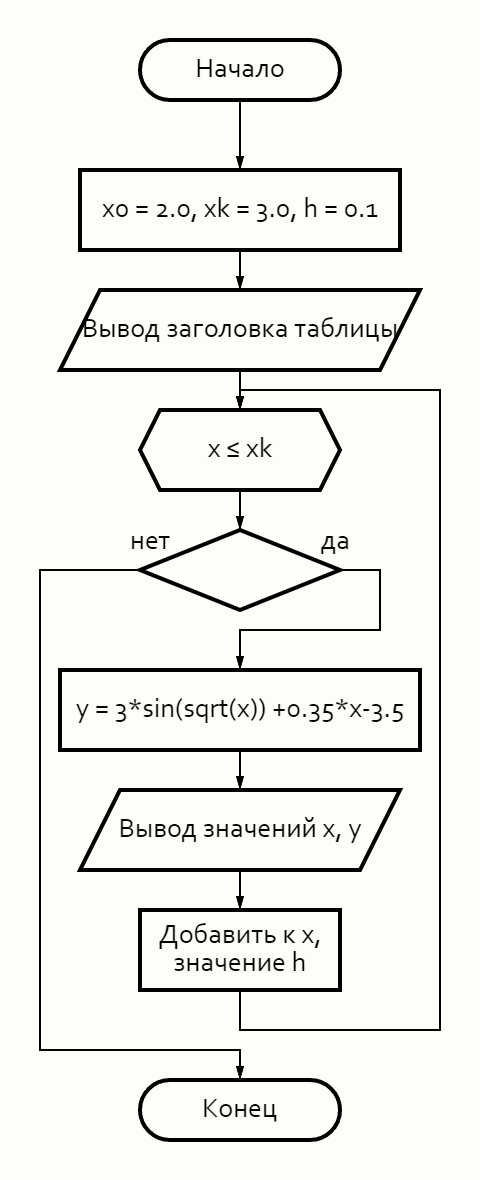
}

**Задание 2. Часть 2**

**Вариант 1**

Построить таблицы функции Пусть x0=2 (началь-ное значение); xk=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;

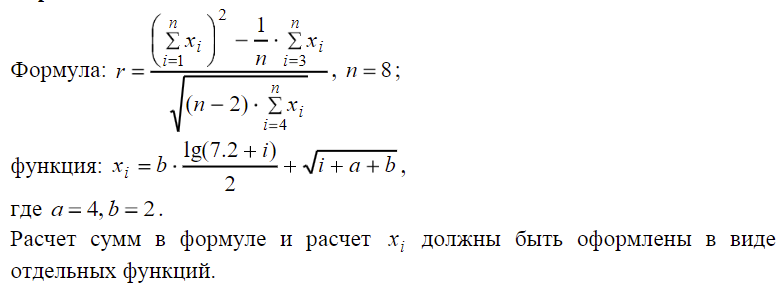
        printf("%.2f\t %.4f\n", x, y);

    }

    return 0;

}

**Задание 3**

**Вариант 6**

**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)++;

        }

    }

}

Файл 5a-output.c

#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)++;

        }

    }

}

Файл 5p-output.c

#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");

    }

}