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 3**

Найти площадь правильного n-угольника, вписанного в окружность радиуса

R по формуле: . Параметры должны вводиться с клавиатуры.

**Блок-схема алгоритма**



**Листинг программного кода**

#include <stdio.h>

#include <math.h>

int main(){

    int n, R;

    long double S;

    printf("Input n: ");

    scanf("%d", &n);

    printf("Input R: ");

    scanf("%d", &R);

    S = (0.5\*n\*R\*R)\*(sin(2\*M\_PI/n));

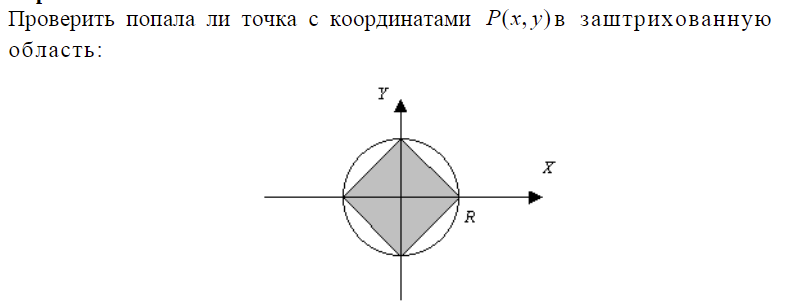
    printf("Result is: %Lf\n", S);

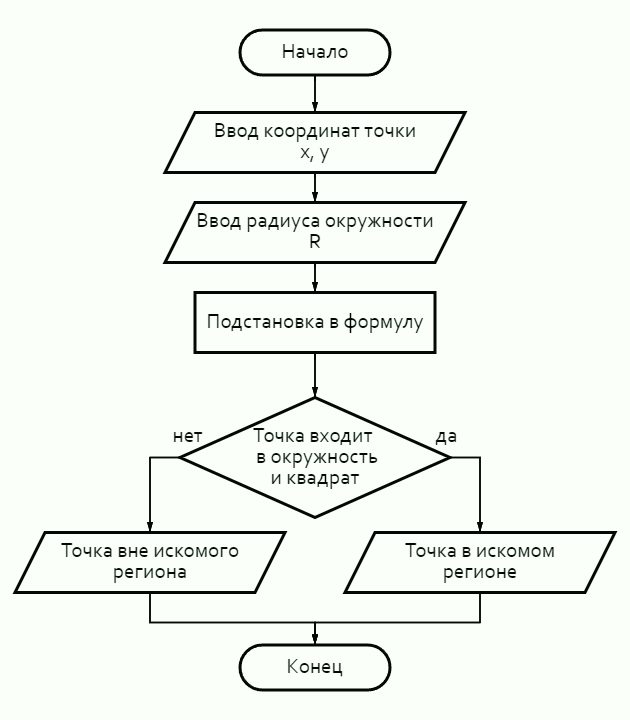
    return 0;

}

**Задание 2. Часть 1**

**Вариант 20**

****

**Блок-схема алгоритма**

**Листинг программного кода**

#include <stdio.h>

#include <math.h>

int main(){

    double x, y, R;

    printf("Enter x/y for the dot (x y): ");

    scanf("%lf %lf", &x, &y);

    printf("Enter raduis - R: ");

    scanf("%lf", &R);

    if ((x \* x + y \* y <= R \* R) && (fabs(x) + fabs(y) <= R)) {

        printf("The point (%.2lf, %.2lf) is inside region.\n", x, y);

    }

    else {

        printf("The point (%.2lf, %.2lf) is outside region.\n", x, y);

    }

    return 0;

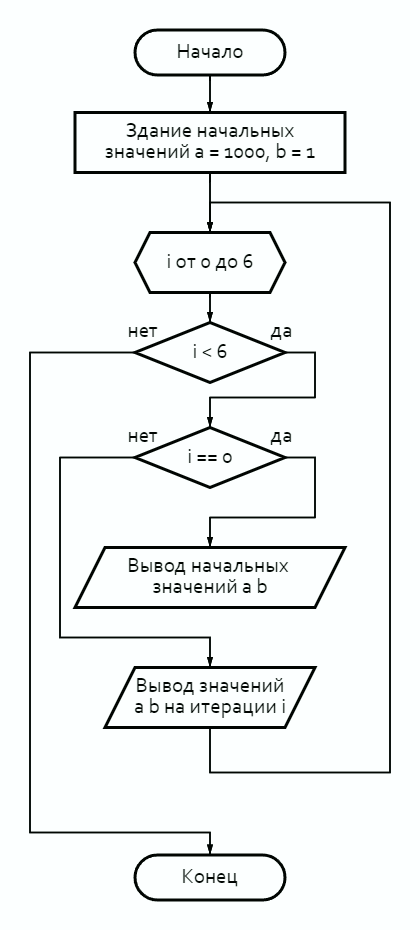
}

**Задание 2. Часть 2**

**Вариант 6**

Вычислить элементы векторов  и , , если , , .

**Блок-схема алгоритма**

****

**Листинг программного кода**

#include <stdio.h>

#include <math.h>

int main(){

    double a = 1000, b = 1;

    double a\_new, b\_new;

    for(int i = 0; i < 6; ++i){

        if (i == 0){

            printf("a = %.2lf, b = %.2lf\n", a, b);

        }

        else{

            a\_new = (a+b)/2;

            b\_new = (sqrt(a\*b));

            a = a\_new, b = b\_new;

            printf("a = %.2lf, b = %.2lf\n", a\_new, b\_new);

        }

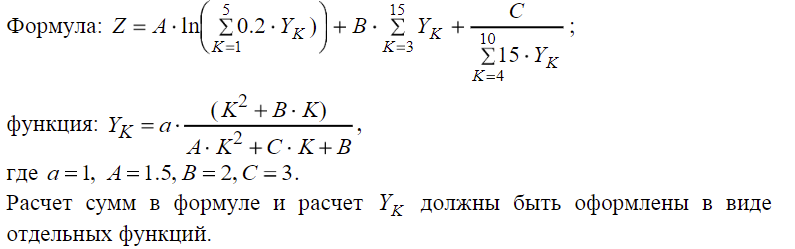
    }

    return 0;

}

**Задание 3**

**Вариант 14**

**Листинг программного кода**

#include <stdio.h>

#include <math.h>

double calculate\_xi(int i, double a, double b) {

    return b \* (log10(7.2 + i) / 2 + sqrt(i + a + b));

}

double calculate\_sum(int start, int end, double a, double b) {

    double sum = 0;

    for (int i = start; i <= end; i++) {

        sum += calculate\_xi(i, a, b);

    }

    return sum;

}

int main() {

    int n = 8;

    double a = 4.0, b = 2.0;

    double sum\_1\_to\_n = calculate\_sum(1, n, a, b);

    double numerator = pow(sum\_1\_to\_n, 2) / n;

    double sum\_3\_to\_n = calculate\_sum(3, n, a, b);

    double second\_term = sum\_3\_to\_n / n;

    double sum\_4\_to\_n = calculate\_sum(4, n, a, b);

    double denominator = sqrt((n - 2) \* sum\_4\_to\_n);

    double r = (numerator - second\_term) / denominator;

    printf("Result (r) = %lf\n", r);

    return 0;

}

**Задание 4 - 1**

**Вариант 9**

Подсчитать количество слов в строке, которая не может начинаться с пробелов, но может заканчиваться ими. Учесть наличие двух и более пробелов между словами (два и более пробелов считать за один).

**Листинг программного кода**

#ifndef WORD\_COUNT\_H

#define WORD\_COUNT\_H

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <stdbool.h>

#define MAX\_LENGTH 1000

void func\_input(char str[MAX\_LENGTH]);

void func\_process(const char str[MAX\_LENGTH], int word\_count[1]);

void func\_output(int word\_count[1]);

#endif

#include "4a.h"

void func\_input(char str[MAX\_LENGTH]) {

    printf("Enter a string: ");

    fgets(str, MAX\_LENGTH, stdin);

    int len = strlen(str);

    if (len > 0 && str[len - 1] == '\n') {

        str[len - 1] = '\0';

    }

}

void func\_process(const char str[MAX\_LENGTH], int word\_count[1]) {

    word\_count[0] = 0;

    bool in\_word = false;

    for (int i = 0; str[i] != '\0'; i++) {

        if (!isspace(str[i])) {

            if (!in\_word) {

                in\_word = true;

                word\_count[0]++;

            }

        }

        else {

            in\_word = false;

        }

    }

}

void func\_output(int word\_count[1]) {

    printf("The number of words is: %d\n", word\_count[0]);

}

int main() {

    char input[MAX\_LENGTH];

    int word\_count[1];

    func\_input(input);

    func\_process(input, word\_count);

    func\_output(word\_count);

    return 0;

}

**Задание 4 - 2**

**Вариант 9**

Подсчитать количество слов в строке, которая не может начинаться с пробелов, но может заканчиваться ими. Учесть наличие двух и более пробелов между словами (два и более пробелов считать за один).

**Листинг программного кода**

#ifndef WORD\_COUNT\_H

#define WORD\_COUNT\_H

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <stdbool.h>

#define MAX\_LENGTH 1000

void func\_input(char \*str);

void func\_process(char \*str, int \*word\_count);

void func\_output(int \*word\_count);

#endif

#include "4p.h"

void func\_input(char \*str) {

    printf("Enter a string: ");

    fgets(str, MAX\_LENGTH, stdin);

    int len = strlen(str);

    if (len > 0 && str[len - 1] == '\n') {

        str[len - 1] = '\0';

    }

}

void func\_process(char \*str, int \*word\_count) {

    \*word\_count = 0;

    bool in\_word = false;

    while (\*str != '\0') {

        if (!isspace(\*str)) {

            if (!in\_word) {

                in\_word = true;

                (\*word\_count)++;

            }

        }

        else {

            in\_word = false;

        }

        str++;

    }

}

void func\_output(int \*word\_count) {

    printf("The number of words is: %d\n", \*word\_count);

}

int main() {

    char input[MAX\_LENGTH];

    int word\_count;

    func\_input(input);

    func\_process(input, &word\_count);

    func\_output(&word\_count);

    return 0;

}

**Задание 5 - 1**

**Вариант 14**

Создать каталог библиотеки из трех книг (по информатике). Он должен содержать шифр книги, УДК, ФИО автора, название книги, год издания, признак наличия книги. Найти книгу с заданным шифром и выдать информацию о наличии данной книги.

**Листинг программного кода**

#ifndef LIBRARY\_CATALOG\_H

#define LIBRARY\_CATALOG\_H

#include <stdio.h>

#include <string.h>

#define SIZE 3

typedef struct {

    char book\_code[10];

    char udk[10];

    char author\_name[50];

    char book\_title[50];

    int publication\_year;

    int is\_available;

} book\_t;

void func\_input(book\_t catalog[], int size);

int func\_process(book\_t catalog[], int size, char search\_code[]);

void func\_output(book\_t book);

#endif

#include "5a.h"

void func\_input(book\_t catalog[], int size) {

    for (int i = 0; i < size; i++) {

        printf("\nBook %d:\n", i + 1);

        printf("Enter book code: ");

        scanf("%s", catalog[i].book\_code);

        printf("Enter UDK: ");

        scanf("%s", catalog[i].udk);

        printf("Enter author's full name: ");

        getchar();

        fgets(catalog[i].author\_name, sizeof(catalog[i].author\_name), stdin);

        catalog[i].author\_name[strcspn(catalog[i].author\_name, "\n")] = '\0';

        printf("Enter book title: ");

        fgets(catalog[i].book\_title, sizeof(catalog[i].book\_title), stdin);

        catalog[i].book\_title[strcspn(catalog[i].book\_title, "\n")] = '\0';

        printf("Enter year of publication: ");

        scanf("%d", &catalog[i].publication\_year);

        printf("Enter availability (1 - available, 0 - not available): ");

        scanf("%d", &catalog[i].is\_available);

    }

}

int func\_process(book\_t catalog[], int size, char search\_code[]) {

    for (int i = 0; i < size; i++) {

        if (strcmp(catalog[i].book\_code, search\_code) == 0) {

            return i;

        }

    }

    return -1;

}

void func\_output(book\_t book) {

    printf("Book Code: %s\n", book.book\_code);

    printf("UDK: %s\n", book.udk);

    printf("Author: %s\n", book.author\_name);

    printf("Title: %s\n", book.book\_title);

    printf("Year: %d\n", book.publication\_year);

    if (book.is\_available == 1) {

        printf("Availability: Available\n");

    }

    else {

        printf("Availability: Not Available\n");

    }

}

int main() {

    book\_t catalog[SIZE];

    char search\_code[10];

    int found\_index;

    printf("Input book details for %d books:\n", SIZE);

    func\_input(catalog, SIZE);

    printf("\nEnter the book code to search: ");

    scanf("%s", search\_code);

    found\_index = func\_process(catalog, SIZE, search\_code);

    if (found\_index != -1) {

        printf("\nBook found!\n");

        func\_output(catalog[found\_index]);

    }

    else {

        printf("\nBook with code \"%s\" not found in the catalog.\n", search\_code);

    }

    return 0;

}

**Задание 5 - 2**

**Вариант14**

Создать каталог библиотеки из трех книг (по информатике). Он должен содержать шифр книги, УДК, ФИО автора, название книги, год издания, признак наличия книги. Найти книгу с заданным шифром и выдать информацию о наличии данной книги.

**Листинг программного кода**

#ifndef LIBRARY\_CATALOG\_H

#define LIBRARY\_CATALOG\_H

#include <stdio.h>

#include <string.h>

#define SIZE 3

typedef struct {

    char book\_code[10];

    char udk[10];

    char author\_name[50];

    char book\_title[50];

    int publication\_year;

    int is\_available;

} book\_t;

void func\_input(book\_t \*catalog, int size);

void func\_process(book\_t \*catalog, int size, char \*search\_code, book\_t \*\*result);

void func\_output(book\_t \*book);

#endif

#include "5p.h"

void func\_input(book\_t \*catalog, int size) {

    for (int i = 0; i < size; i++) {

        printf("\nBook %d:\n", i + 1);

        printf("Enter book code: ");

        scanf("%s", catalog[i].book\_code);

        printf("Enter UDK: ");

        scanf("%s", catalog[i].udk);

        printf("Enter author's full name: ");

        getchar();

        fgets(catalog[i].author\_name, sizeof(catalog[i].author\_name), stdin);

        catalog[i].author\_name[strcspn(catalog[i].author\_name, "\n")] = '\0';

        printf("Enter book title: ");

        fgets(catalog[i].book\_title, sizeof(catalog[i].book\_title), stdin);

        catalog[i].book\_title[strcspn(catalog[i].book\_title, "\n")] = '\0';

        printf("Enter year of publication: ");

        scanf("%d", &catalog[i].publication\_year);

        printf("Enter availability (1 - available, 0 - not available): ");

        scanf("%d", &catalog[i].is\_available);

    }

}

void func\_process(book\_t \*catalog, int size, char \*search\_code, book\_t \*\*result) {

    \*result = NULL;

    for (int i = 0; i < size; i++) {

        if (strcmp(catalog[i].book\_code, search\_code) == 0) {

            \*result = &catalog[i];

            break;

        }

    }

}

void func\_output(book\_t \*book) {

    printf("Book Code: %s\n", book->book\_code);

    printf("UDK: %s\n", book->udk);

    printf("Author: %s\n", book->author\_name);

    printf("Title: %s\n", book->book\_title);

    printf("Year: %d\n", book->publication\_year);

    if (book->is\_available == 1) {

        printf("Availability: Available\n");

    }

    else {

        printf("Availability: Not Available\n");

    }

}

int main() {

    book\_t catalog[SIZE];

    char search\_code[10];

    book\_t \*found\_book = NULL;

    printf("Input book details for %d books:\n", SIZE);

    func\_input(catalog, SIZE);

    printf("\nEnter the book code to search: ");

    scanf("%s", search\_code);

    func\_process(catalog, SIZE, search\_code, &found\_book);

    if (found\_book != NULL) {

        printf("\nBook found!\n");

        func\_output(found\_book);

    }

    else {

        printf("\nBook with code \"%s\" not found in the catalog.\n", search\_code);

    }

    return 0;

}