

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

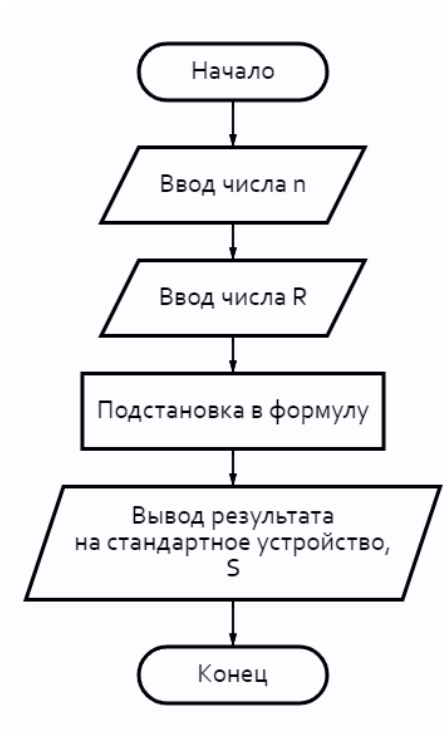
ЗАДАНИЕ 1

ВАРИАНТ 3

Найти площадь правильного n-угольника, вписанного в окружность радиуса

R по формуле: $S = \frac{1}{2} n R^2 \sin\left(\frac{2\pi}{n}\right)$. 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));

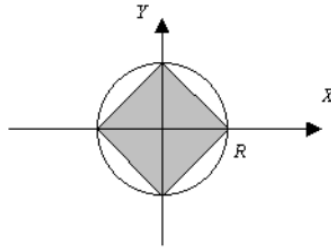
    printf("Result is: %Lf\n", S);

    return 0;
}
```

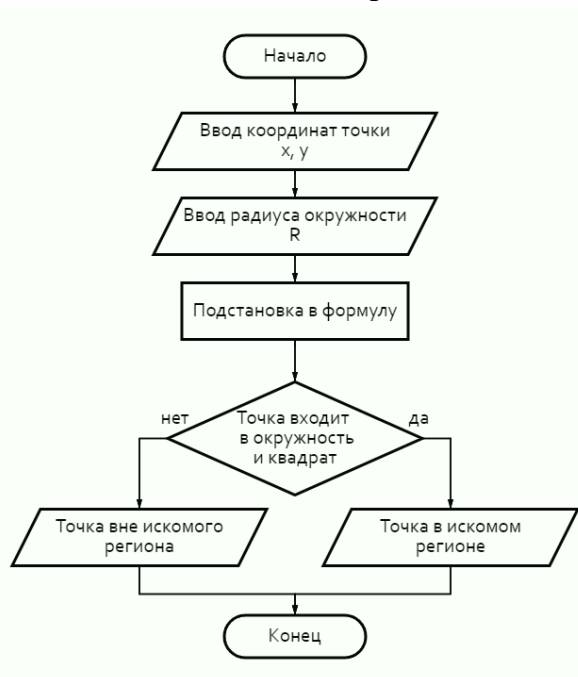
ЗАДАНИЕ 2. ЧАСТЬ 1

ВАРИАНТ 20

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 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;
}
```

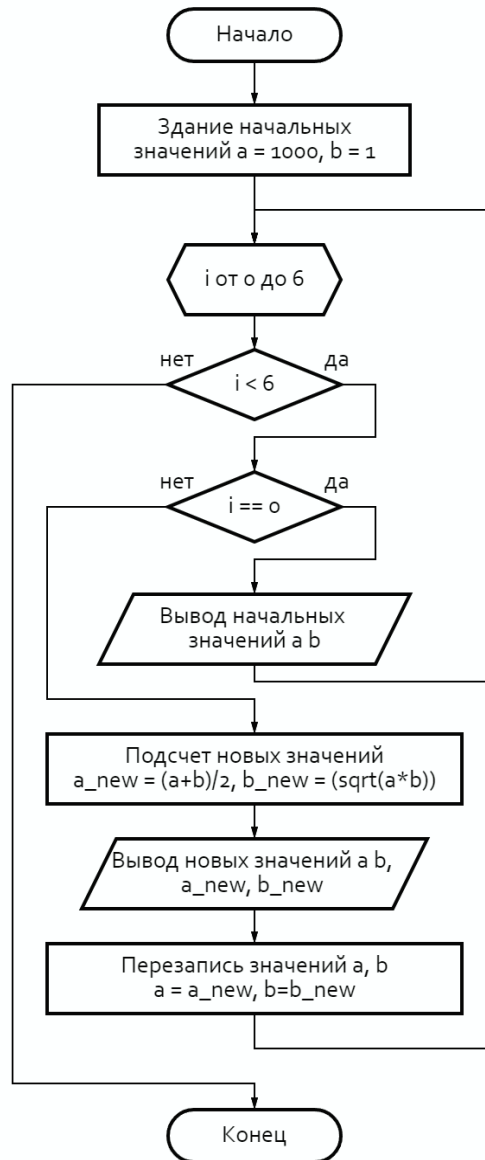
ЗАДАНИЕ 2. ЧАСТЬ 2

ВАРИАНТ 6

Вычислить элементы векторов a_i и b_i , $i=0,1,\dots,6$, если $a_0=1000, b_0=1$ $a_i = \frac{(a_{i-1} + b_{i-1})}{2}$,

$$b_i = \sqrt{a_{i-1} \cdot b_{i-1}}, i=0,1,\dots,6.$$

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

Формула:
$$Z = A \cdot \ln \left(\sum_{K=1}^5 0.2 \cdot Y_K \right) + B \cdot \sum_{K=3}^{15} Y_K + \frac{C}{\sum_{K=4}^{10} 15 \cdot Y_K};$$

функция:
$$Y_K = a \cdot \frac{(K^2 + B \cdot K)}{A \cdot K^2 + C \cdot K + B},$$

где $a=1$, $A=1.5$, $B=2$, $C=3$.

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

Листинг программного кода

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

#define A 1.5
#define B 2
#define C 3
#define a 1

double calculate_y_k(int k) {
    return (a * (k * k + B * k)) / (A * k * k + C * k + B);
}

double calculate_sum1() {
    double sum = 0;
    for (int k = 1; k <= 5; k++) {
        sum += 0.2 * calculate_y_k(k);
    }
    return sum;
}

double calculate_sum2() {
    double sum = 0;
    for (int k = 3; k <= 15; k++) {
        sum += calculate_y_k(k);
    }
    return sum;
}

double calculate_sum3() {
    double sum = 0;
    for (int k = 4; k <= 10; k++) {
        sum += calculate_y_k(k);
    }
    return sum;
}

double calculate_z() {
    double sum1 = calculate_sum1();
```

```
double sum2 = calculate_sum2();
double sum3 = calculate_sum3();

double z = A * log(sum1) + B * sum2 + C / (15 * sum3);
return z;
}

int main() {
    double z = calculate_z();
    printf("Z is: %.6f\n", z);
    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;
}

```