Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 19

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели на функции»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Преподаватель: асс. Андронова М.В.

2023, Минск

4. В соответствии со своим вариантом ***отделить корни*** двух уравнений и вычислить их методом ***дихотомии*** для исходных данных из таблицы, приведенной ниже. Точность вычислений принять равной **e** = 0,001 для всех вариантов.

Операторы метода вычисления корня оформить в виде ***функции пользователя***, уравнения записать также в виде ***функций пользователя***.

В главной функции предусмотреть ввод исходных данных, обращения к функции, реализующей метод дихотомии для двух уравнений. В процессе выполнения программы определить корни двух уравнений. Использовать ***указатель на функцию****.*

|  |  |
| --- | --- |
| 11 | ex + 2x2 – 3,  x3 + 3 |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

#include <iomanip>

using namespace std;

void callfunc(double& a, double& b) { // функция для считывания концов интервала

cout << "Введите a, b" << '\n';

cin >> a >> b;

}

double func2(double x) {

return(x \* x \* x + 3);

}

double func1(double x) {

return(exp(x) + 2 \* x \* x - 3);

}

double dihotomy(double &result, void(\*callfunc)(double&, double&), double(\*func)(double)) { // функция для решения уравнений

double y1, y2;

double a, b;

callfunc(a, b);

while (abs(a - b) > 2 \* 0.001) {

result = (a + b) / 2;

y1 = func(result);

y2 = func(a);

if (y1 \* y2 <= 0) {

b = result;

}

else {

a = result;

}

}

return result;

}

int main() {

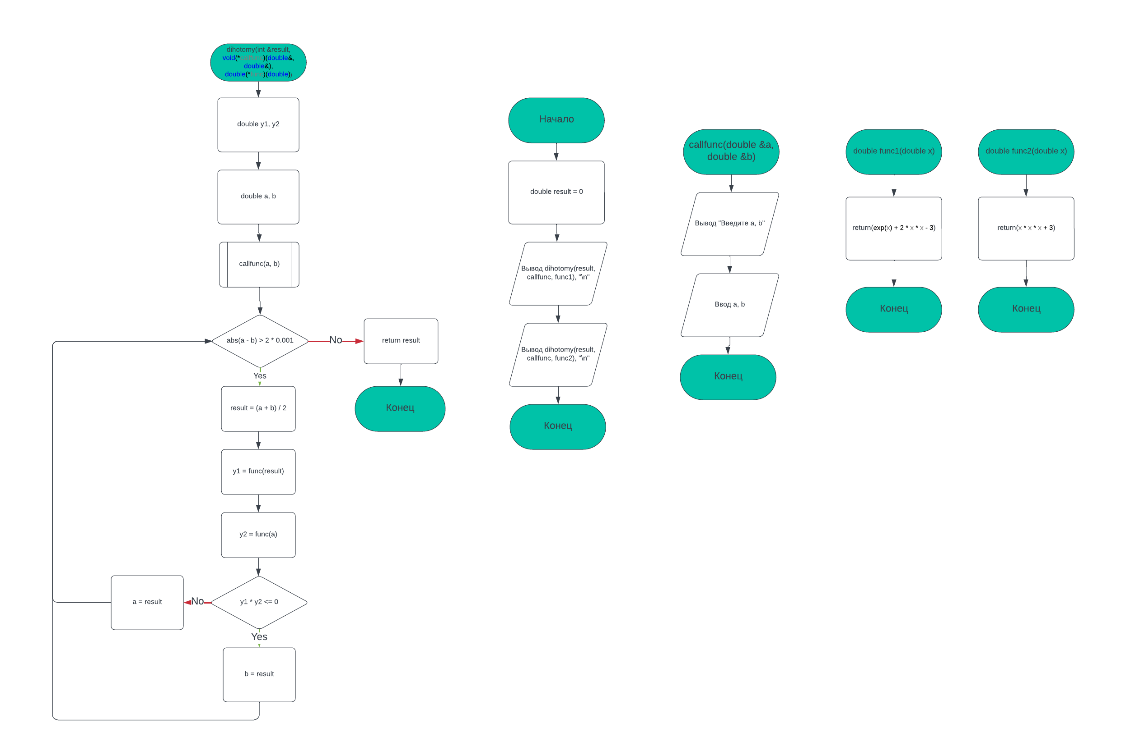
SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double result = 0;

cout << dihotomy(result, callfunc, func1) << '\n';

cout << dihotomy(result, callfunc, func2) << '\n';

}

­­

|  |  |
| --- | --- |
| 11 | Написать функцию с переменным числом параметров, которая находит минимальное значение матрицы. С ее помощью найти минимальные значения в n матрицах. |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

using namespace std;

void min\_el(int a,int n, int m, int\* first...) { // функция с переменным числом параметров

int\*\* p = &first;

for (int k = 0; k < a; k++) {

int mi\_elm = \*\*p;

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

for (int j = 0; j < m; j++) {

cout << \*\*p << " ";

if (mi\_elm > \*\*p) {

mi\_elm = \*\*p;

}

++\*(p);

}

cout << '\n';

p++;

}

cout << mi\_elm << '\n';

}

cout << '\n';

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

int a[5][6];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 6; j++) {

a[i][j] = rand() % 100 + 1;

}

}

int b[5][6];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 6; j++) {

b[i][j] = rand() % 100 + 1;

}

}

int c[5][6];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 6; j++) {

c[i][j] = rand() % 100 + 1;

}

}

int d[5][6];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 6; j++) {

d[i][j] = rand() % 100 + 1;

}

}

int f[5][6];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 6; j++) {

f[i][j] = rand() % 100 + 1;

}

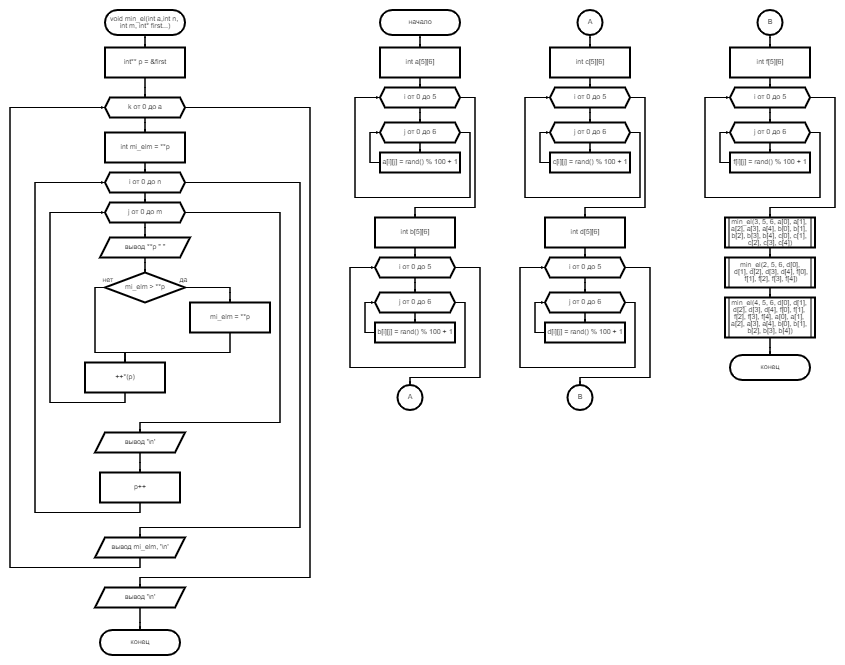
}

min\_el(3, 5, 6, a[0], a[1], a[2], a[3], a[4], b[0], b[1], b[2], b[3], b[4], c[0], c[1], c[2], c[3], c[4]);

min\_el(2, 5, 6, d[0], d[1], d[2], d[3], d[4], f[0], f[1], f[2], f[3], f[4]);

min\_el(4, 5, 6, d[0], d[1], d[2], d[3], d[4], f[0], f[1], f[2], f[3], f[4], a[0], a[1], a[2], a[3], a[4], b[0], b[1], b[2], b[3], b[4]);

}



­­

|  |  |
| --- | --- |
| 4 | x3 + 2x – 1,  ex − 2 |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

#include <iomanip>

using namespace std;

void callfunc(double& a, double& b) { // функция для считывания концов интервала

cout << "Введите a, b" << '\n';

cin >> a >> b;

}

double func2(double x) {

return(exp(x) - 2);

}

double func1(double x) {

return(x \* x \* x + 2 \* x - 1);

}

double dihotomy(double &result, void(\*callfunc)(double&, double&), double(\*func)(double)) { // функция для решения уравнений

double y1, y2;

double a, b;

callfunc(a, b);

while (abs(a - b) > 2 \* 0.001) {

result = (a + b) / 2;

y1 = func(result);

y2 = func(a);

if (y1 \* y2 <= 0) {

b = result;

}

else {

a = result;

}

}

return result;

}

int main() {

SetConsoleCP(1251);

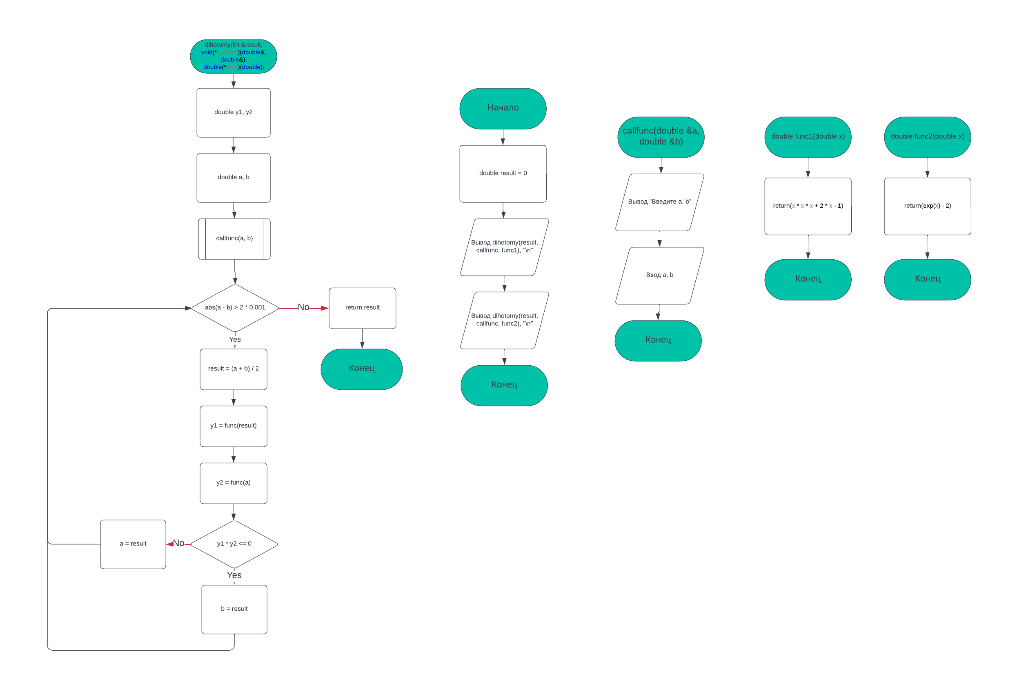
SetConsoleOutputCP(1251);

double result = 0;

cout << dihotomy(result, callfunc, func1) << '\n';

cout << dihotomy(result, callfunc, func2) << '\n';

}



|  |  |
| --- | --- |
| 4 | Написать функцию **mn** с переменным числом параметров, которая находит максимальное из чисел типа **int**. |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

using namespace std;

int mn(int n, int first ...) { // функция с переменным количеством параметров

int \*x = &first;

int max = \*x;

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

if (\*x > max) {

max = \*x;

}

x++;

}

return max;

}

int main() {

SetConsoleCP(1251);

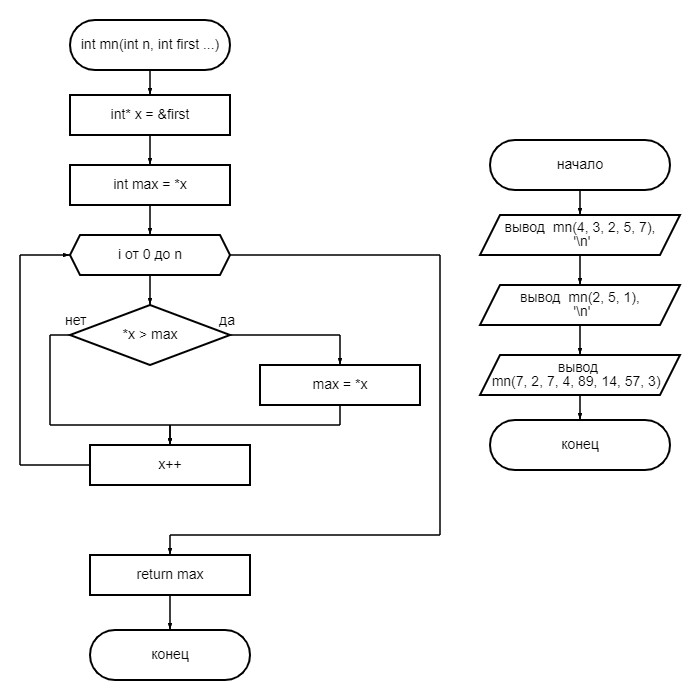
SetConsoleOutputCP(1251);

cout << mn(4, 3, 2, 5, 7) << '\n';

cout << mn(2, 5, 1) << '\n';

cout << mn(7, 2, 7, 4, 89, 14, 57, 3);

}



|  |  |
| --- | --- |
| 14 | e x + x − 4,  x2 – 4 |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

#include <iomanip>

using namespace std;

void callfunc(double& a, double& b) { // функция для считывания концов интервала

cout << "Введите a, b" << '\n';

cin >> a >> b;

}

double func2(double x) {

return(x \* x - 4);

}

double func1(double x) {

return(exp(x) + x - 4);

}

double dihotomy(double &result, void(\*callfunc)(double&, double&), double(\*func)(double)) { // функция для решения уравнений

double y1, y2;

double a, b;

callfunc(a, b);

while (abs(a - b) > 2 \* 0.001) {

result = (a + b) / 2;

y1 = func(result);

y2 = func(a);

if (y1 \* y2 <= 0) {

b = result;

}

else {

a = result;

}

}

return result;

}

int main() {

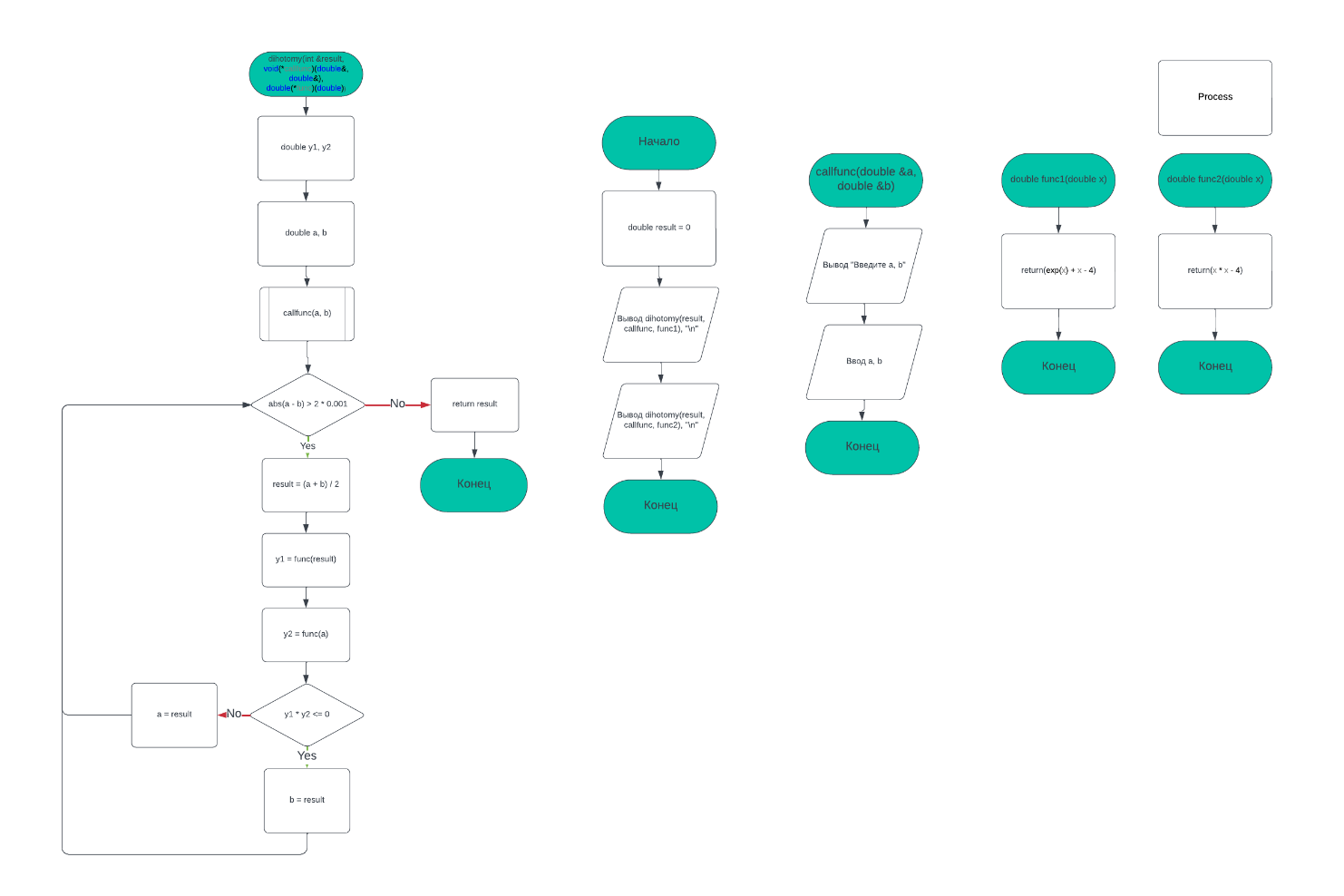
SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double result = 0;

cout << dihotomy(result, callfunc, func1) << '\n';

cout << dihotomy(result, callfunc, func2) << '\n';

}

|  |  |
| --- | --- |
| 14 | Написать функцию **compr**, которая «сжимает» строку, удаляя из нее все пробелы. С ее помощью сжать различные строки. |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <string>

using namespace std;

void compr(int n, const char a[]...) { // функция с переменным количеством параметров

const char\*\* f = &a;

string b = \*f;

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

b = \*f;

for (int j = 0; j < b.length(); j++) {

if (b[j] == ' ') {

b.erase(j, 1);

}

}

cout << b << '\n';

f++;

}

cout << '\n';

}

int main() {

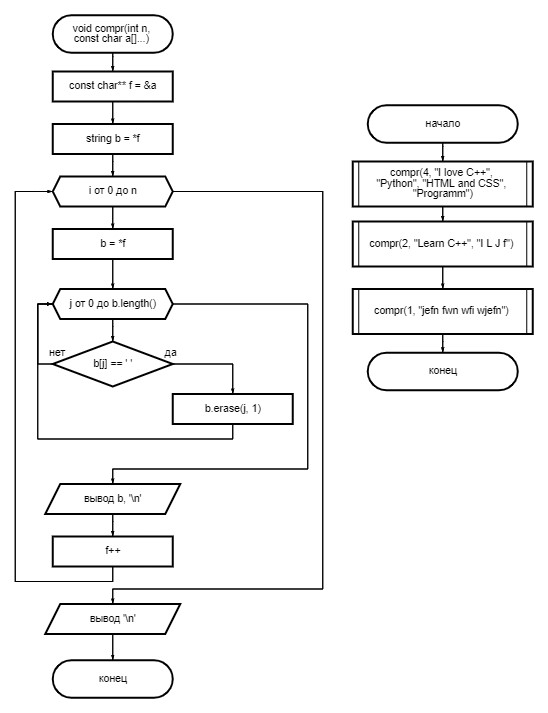
SetConsoleCP(1251);

SetConsoleOutputCP(1251);

compr(4, "I love C++", "Python", "HTML and CSS", "Programm");

compr(2, "Learn C++", "I L J f");

compr(1, "jefn fwn wfi wjefn");

}

|  |  |
| --- | --- |
| 8 | x3 + 3x – 1,  ex − 4 |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

#include <iomanip>

using namespace std;

void callfunc(double& a, double& b) { // функция для считывания концов интервала

cout << "Введите a, b" << '\n';

cin >> a >> b;

}

double func2(double x) {

return(exp(x) - 4);

}

double func1(double x) {

return(x \* x \* x + 3 \* x - 1);

}

double dihotomy(double &result, void(\*callfunc)(double&, double&), double(\*func)(double)) { // функция для решения уравнений

double y1, y2;

double a, b;

callfunc(a, b);

while (abs(a - b) > 2 \* 0.001) {

result = (a + b) / 2;

y1 = func(result);

y2 = func(a);

if (y1 \* y2 <= 0) {

b = result;

}

else {

a = result;

}

}

return result;

}

int main() {

SetConsoleCP(1251);

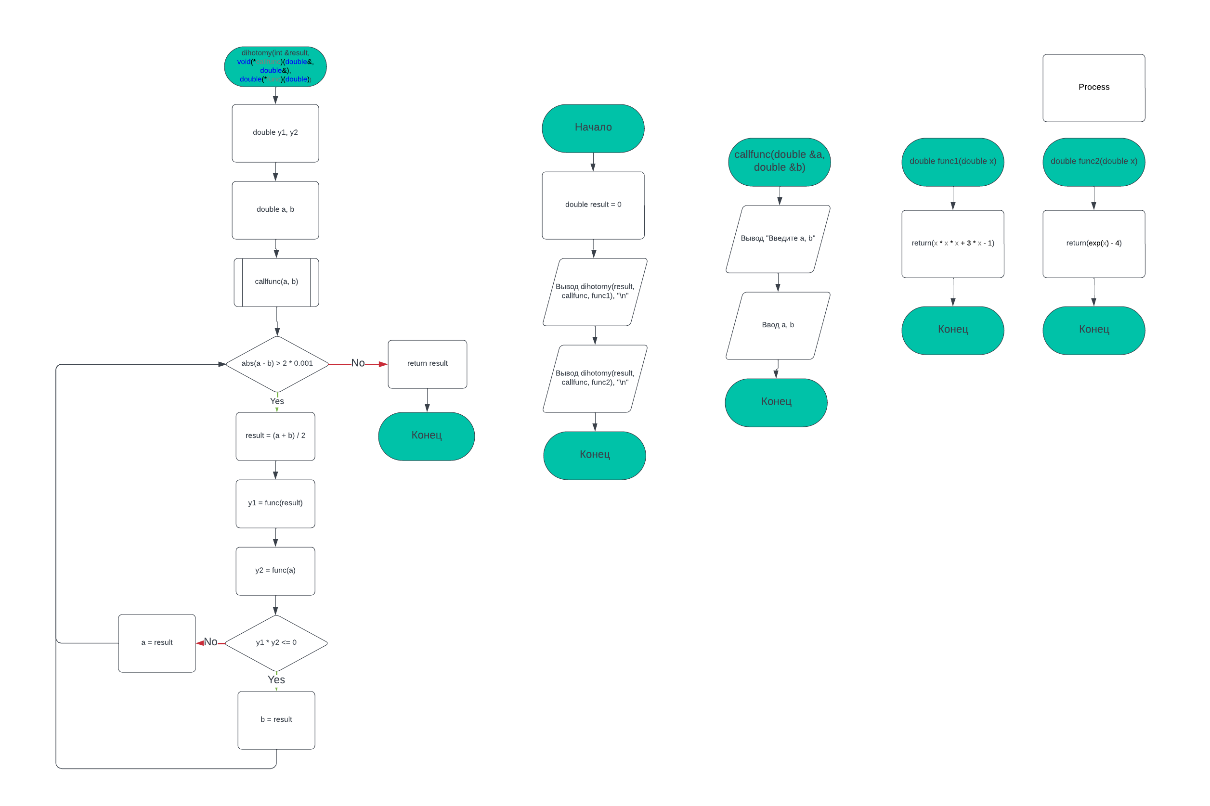
SetConsoleOutputCP(1251);

double result = 0;

cout << dihotomy(result, callfunc, func1) << '\n';

cout << dihotomy(result, callfunc, func2) << '\n';

}



|  |  |
| --- | --- |
| 8 | Написать функцию **mn** с переменным числом параметров, которая находит минимальное из чисел типа **int**. |

#include <iostream>

#include <Windows.h>

#include <ctype.h>

#include <cstring>

using namespace std;

int mn(int n, int first ...) { // функция с переменным количеством параметров

int\* x = &first;

int min = \*x;

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

if (\*x < min) {

min = \*x;

}

x++;

}

return min;

}

int main() {

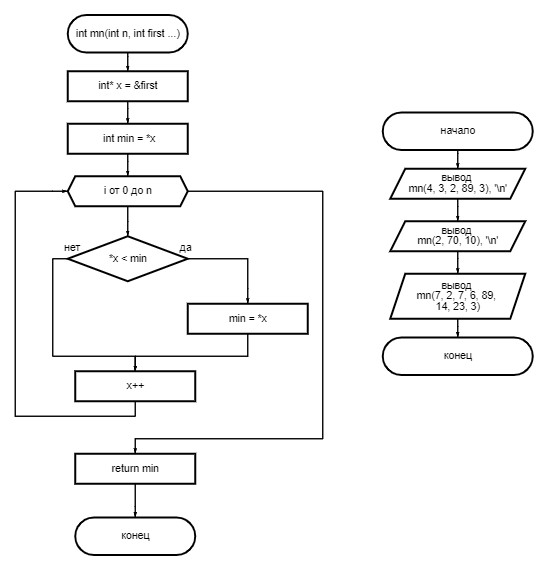
SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << mn(4, 3, 2, 89, 3) << '\n';

cout << mn(2, 70, 10) << '\n';

cout << mn(7, 2, 7, 6, 89, 14, 23, 3);

}