Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

 Лабараторная работа 9

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Рашэнне інжынерных задач на аснове цыклічных праграм»

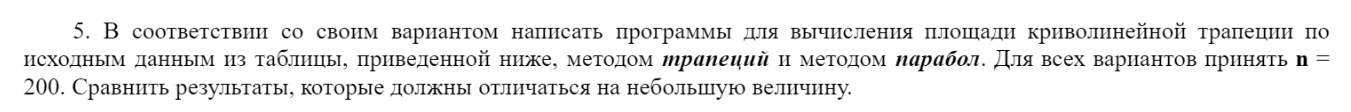
Выканала:

Студэнтка 1 курса 6 группы

Лускіна Вольга Аляксандраўна

Выкладчык: асс. Андронава М.В.

2023, Мінск



Варыянт 11

Заданне 1



1) Метад трапецый:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 1, b = 3, n = 200, s = 0, x, h;

h = (b - a) / n; // Вылічэнне крока разбіення

x = a;

for (x; x <= (b - h); x += h) // Цыкл для падліку сумы плошчаў трапецый; падлічваем плошчу бегучай трапецыі і дадаем яе да агульнай сумы

{

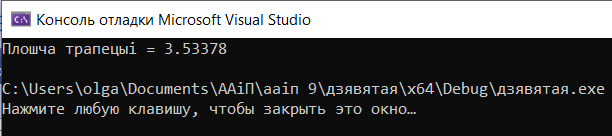
s = s + h \* (sin(x) + 1 + sin(x + h) + 1);

x = x + h; // Пераход да наступнай кропкі

}

cout << "Плошча трапецыі = " << s << endl;

}



2) Метад парабал

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double s = 0, x, n = 200, h, a = 1, b = 3, s1 = 0, s2 = 0, i = 1; // s1 i s2 - сумы значэнняў функцыі sin(x) + 1 у пэўных кропках

h = (b - a) / (2 \* n); // Вылічэнне крока разбіення

x = a + 2 \* h;

for (i; i < n; i++) // Цыкл падліку сумы значэнняў функцыі ў пэўных кропках

{

s2 += sin(x) + 1; // Дадаем значэнне функцыі да сумы s2

x += h; // Пераход да наступнай кропкі

s1 += sin(x) + 1; // Дадаем значэнне функцыі да сумы s1

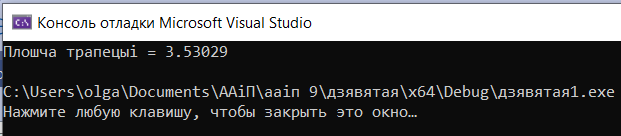
x += h; // Пераход да наступнай кропкі

}

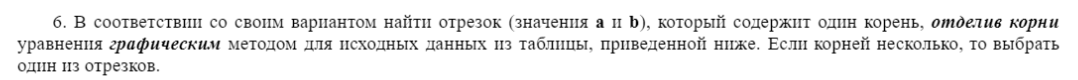
s = (h / 3) \* ((sin(a) + 1) + 4 \* (sin(a + h) + 1) + 4 \* s1 + 2 \* s2 + sin(b) + 1); // Падлічваем агульную плошчу

cout << "Плошча трапецыі = " << s << endl;

}



Заданне 2







1) Графічаскі метад

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, b;

double x1, e = 0.0001, x = 0;

cout << "Увядзіце значэння a і b " << endl;

cin >> a >> b;

if ((2\*a + a^3 - 7)\*(3\*a^2 > 0))

x1 = a; // Праверка ўмовы і выбар пачатковага значэння х1

else x1 = b;

while (abs(x1 - x) > e) // Цыкл для падліку набліжанага значэння кораня (метадам Ньютана)

{

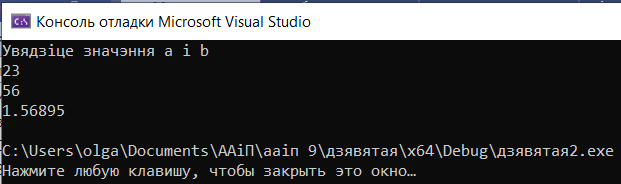
x = x1;

x1 = x - (x + x + x \* x \* x - 7) / (3 \* pow(x, 2) + 1);

}

cout << x1 << endl;

}



2) Метад дзітахоміі

#include <iostream>

#include <Windows.h>

#include <cmath>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a, b, x, e = 0.0001;

cout << "Увядзіце значэнні a і b: " << endl;

cin >> a >> b;

double fa = 2 \* a + pow(a, 3) - 7;

double fb = 2 \* b + pow(b, 3) - 7; // Падлік значэння функцыі на канцах інтэрвала: fa=f(a) i fb=f(b)

if (fa \* fb > 0) // Праверка, каб уведзеныя лікі былі розных знакаў

{

cout << "На дадзеным інтэрвале няма кораня" << endl;

}

x = (a + b) / 2.0;

double fx = 2 \* x + pow(x, 3) - 7;

while (abs(fx) > e) // Цыкл для падліку набліжанага значэння кораня (метадам дзялення напалову)

{

if (fa \* fx < 0)

{

b = x;

fb = fx;

}

else

{

a = x;

fa = fx;

} // Абнаўляем межы інтэрвала ў залежнасці ад знака перамнажэння значэнняў функцыі

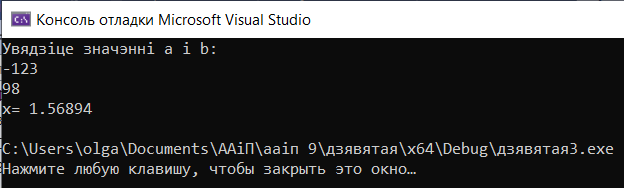
x = (a + b) / 2.0;

fx = 2 \* x + pow(x, 3) - 7; // Падлік новага значэння набліжэння кораня і значэння функцыі ў гэтым набліжэнні

}

cout << "x= " << x << endl;

}

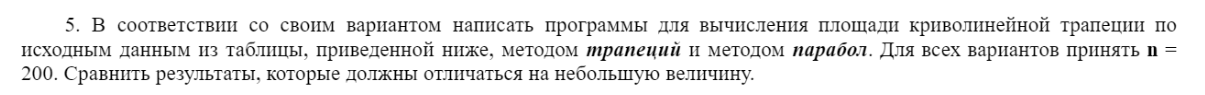




Дадатковыя заданні:

Варыянт 1

Заданне 1





1) Метад трапецый:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 1, b = 3, n = 200, s = 0, x, h;

h = (b - a) / n; // Вылічэнне крока разбіення

x = a;

for (x; x <= (b - h); x += h) // Цыкл для падліку сумы плошчаў трапецый; падлічваем плошчу бегучай трапецыі і дадаем яе да агульнай сумы

{

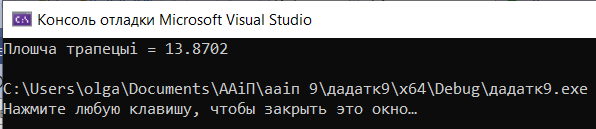
s = s + h \* (pow(x,3) - 3 + pow(x+h,3) -3);

x = x + h; // Пераход да наступнай кропкі

}

cout << "Плошча трапецыі = " << s << endl;

}



2) Метад парабал

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double s = 0, x, n = 200, h, a = 1, b = 3, s1 = 0, s2 = 0, i = 1; // s1 i s2 - сумы значэнняў функцыі x3 – 3 у пэўных кропках

h = (b - a) / (2 \* n); // Вылічэнне крока разбіення

x = a + 2 \* h;

for (i; i < n; i++) // Цыкл падліку сумы значэнняў функцыі ў пэўных кропках

{

s2 += pow(x, 3) - 3; // Дадаем значэнне функцыі да сумы s2

x += h; // Пераход да наступнай кропкі

s1 += pow(x, 3) - 3; // Дадаем значэнне функцыі да сумы s1

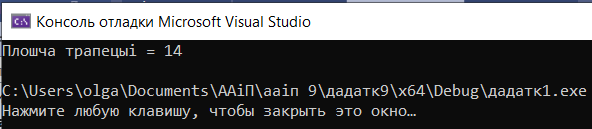
x += h; // Пераход да наступнай кропкі

}

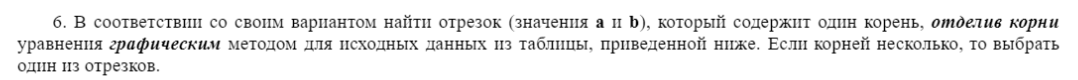
s = (h / 3) \* ((pow(a, 3) - 3) + 4 \* (pow(a + h, 3) - 3) + 4 \* s1 + 2 \* s2 + pow(b, 3) - 3); // Падлічваем агульную плошчу

cout << "Плошча трапецыі = " << s << endl;

}



Заданне 2







1) Графічаскі метад

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, b;

double x1, e = 0.0001, x = 0;

cout << "Увядзіце значэння a і b " << endl;

cin >> a >> b;

if ((a ^ 3 + a - 7) \* (3 \* a ^ 2 > 0))

x1 = a; // Праверка ўмовы і выбар пачатковага значэння х1

else x1 = b;

while (abs(x1 - x) > e) // Цыкл для падліку набліжанага значэння кораня (метадам Ньютана)

{

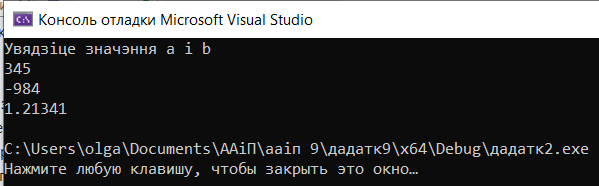
x = x1;

x1 = x - (x + x \* x \* x - 3) / (3 \* pow(x, 2) + 1);

}

cout << x1 << endl;

}



2) Метад дзітахоміі

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a, b, x, e = 0.0001;

cout << "Увядзіце значэнні a і b: " << endl;

cin >> a >> b;

double fa = pow(a, 3) + a - 3;

double fb = pow(b, 3) + b - 3; // Падлік значэння функцыі на канцах інтэрвала: fa=f(a) i fb=f(b)

if (fa \* fb > 0) // Праверка, каб уведзеныя лікі былі розных знакаў

{

cout << "На дадзеным інтэрвале няма кораня" << endl;

}

x = (a + b) / 2.0;

double fx = pow(x, 3) + x - 3;

while (abs(fx) > e) // Цыкл для падліку набліжанага значэння кораня (метадам дзялення напалову)

{

if (fa \* fx < 0)

{

b = x;

fb = fx;

}

else

{

a = x;

fa = fx;

} // Абнаўляем межы інтэрвала ў залежнасці ад знака перамнажэння значэнняў функцыі

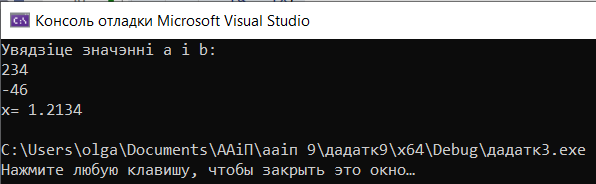
x = (a + b) / 2.0;

fx = pow(x, 3) + x - 3; // Падлік новага значэння набліжэння кораня і значэння функцыі ў гэтым набліжэнні

}

cout << "x= " << x << endl;

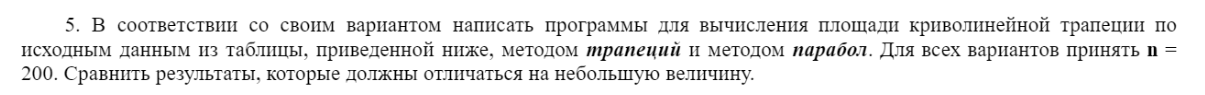
}





Варыянт 3

Заданне 1





1) Метад трапецый:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 1, b = 6, n = 200, s = 0, x, h;

h = (b - a) / n; // Вылічэнне крока разбіення

x = a;

for (x; x <= (b - h); x += h) // Цыкл для падліку сумы плошчаў трапецый; падлічваем плошчу бегучай трапецыі і дадаем яе да агульнай сумы

{

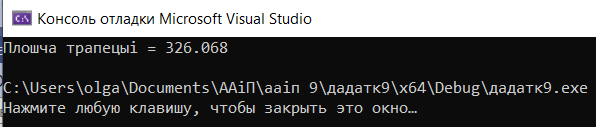
s = s + h \* (pow(x,3) + 1 + pow(x+h,3) + 1);

x = x + h; // Пераход да наступнай кропкі

}

cout << "Плошча трапецыі = " << s << endl;

}



2) Метад парабал

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double s = 0, x, n = 200, h, a = 1, b = 6, s1 = 0, s2 = 0, i = 1; // s1 i s2 - сумы значэнняў функцыі x3 – 3 у пэўных кропках

h = (b - a) / (2 \* n); // Вылічэнне крока разбіення

x = a + 2 \* h;

for (i; i < n; i++) // Цыкл падліку сумы значэнняў функцыі ў пэўных кропках

{

s2 += pow(x, 3) + 1; // Дадаем значэнне функцыі да сумы s2

x += h; // Пераход да наступнай кропкі

s1 += pow(x, 3) + 1; // Дадаем значэнне функцыі да сумы s1

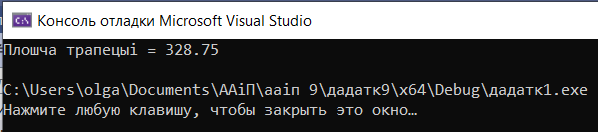
x += h; // Пераход да наступнай кропкі

}

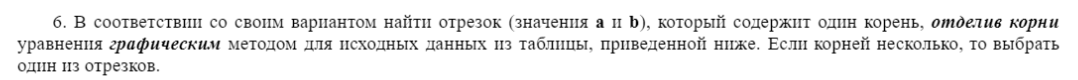
s = (h / 3) \* ((pow(a, 3) + 1) + 4 \* (pow(a + h, 3) + 1) + 4 \* s1 + 2 \* s2 + pow(b, 3) + 1); // Падлічваем агульную плошчу

cout << "Плошча трапецыі = " << s << endl;

}



Заданне 2







1) Графічаскі метад

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, b;

double x1, e = 0.0001, x = 0;

cout << "Увядзіце значэння a і b " << endl;

cin >> a >> b;

if ((a ^ 3 + 2 \* a - 1) \* (3 \* a ^ 2 > 0))

x1 = a; // Праверка ўмовы і выбар пачатковага значэння х1

else x1 = b;

while (abs(x1 - x) > e) // Цыкл для падліку набліжанага значэння кораня (метадам Ньютана)

{

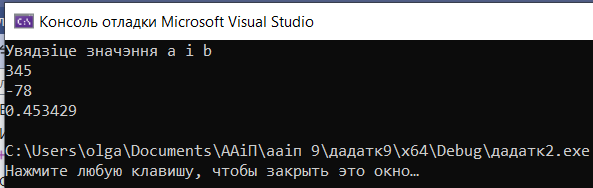
x = x1;

x1 = x - (x \* x \* x + 2 \* x - 1) / (3 \* pow(x, 2) + 1);

}

cout << x1 << endl;

}



2) Метад дзітахоміі

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a, b, x, e = 0.0001;

cout << "Увядзіце значэнні a і b: " << endl;

cin >> a >> b;

double fa = pow(a, 3) + 2 \* a - 1;

double fb = pow(b, 3) + 2 \* b - 1; // Падлік значэння функцыі на канцах інтэрвала: fa=f(a) i fb=f(b)

if (fa \* fb > 0) // Праверка, каб уведзеныя лікі былі розных знакаў

{

cout << "На дадзеным інтэрвале няма кораня" << endl;

}

x = (a + b) / 2.0;

double fx = pow(x, 3) + 2 \* x - 1;

while (abs(fx) > e) // Цыкл для падліку набліжанага значэння кораня (метадам дзялення напалову)

{

if (fa \* fx < 0)

{

b = x;

fb = fx;

}

else

{

a = x;

fa = fx;

} // Абнаўляем межы інтэрвала ў залежнасці ад знака перамнажэння значэнняў функцыі

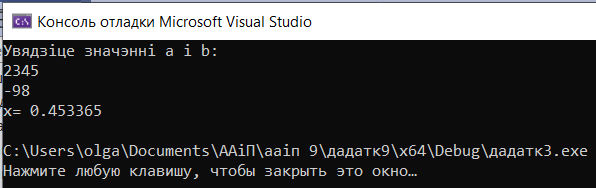
x = (a + b) / 2.0;

fx = pow(x, 3) + 2 \* x - 1; // Падлік новага значэння набліжэння кораня і значэння функцыі ў гэтым набліжэнні

}

cout << "x= " << x << endl;

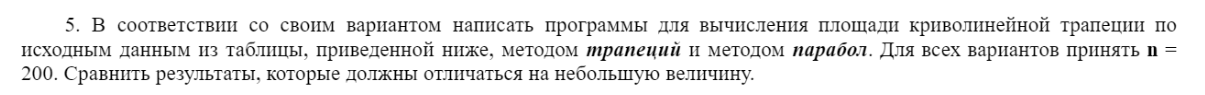
}





Варыянт 8

Заданне 1





1) Метад трапецый:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 0, b = 3, n = 200, s = 0, x, h;

h = (b - a) / n; // Вылічэнне крока разбіення

x = a;

for (x; x <= (b - h); x += h) // Цыкл для падліку сумы плошчаў трапецый; падлічваем плошчу бегучай трапецыі і дадаем яе да агульнай сумы

{

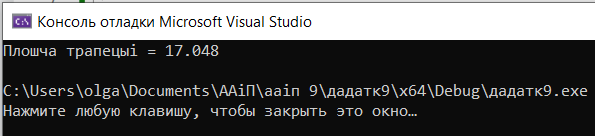
s = s + h \* (pow(x,3) - 1 + pow(x+h,3) - 1);

x = x + h; // Пераход да наступнай кропкі

}

cout << "Плошча трапецыі = " << s << endl;

}



2) Метад парабал

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double s = 0, x, n = 200, h, a = 0, b = 3, s1 = 0, s2 = 0, i = 1; // s1 i s2 - сумы значэнняў функцыі x3 – 3 у пэўных кропках

h = (b - a) / (2 \* n); // Вылічэнне крока разбіення

x = a + 2 \* h;

for (i; i < n; i++) // Цыкл падліку сумы значэнняў функцыі ў пэўных кропках

{

s2 += pow(x, 3) - 1; // Дадаем значэнне функцыі да сумы s2

x += h; // Пераход да наступнай кропкі

s1 += pow(x, 3) - 1; // Дадаем значэнне функцыі да сумы s1

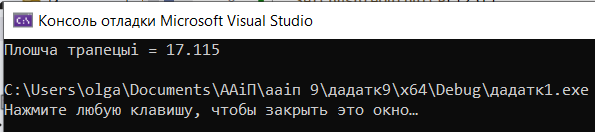
x += h; // Пераход да наступнай кропкі

}

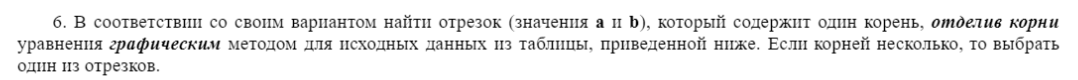
s = (h / 3) \* ((pow(a, 3) - 1) + 4 \* (pow(a + h, 3) - 1) + 4 \* s1 + 2 \* s2 - pow(b, 3) - 1); // Падлічваем агульную плошчу

cout << "Плошча трапецыі = " << s << endl;

}



Заданне 2







#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, b;

double x1, e = 0.0001, x = 0;

cout << "Увядзіце значэння a і b " << endl;

cin >> a >> b;

if ((a ^ 3 + a - 4) \* (3 \* a ^ 2 > 0))

x1 = a; // Праверка ўмовы і выбар пачатковага значэння х1

else x1 = b;

while (abs(x1 - x) > e) // Цыкл для падліку набліжанага значэння кораня (метадам Ньютана)

{

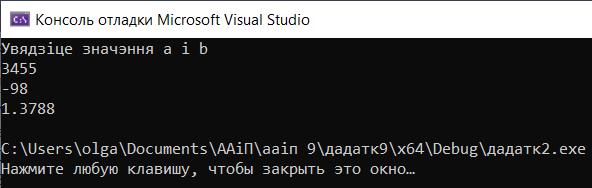
x = x1;

x1 = x - (x \* x \* x + x - 4) / (3 \* pow(x, 2) + 1);

}

cout << x1 << endl;

}



2) Метад дзітахоміі

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a, b, x, e = 0.0001;

cout << "Увядзіце значэнні a і b: " << endl;

cin >> a >> b;

double fa = pow(a, 3) + a - 4;

double fb = pow(b, 3) + b - 4; // Падлік значэння функцыі на канцах інтэрвала: fa=f(a) i fb=f(b)

if (fa \* fb > 0) // Праверка, каб уведзеныя лікі былі розных знакаў

{

cout << "На дадзеным інтэрвале няма кораня" << endl;

}

x = (a + b) / 2.0;

double fx = pow(x, 3) + x - 4;

while (abs(fx) > e) // Цыкл для падліку набліжанага значэння кораня (метадам дзялення напалову)

{

if (fa \* fx < 0)

{

b = x;

fb = fx;

}

else

{

a = x;

fa = fx;

} // Абнаўляем межы інтэрвала ў залежнасці ад знака перамнажэння значэнняў функцыі

x = (a + b) / 2.0;

fx = pow(x, 3) + x - 4; // Падлік новага значэння набліжэння кораня і значэння функцыі ў гэтым набліжэнні

}

cout << "x= " << x << endl;

}

