**УСЛОЖНЕННЫЕ ЗАДАНИЯ**

/\* Код Присяжного Максима

Определите класс для дробей -рациональных чисел, являющихся отношением двух целых чисел. Напишите функции - члены для сложения,

вычетания, умножения и деления дробей и приведения дроби к общему знаменателю. Выполните тестирование и отладку созданной вами

программы обработки дробей.

\*/

#include <iostream>

#include <conio.h>

void main()

{

class DrobCS

{

public:

// ПЕРЕМЕНННЫЕ

int chisl1,znam1;

int chisl2,znam2;

// МЕТОДЫ

void InputDrob()

{// ВВОД ДРОБИ

do

{

system("cls");

std::cout<<"Input chislitel of drob A\n";

std::cin>>chisl1;

std::cout<<"Input znamenatel of drob A !=0\n";

std::cin>>znam1;

std::cout<<"Input chislitel of drob B\n";

std::cin>>chisl2;

std::cout<<"Input znamenatel of drob B !=0\n";

std::cin>>znam2;

}

while (znam1==0 || znam2==0);

}

void OutputDrob()

{// ФОРМАТНЫЙ ВЫВОД ДРОБИ

char string[16];

int k,k1;

int max;

sprintf(string,"%g",chisl1);

k=strlen(string);

sprintf(string,"%g",znam1);

k1=strlen(string);

max=((k>k1)?k:k1);

std::cout<<'\n';

std::cout.width(max+1);

std::cout.setf(std::ios::left);

std::cout<<"A";

std::cout<<"B\n\n";

std::cout.width(max+1);

std::cout.setf(std::ios::left);

std::cout<<chisl1;

std::cout<<chisl2<<'\n';

for (int i=1;i<=max;i++) std::cout<<'\_';

std::cout<<' ';

for (int i=1;i<=max;i++) std::cout<<'\_';

std::cout<<'\n';

std::cout.width(max+1);

std::cout.setf(std::ios::left);

std::cout<<znam1;

std::cout<<znam2<<'\n';

}

double SumDrob()

{// СУММА

return (chisl1/(double)znam1+chisl2/(double)znam2);

}

double SubDrobAB()

{// РАЗНОСТЬ А-В

return (chisl1/(double)znam1-chisl2/(double)znam2);

}

double SubDrobBA()

{// РАЗНОСТЬ В-А

return (chisl2/(double)znam2-chisl1/(double)znam1);

}

double MulDrob()

{// ПРОИЗВЕДЕНИЕ ДРОБЕЙ

return (chisl2/(double)znam2\*chisl1/(double)znam1);

}

double DivDrobAB()

{// ДЕЛЕНИЕ А/B

return ((chisl1/(double)znam1)/(chisl2/(double)znam2));

}

double DivDrobBA()

{// ДЕЛЕНИЕ B/A

return ((chisl2/(double)znam2)/(chisl1/(double)znam1));

}

int gcd()

{// НОД (ВСПОМАГАТЕЛЬНЫЙ МЕТОД)

while (znam1 != znam2)

if (znam1 > znam2)

znam1 -= znam2;

else

znam2 -= znam1;

return znam1;

}// НОК

int NokDrobBA ()

{

return (znam1 \* znam2)/ gcd ();

}

};

DrobCS drob;

drob.InputDrob();

drob.OutputDrob();

std::cout<<"\nA+B="<<drob.SumDrob();

std::cout<<"\nA-B="<<drob.SubDrobAB();

std::cout<<"\nB-A="<<drob.SubDrobBA();

std::cout<<"\nA\*B="<<drob.MulDrob();

std::cout<<"\nA/B="<<drob.DivDrobAB();

std::cout<<"\nB/A="<<drob.DivDrobBA();

std::cout<<"\nNOK(A,B)="<<drob.NokDrobBA();

\_getch();

}

**USER SCREEN:**

Input chislitel of drob A

2

Input znamenatel of drob A !=0

3

Input chislitel of drob B

3

Input znamenatel of drob B !=0

5

A B

2 3

\_\_ \_\_

3 5

A+B=1.26667

A-B=0.0666667

B-A=-0.0666667

A\*B=0.4

A/B=1.11111

B/A=0.9

NOK(A,B)=15

**ОСНОВНЫЕ ЗАДАНИЯ**

/\*Код Присяжного Максима

9) Дано целое число N. Сформировать матрицу a(i,j) порядка N, для которой: a(i,j)=sin(i+j+k);

где k - псевдослучайное число. Упорядочить строки матрицы по убыванию минимальных элементов строк.

\*/

#include <iostream>

#include <conio.h>

#include <time.h>

void main()

{

class ArrayCS

{

public:

// ПЕРЕМЕНННЫЕ

long double mas[100][100];

int N;

// МЕТОДЫ

void InputArr() // ВВОД МАССИВА

{

int k;

std::cout<<"Input size of array:"<<'\n';

std::cin>>N;

system("cls");

srand(time(NULL));

for (int i=0;i<N;i++)

for (int j=0;j<N;j++)

{

k=rand() % 65535 - 32767;

mas[i][j]=sin(long double(i+j+k));

}

}

void OutputArr() // ВЫВОД МАССИВА

{

for (int i=0;i<N;i++)

{

for (int j=0;j<N;j++)

{

//std::cout<<mas[i][j]<<' ';

std::cout<<mas[i][j]<<' ';

}

std::cout<<'\n';

}

}

void SortArray() // СОРТИРОВКА МАССИВА ПО УБЫВАНИЮ

{

std::cout<<"Press ENTER to sort by decrement:"<<"\n\n";

char key;

do

key=\_getch();

while (key!=13);

long double mas2[100001];

long double buffer=mas[0][0];

long double buffer2=0;

int k=0;

for (int i=0;i<N;i++)

{

buffer=mas[i][0];

for (int j=1;j<N;j++) if (mas[i][j]<buffer) buffer=mas[i][j];

\*(mas2+k)=buffer;

k++;

}

buffer=0;

std::cout<<"Minimums:\n\n";

for (int i=0;i<k;i++) std::cout<<mas2[i]<<'\n';

for (int i=1; i<N ; i++)

for (int j=0; j<N-i; j++)

if (\*(mas2 +j+1)>mas2 [j])

{

buffer=\*(mas2+j);

\*(mas2+j)=\*(mas2+j+1);

\*(mas2+j+1)=buffer;

for (k=0;k<N;k++)

{

buffer2=mas[j][k];

mas[j][k]=mas[j+1][k];

mas[j+1][k]=buffer2;

}

}

std::cout<<'\n';

std::cout<<"Sorted minimums:\n\n";

for (int i=0;i<k;i++) std::cout<<\*(mas2+i)<<'\n';

}

};

ArrayCS a;

a.InputArr();

std::cout<<"Your array:"<<"\n\n";

a.OutputArr();

std::cout<<'\n';

a.SortArray();

std::cout<<'\n';

std::cout<<"Sorted array:"<<"\n\n";

a.OutputArr();

\_getch();

}

**USER SCREEN:**

Input size of array:

5

Your array:

-0.938836 -0.995449 0.304265 -0.393488 -0.937063

0.973139 0.99708 0.732308 0.466092 0.313715

0.810228 -0.777726 0.0273346 0.314231 0.999739

-0.964068 0.482104 -0.98525 -0.273545 -0.134383

0.998375 -0.929485 0.594836 0.41962 -0.900665

Press ENTER to sort by decrement:

Minimums:

-0.995449

0.313715

-0.777726

-0.98525

-0.929485

Sorted minimums:

0.313715

-0.777726

-0.929485

-0.98525

-0.995449

Sorted array:

0.973139 0.99708 0.732308 0.466092 0.313715

0.810228 -0.777726 0.0273346 0.314231 0.999739

0.998375 -0.929485 0.594836 0.41962 -0.900665

-0.964068 0.482104 -0.98525 -0.273545 -0.134383

-0.938836 -0.995449 0.304265 -0.393488 -0.937063