**Лабораторна робота №6**

**Виконав студент групи ТІ-92 1-го курсу**

**П.І.Б Черноусов Денис Ігорович**

Завдання :

Задача 6.1. Дано неполное определение класса:

class strtype {

char \*p;

int len;

public:

char \*getstring() { return p; }

int getlength() { return len; }

};

Добавьте в это определение два конструктора. В первом не должно быть параметров. Он должен выделять 255 байтов памяти (с помощью оператора new), инициализировать эту память нулевой строкой и устанавливать переменную len равной 255. Во втором конструкторе должно быть два параметра. Первый — это строка, используемая при инициализации, второй — число выделяемых байтов. Во второй версии конструктора должно выделяться заданное количество памяти, в которую должна помещаться копия строки. Необходимо реализовать полный контроль границ массива (подсказка: см. Unit 5) и, разработать в короткую программу вывода, показать, что оба конструктора работают так, как это было задумано.

Задача 6.2. В Unit 2 (см. файл Pract02.pdf, Задача 2.14) вы создали имитацию секундомера. Модифицируйте ваше решение так, чтобы в классе stopwatch был и конструктор без параметров (как это уже сделано) и его перегруженная версия для доступа к системному времени через стандартную функцию clock(). Покажите, что внесенные изменения работают. Совет от ВВ: Подумайте о том, каким образом перегруженный конструктор может быть полезен для ваших собственных программных задач, например, в курсовой работе.

Задача 6.3. Конструктор копий вызывается и в тех случаях, когда функция генерирует временный объект, используемый в качестве ее возвращаемого значения (для тех функций, которые возвращают объекты). Зная это, рассмотрим следующий результат работы программы:

Constructing normally

Constructing normally

Constructing copy

Constructing copy

Объясните, что именно там происходит и почему.

Задача 6.4. Объясните, почему следующая программа заканчивается сообщением Aborted (core dumped) и исправьте ее:

#include

#include using namespace std;

class myсlass { int \*ptr;

public: myclass(int i);

~myсlass() { delete ptr; }

friend int getval(myclass obj);

myclass::myсlass(int i) {

ptr = new int;

if (!ptr) { exit(1); }

\*ptr = i; }

int getval(myclass obj) { return \*obj.ptr; // get value

}

int main() {

myclass a(1), b(2);

cout << getval(a) << " " << getval(b) << "\n";

cout << getval(a) << " " << getval(b);

return 0;

}

Задача 6.5. В стандартной библиотеке C++ существует функция strtol(), имеющая следующий прототип: long strtoi(const char \*start, const \*\*end, int base); Функция преобразует обозначающую число строку, на которую ссылается указатель start, в длинное целое. Число base задает основание системы счисления этого числа. При возвращении функцией своего значения указатель end ссылается на символ в строке, следующий сразу за последней цифрой строки. озвращаемое длинное целое эквивалентно тому числу, которое записано в строке. Диапазон значений base от 2 до 38. Однако наиболее часто основание системы счисления равно 10. Создайте функцию mystrtol(), работающую точно так же, как и функция strtol(), но аргумент 10 должен передаваться параметру base по умолчанию. Разрешается свободно пользоваться функцией strtol() для фактического преобразования. Для этого в программу требуется включить заголовок . Покажите, что ваша версия работает правильно.

Задача 6.6. В большинстве компиляторов C++ применяются нестандартные функции, управляющие позиционированием курсора и другими аналогичными действиями. Вы помните, что вам ранее уже было разрешено пользоваться любыми компиляторами, а не только g++? Если в используемом вами компиляторе применяются такие функции, создайте функцию myclreol(), которая стирает строку, начиная от текущей позиции курсора до конца строки. Передайте этой функции параметр, задающий число стираемых позиций. Если параметр не задавать, то по умолчанию должна стираться вся строка. В противном случае должно стираться число символьных позиций, заданное параметром.

Задача 6.7. Попытайтесь провести компиляцию всех программ (см. Unit6, "Overloading And Ambiguity"), в которых имеет место неоднозначность. Сопоставьте сообщения компилятора, объясните и запомните эти сообщения об ошибках. Это поможет вам сразу распознать ошибки неоднозначности, если они появятся в ваших программах. Задача 6.8. Ниже приведены две перегруженные функции. Покажите в демонстрационной программе, как получить и как использовать адрес каждой из них: int dif(int a, int b) { return a-b; } float dif(float a, float b) { return a-b; }

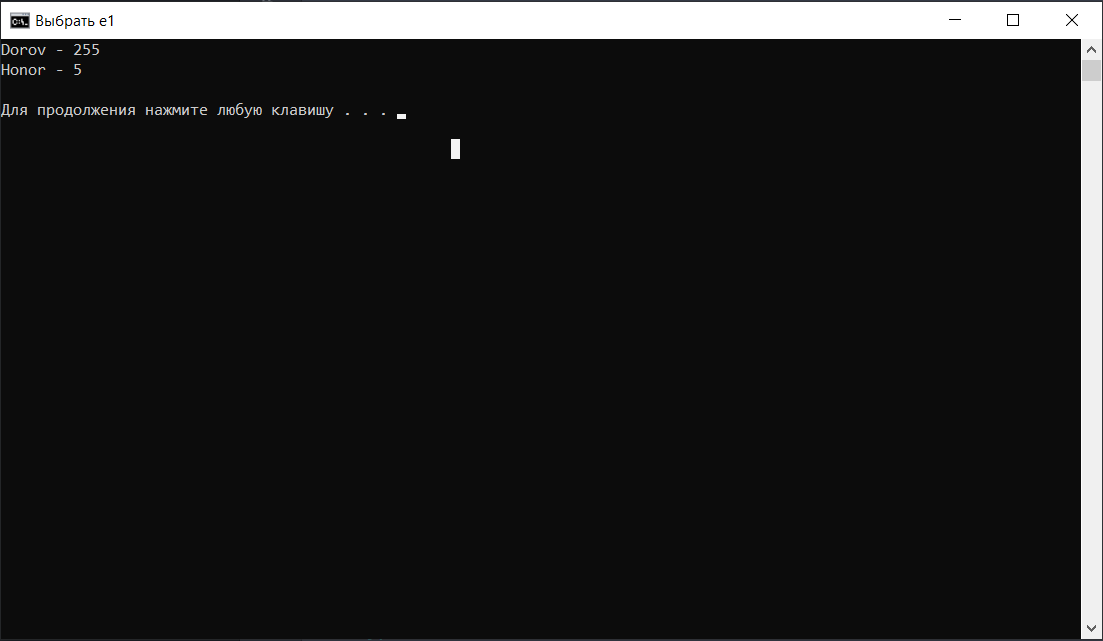
Задача 6.9. Перегрузите конструктор Date() (См. Unit6 "Select the most convenient method of initializing an object") так, чтобы он имел параметр типа time\_t. Подсказка: Вспомните, что time\_t - это тип данных, определенный стандартными библиотечными функциями времени и даты компилятора C++.

Задача 6.10. Создайте функцию reverse() с двумя параметрами. Первый параметр str - это указатель на строку, порядок следования символов в которой, после возвращения функцией своего значения, должен быть заменен на обратный. Второй параметр count задает количество переставляемых в строке str символов. Значение count по умолчанию должно быть таким, чтобы в случае его не задания функция reverse() меняла порядок следования символов в целой строке.

Задача 6.11. Создайте функцию order(), которая получает два параметра-ссылки на целые. Если первый аргумент больше второго, поменяйте их значения. В противном случае ничего делать не надо. Таким образом, порядок следования двух аргументов, используемых при вызове функции order(), должен быть таким, чтобы всегда после возвращения функцией своего значения первый аргумент был меньше второго. Например, если дано int x=1, y=0; order(x, y); то после вызова функции х будет равен 0, а у будет равен 1.

Задача 6.12. Пусть дано следующее неполное описание класса, добавьте конструкторы так, чтобы оба объявления в функции main() были правильны. Подсказка: вам необходимо дважды перегрузить конструктор samp(). class samp { int a; public: // add constructor functions int get\_a() { return a; } }; int main() { samp obj(88); // init obj a to 88 samp objarray[10]; // non - initialized 10-element array // ... }

Завдання 6.1



Код :

#include <iostream>

#include <cstring>

using namespace std;

class strtype {

char \*p;

int len;

public:

char \*getstring() { return p; }

int getlength() { return len; }

strtype(){

len = 255;

p = new char[255];

strcpy(p,"Dorov");

}

strtype (char\* s){

len = strlen(s) ;

p = new char[len];

strcpy( p , s );

}

};

int main(){

strtype ob1;

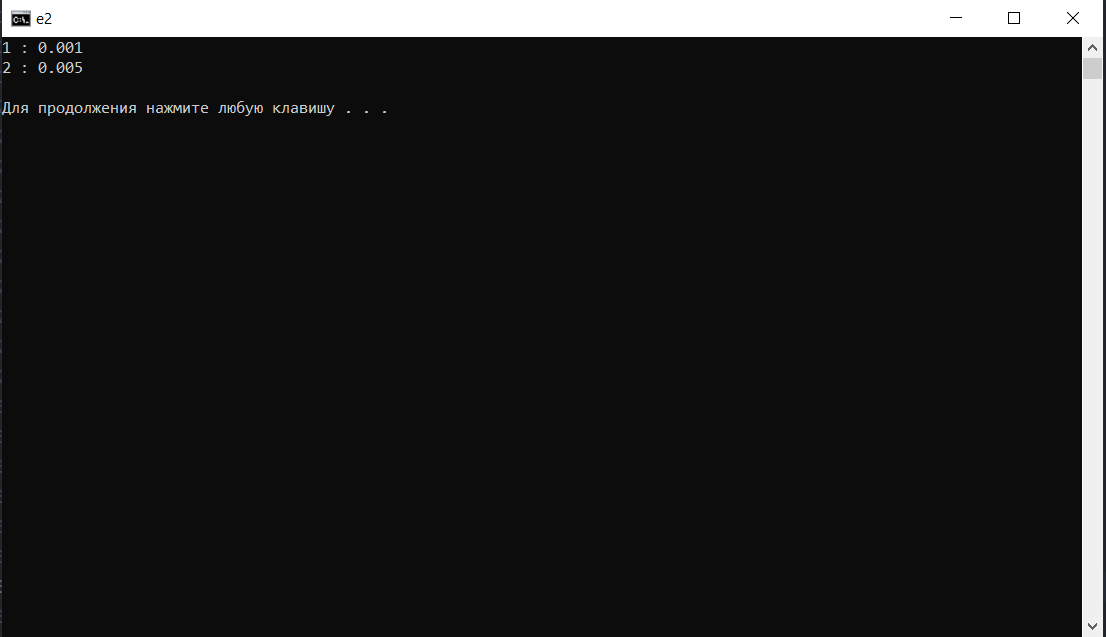
printf("%s - %d \n", ob1.getstring() , ob1.getlength() );

strtype ob2("Honor");

printf("%s - %d \n", ob2.getstring() , ob2.getlength() );

}

Завдання 6.2



Код:

#include <iostream>

#include <ctime>

using namespace std;

class timer {

clock\_t start1;

clock\_t start2;

public:

timer();

~timer();

timer(clock\_t q){

start1 = q ;

}

};

timer::timer() {

start2 = clock();

}

int n = 1 ;

timer::~timer() {

clock\_t end;

end = clock();

if (n) {

cout << "1 : "<< (end-start1) / (double)CLOCKS\_PER\_SEC << endl ;

n = 0 ;

}else cout << "2 : "<< (end-start2) / (double)CLOCKS\_PER\_SEC << endl ;

}

int main() {

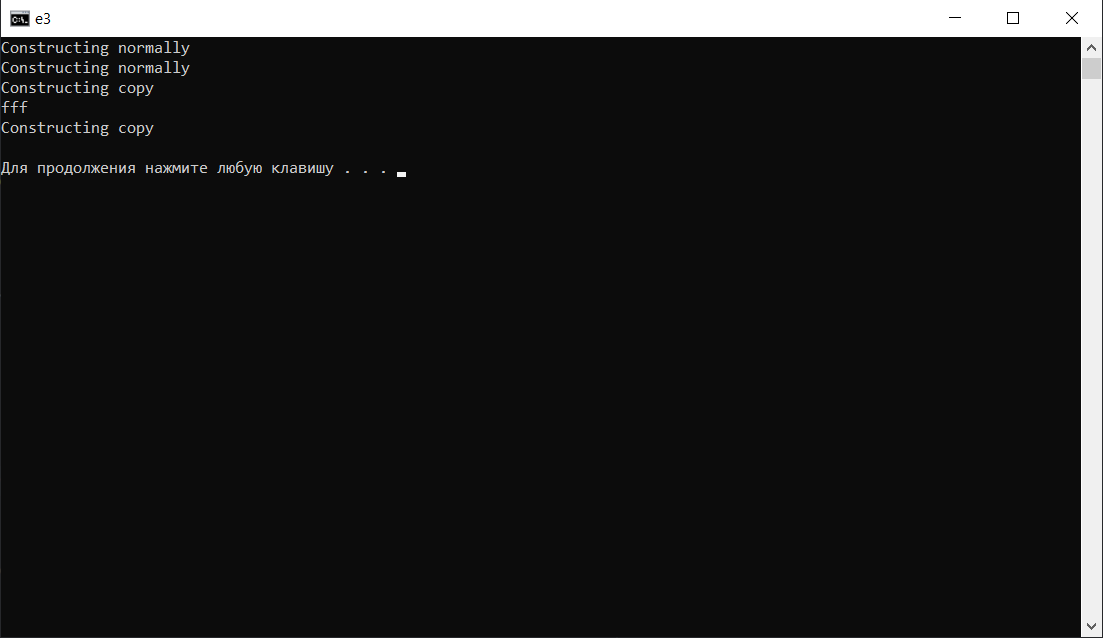
clock\_t q = clock();

timer ob1(q);

timer ob2;

}

Завдання 6.3



Код :

#include <iostream>

using namespace std;

class myclass {

public:

myclass();

myclass( myclass &obj );

myclass f();

myclass g( myclass obj );

};

myclass::myclass() {

cout << "Constructing normally\n";

}

myclass::myclass( myclass &obj) {

cout << "Constructing copy\n";

}

myclass myclass::f() {

myclass temp; // 2ий Constructing normally

return temp;

}

myclass myclass::g( myclass obj ) { // 1ий Constructing copy

printf("fff\n" );

return obj; // 2ий Constructing copy

}

int main() {

myclass obj; // 1ий Constructing normally

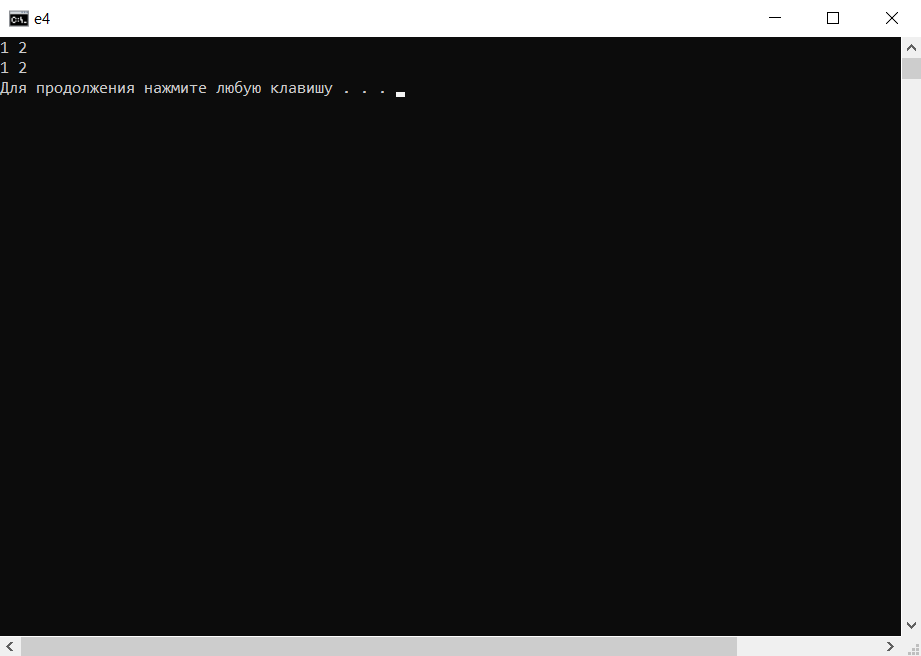
obj = obj.f();

obj.g(obj);

return 0;

}

Завдання 6.4



Код :

#include <iostream >

#include <cstdlib >

using namespace std;

class myclass {

int \*ptr;

public:

myclass(int i);

~myclass() { delete ptr; }

friend int getval(myclass \* obj);

};

myclass::myclass(int i) {

ptr = new int;

if (!ptr) { exit(1); }

\*ptr = i;

}

int getval(myclass \*obj) {

return \*obj->ptr;

}

int main() {

myclass a(1), b(2);

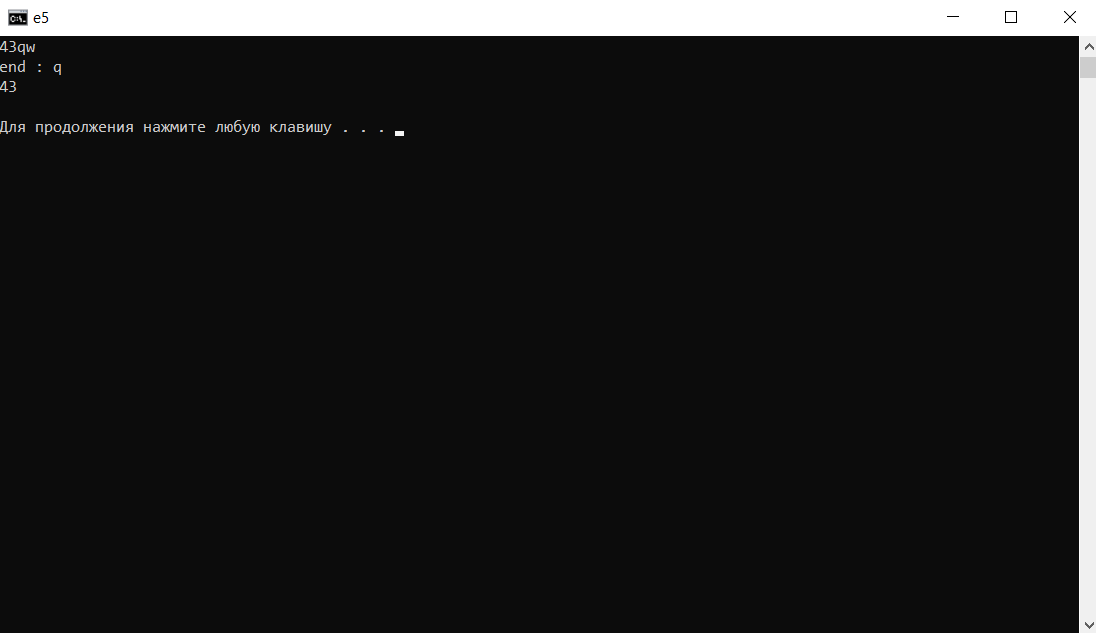
cout << getval(&a) << " " << getval(&b) << "\n";

cout << getval(&a) << " " << getval(&b);

return 0 ;

}

Завдання 6.5



Код :

#include <iostream >

#include <cstdlib >

using namespace std;

long int mystrtol (char str[30], char \* ch ){

long int l = strtol(str, &ch, 10);

printf("end : %c\n", \*ch );

return l;

}

int main(){

char str [30] ;

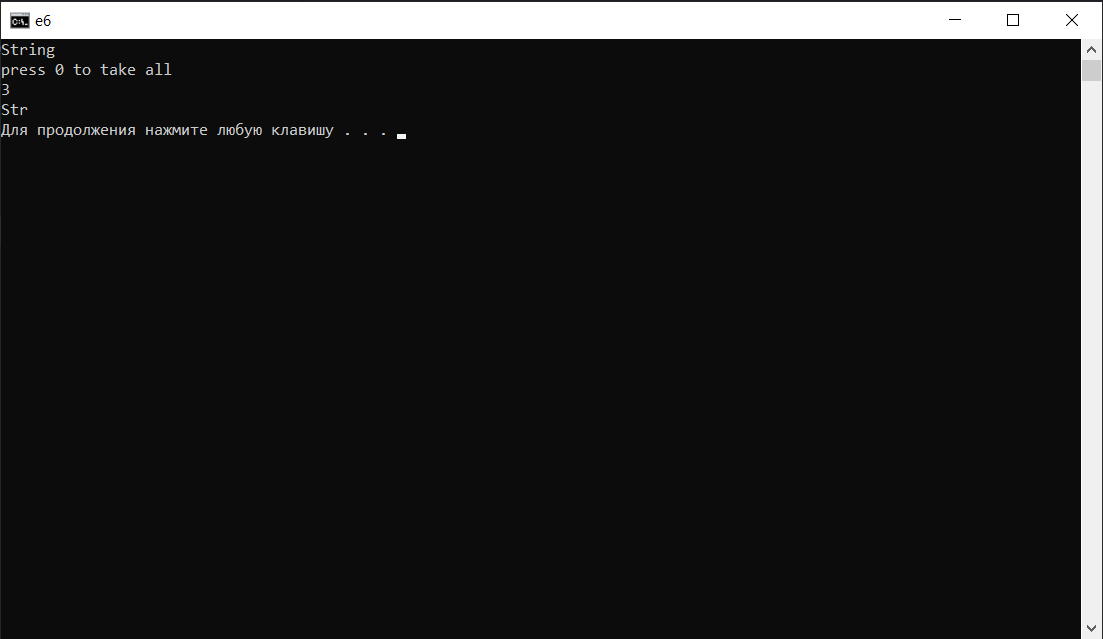
cin >> str ;

char \* ch ;

printf("%d\n", mystrtol (str , ch ) );

}

Завдання 6.6



Код :

#include <iostream>

#include <cstring>

#include <Windows.h>

using namespace std;

void myclreol(int len){

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

cout << '\b';

}

for (int j = 0; j < len-1; j++) {

cout << ' ';

}

}

int main() {

char l[10];

scanf("%s", l );

int len ;

printf("press 0 to take all\n" );

cin >> len ;

if (len == 0) {

len = strlen(l);

}

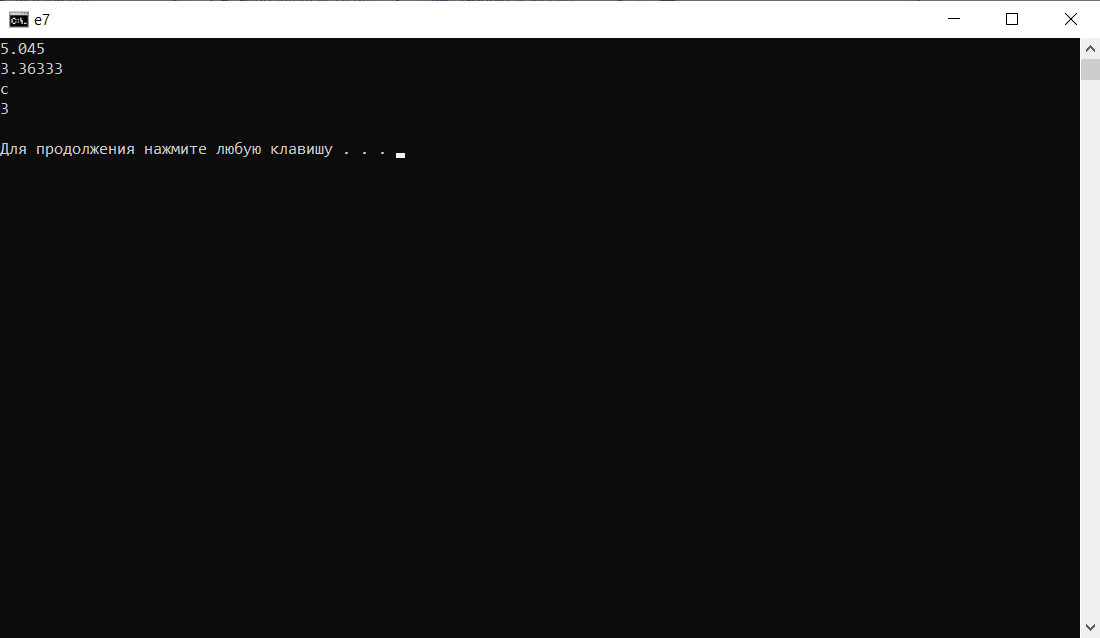
printf("%s", l);

Sleep(2000);

myclreol(len);

}

Завдання 6.7



Код :

#include <iostream>

using namespace std;

int n(int a , int b) {return a+b;}

int n(int a , int &b){return a-b;}

void c (unsigned char c){

cout << c << endl ;

}

void c (char c){

cout << c << endl ;

}

float f (float t ){

return t / 2.0 ;

}

double f (double t){

return t / 3.0 ;

}

int main (){

float x = 10.09;

cout << f(x) << endl ;

double y = 10.09 ;

cout << f(y)<< endl ;

// cout << f (10) << endl ;

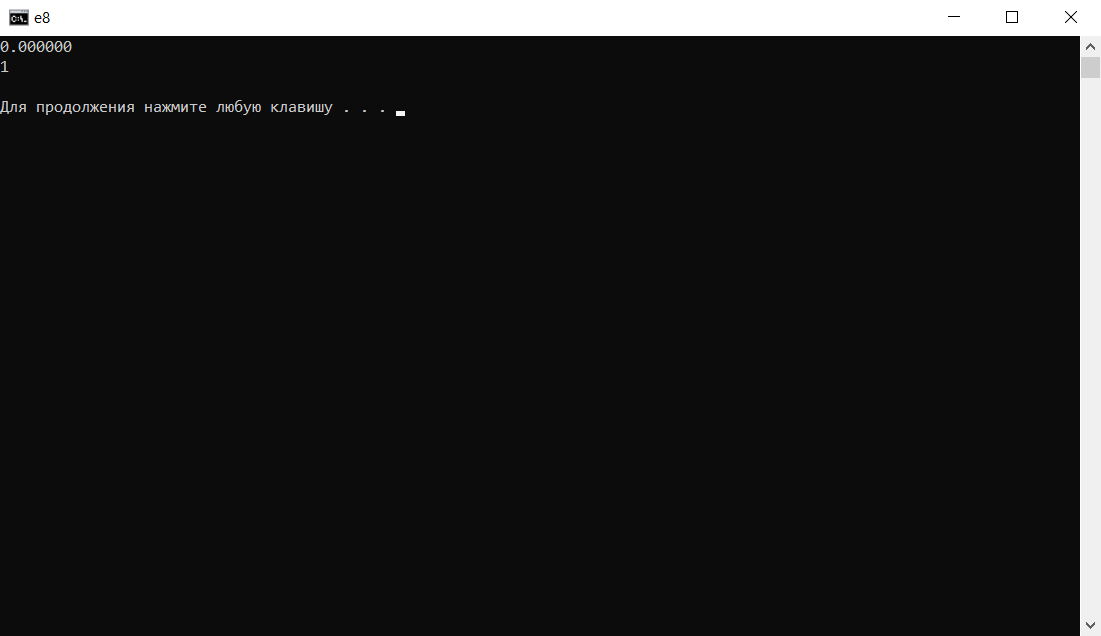
c('c');

// c(5);

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

}

Завдання 6.8



Код :

#include <iostream>

using namespace std;

int dif(int a, int b) {

return a-b;

}

float dif(float a, float b) {

return a-b;

}

int main (){

float (\*f)(float , float );

f = dif ;

int (\*i) (int , int);

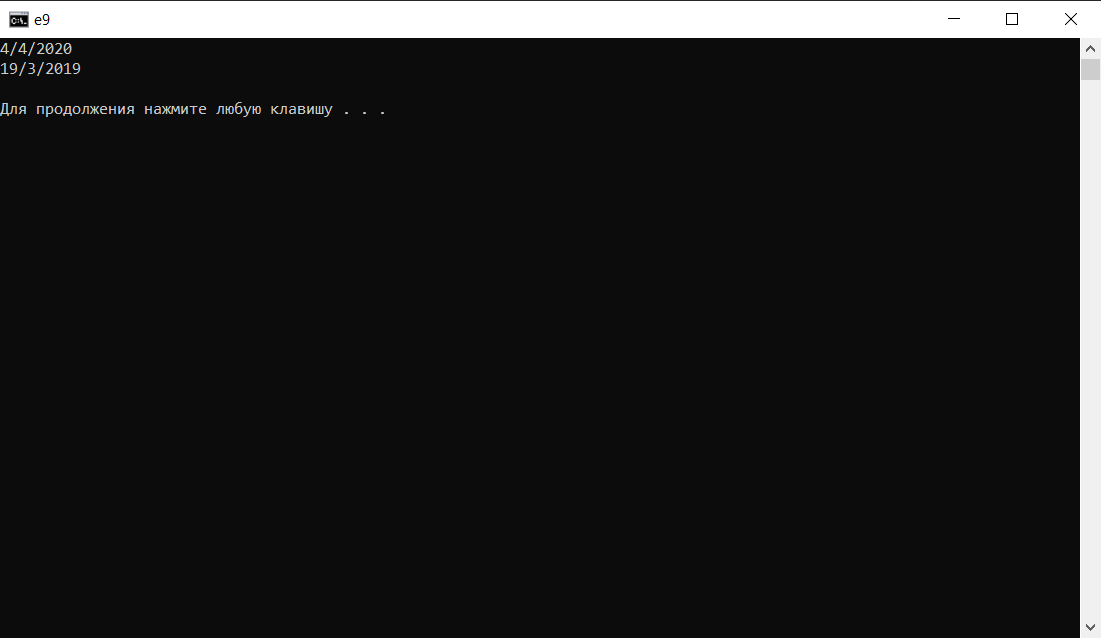
i = dif ;

printf("%f\n", f(1.2 , 1.2) );

printf("%d\n", i(2 , 1) );

}

Завдання 6.9



Код :

#include <iostream>

#include <cstdio>

#include <ctime>

using namespace std;

class Date{

int day, mon, year;

public:

Date(int d, int m, int y){

this->day = d;

this->mon = m;

this->year = y;

}

Date(const char \*str){

sscanf(str, "%d%\*c%d%\*c%d", &day, &mon, &year);

}

Date(time\_t time){

tm\* lt = localtime(&time);

year = (int)(lt->tm\_year + 1900);

mon = (int)(lt->tm\_mon + 1 );

day = (int)(lt->tm\_wday-2);

}

void show(){

cout << day << '/' << mon << '/' << year << '\n';

}

};

int main() {

Date date(time(0));

date.show();

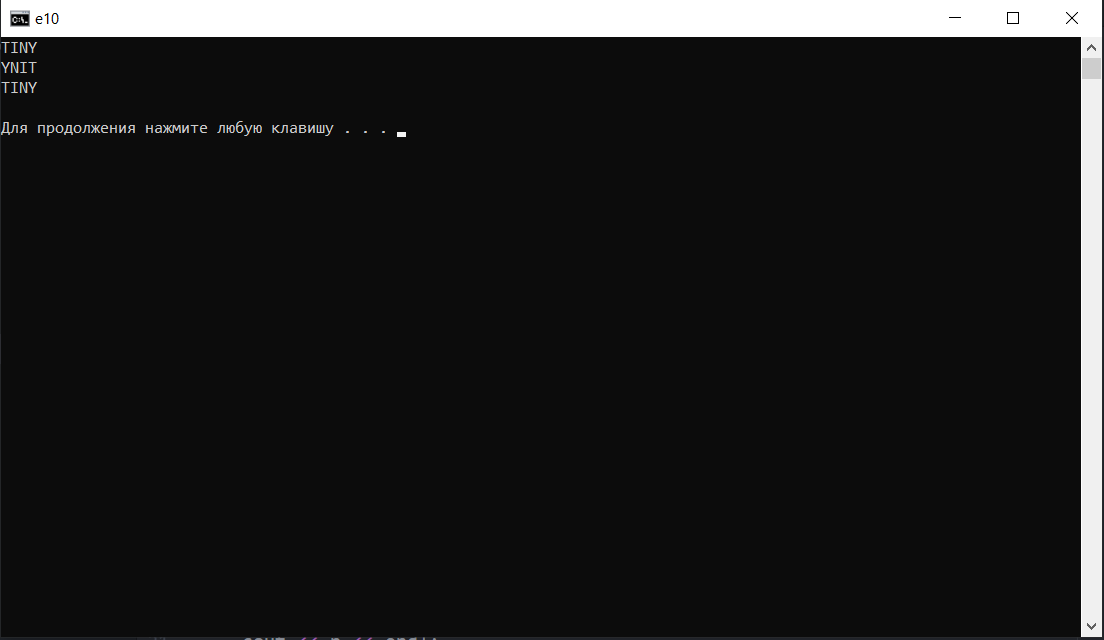
Date date2("19/03/2019");

date2.show();

return 0;

}

Завдання 6.10



Код :

#include <iostream>

#include <cstring>

using namespace std;

void rev(char \*str, int counter = 0){

int j;

if(counter == 0) j = strlen(str) - 1 ;

else j = counter;

char temp;

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

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

int main() {

char \* p = new char[50];

cin >> p ;

rev(p, strlen(p)-1);

cout << p << endl;

rev(p);

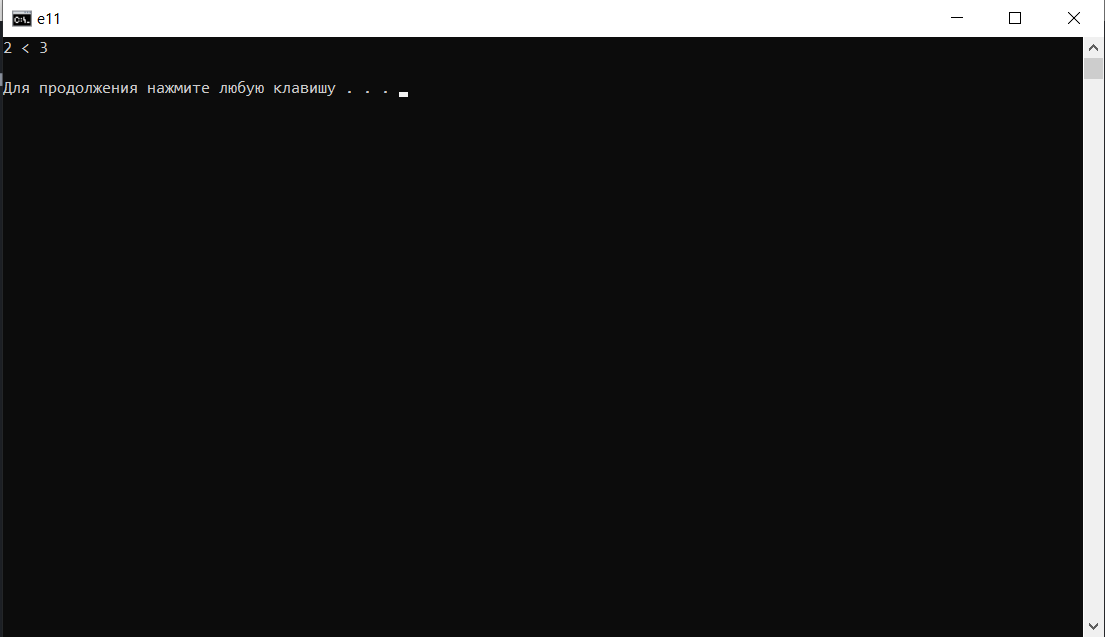
cout << p << endl;

delete [] p;

return 0;

}

Завдання 6.11



Код :

#include <iostream>

using namespace std ;

void order(int &a , int &b ){

int temp;

if(a > b){

temp = a ;

a = b ;

b = temp;

}

}

int main (){

int x = 3;

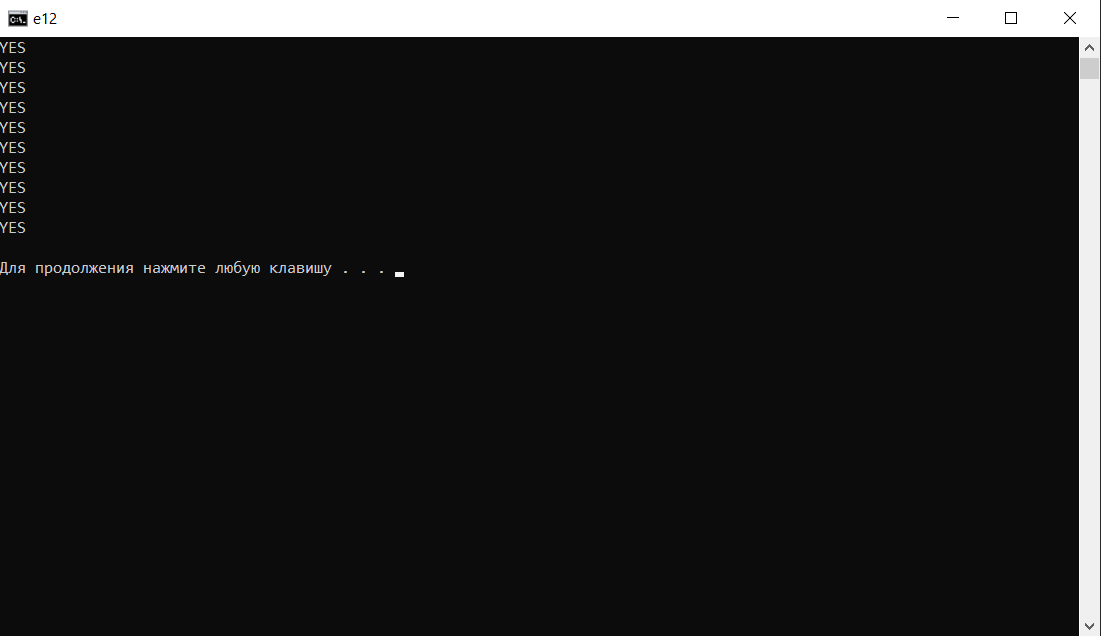
int y = 2 ;

order( x , y );

printf("%d < %d\n", x , y );

}

Завдання 6.12



Код :

#include <iostream>

using namespace std ;

class samp {

int a;

public:

samp(int q){a = q;}

samp(){ if( 2+2 == 4) printf("YES\n" );}

};

int main() {

samp obj(88);

samp objarray[10];

}