**ПРАКТИЧНЕ ЗАВДАННЯ № 9**

**Завдання №1 :**

Создайте иерархию классов для хранения информации о летательных аппаратах. Начните с общего базового класса Airship, предназначенного для хранения количества перевозимых пассажиров и количества перевозимого груза в фунтах или килограммах, или тоннах. Затем создайте два производных от Airship класса Airplane и Balloon (дирижабль). Класс Airplane должен хранить тип самолетного двигателя (винтовой или реактивный) и дальность полета в милях или километрах. Класс Balloon должен хранить информацию о типе газа, используемого для подъема дирижабля (водород или гелий), и его максимальный потолок (в футах или метрах). Создайте программу для демонстрации работы этой иерархии классов.

**Код програми:**

#include <iostream>

using namespace std;

class Airship {

protected:

int passengers;

double baggage;

};

class Airplane : public Airship {

private:

string engine;

int distance;

public:

Airplane(int p, double b, string e, int d) {

passengers = p;

baggage = b;

engine = e;

distance = d;

}

void show() {

cout << "Airplane data: " << endl;

cout << "passengers " << passengers << " baggage " << baggage << endl;

cout << "engine type " << engine << " distance " << distance << "\n\n";

}

};

class Balloon : public Airship {

private:

string gas\_type;

int parachute\_max;

public:

Balloon (int p, double b, string g, int max){

passengers = p;

baggage = b;

gas\_type = g;

parachute\_max = max;

}

void show(){

cout << "Balloon data: " << endl;

cout << "passengers " << passengers << " baggage " << baggage << endl;

cout << "gas\_type " << gas\_type << " parachute\_max " << parachute\_max << endl;

}

};

int main() {

string engine\_type, gas\_type;

cout << "engine type: reactive or propeller-driven" << endl;

cout << "gas type: hydrogen or helium" << endl;

cin >> engine\_type >> gas\_type;

Airplane a(500, 456.345, engine\_type, 345366);

Balloon b(13, 123, gas\_type, 1323);

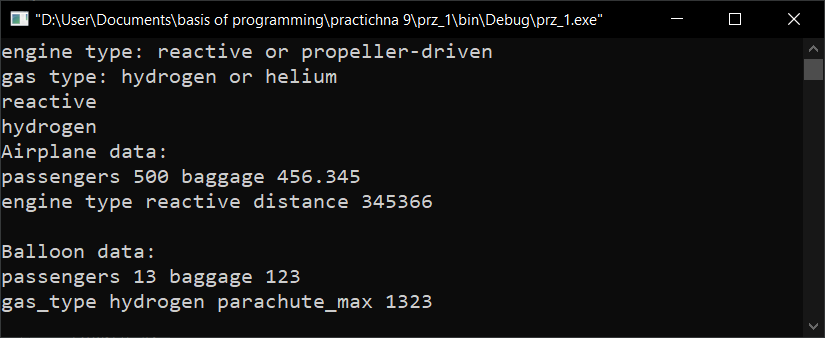
a.show();

b.show();

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №2 :**

Напишите программу, которая бы устанавливала флаги для потока cout так, чтобы целые, если они положительны, выводились со знаком +. Покажите, что ваш набор флагов формата правилен.

**Код програми:**

#include <iostream>

using namespace std;

int main() {

int array[5] = { 1, 3, -5, 7, -9};

cout.setf(ios::showpos);

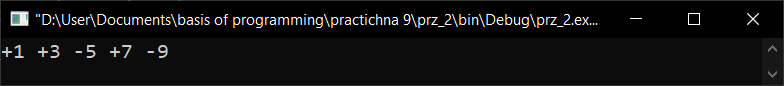
for(int i = 0; i < 5; ++i)

cout << array[i] << ' ';

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №3 :**

Напишите программу, которая бы устанавливала флаги для потока cout так, чтобы всегда при выводе дробных значений были показаны десятичные точки. Кроме этого, значения с плавающей точкой должны выводиться в научной нотации с символом "Е" в верхнем регистре.

**Код програми:**

#include <iostream>

using namespace std;

int main() {

double array[5] = { 10.68, 3.875, -5, 7, -9.89};

cout.setf(ios::showpos | ios:: uppercase | ios:: scientific);

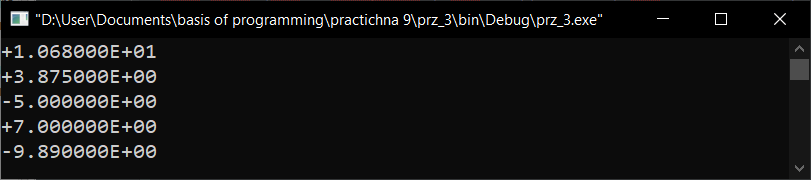
for(int i = 0; i < 5; ++i)

cout << array[i] << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №4 :**

Напишите программу, которая сохраняет текущее состояние флагов формата, устанавливает флаги showbase и hex, выводит на экран значение 100, а затем возвращает флаги в исходное состояние.

**Код програми:**

#include <iostream>

using namespace std;

int main() {

ios::fmtflags flag = cout.flags();

cout.unsetf(ios::dec); cout.setf(ios::hex | ios::showbase);

cout << 100 << endl;

cout.flags(flag);

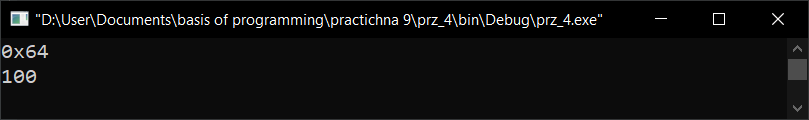
cout << 100 << endl;

return 0;

}

**Код програми:**

**Результати роботи програми (скриншоти)**



**Завдання №5 :**

Разработайте программу для печати таблицы натуральных и десятичных логарифмов чисел от 2 до 100. Формат таблицы следующий: правое выравнивание, ширина поля — 10 символов, точность — 5 десятичных позиций.

**Код програми:**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

cout.precision(5);

for(double i = 2; i < 101; ++i){

cout.width(10);

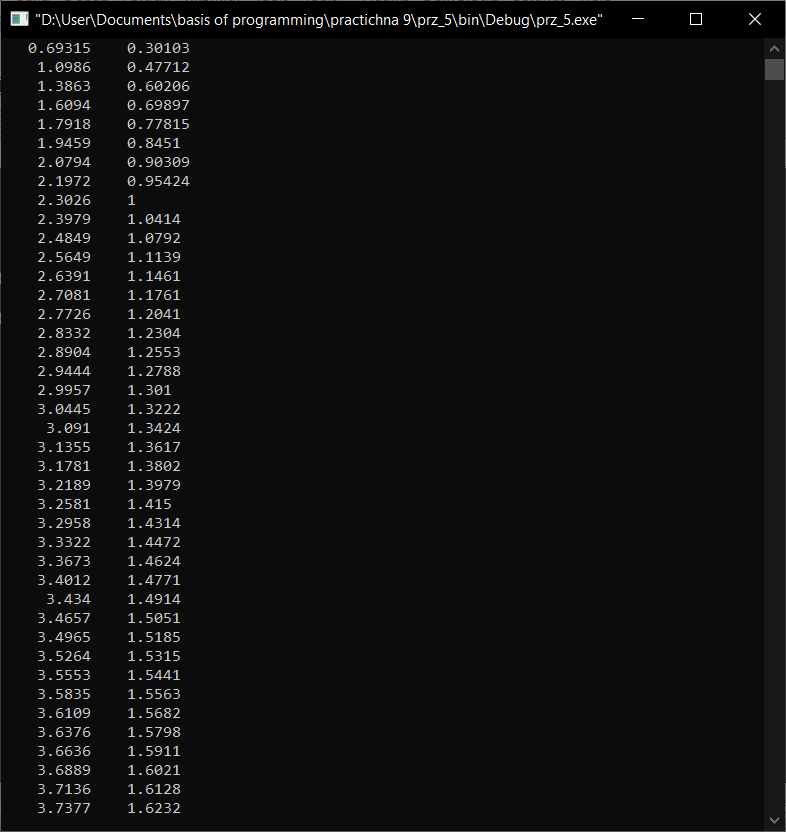
cout << log(i) << " " << log10(i) << endl;

}

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №6 :**

Создайте функцию center() со следующим прототипом:

void center(char \*s);

Эта функция должна устанавливать заданную строку в центр экрана. Для реализации этой задачи воспользуйтесь функцией width(). Предполагается, что ширина экрана равна 80 символам. (Для простоты считайте, что длина строки не превышает 80 символов.) Напишите программу, иллюстрирующую работу этой функции

**Код програми:**

#include <iostream>

#include <cstring>

using namespace std;

void center(char \*str){

cout.width((strlen(str) + 80) / 2);

cout << str << endl;

}

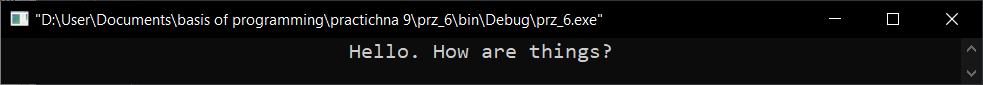
int main() {

center((char \*)"Hello. How are things?");

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №7 :**

Выполните еще раз Задачу 9.5 и 9.6, только теперь, вместо функций-членов и флагов формата, используйте манипуляторы ввода/вывода.

**A)**

**Код програми:**

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int main() {

cout.precision(5);

for(double i = 2; i < 101; ++i){

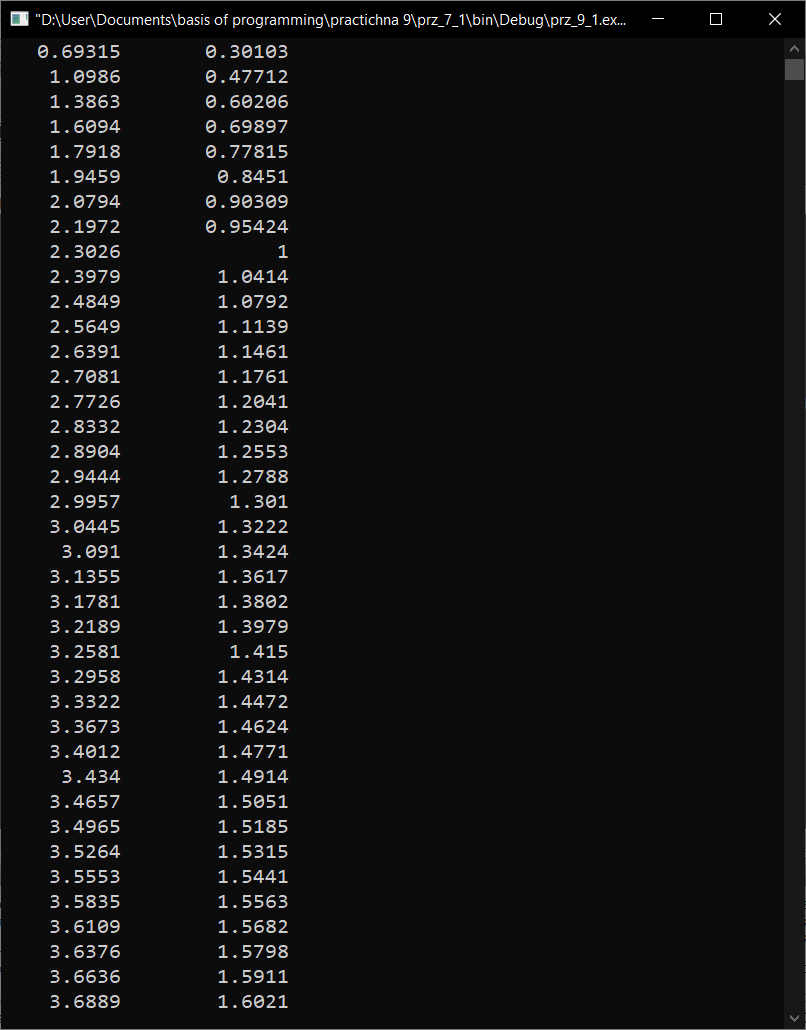
cout << setw(10) << log(i) << " " << setw(10) << log10(i) << endl;

}

return 0;

}

**Результати роботи програми (скриншоти)**



**B)**

**Код програми:**

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

void center(char \*str){

cout << setw((strlen(str) + 80) / 2) << str << endl;

}

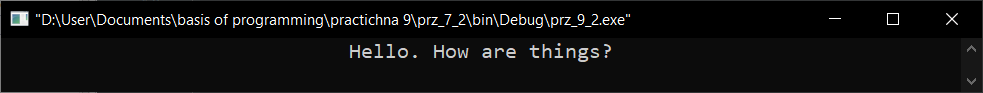
int main() {

center((char \*)"Hello. How are things?");

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №8 :**

Составьте инструкцию для вывода числа 100 в шестнадцатеричной системе счисления с отображением основания системы счисления (0x). Для выполнения задания воспользуйтесь манипулятором setiosflags().

**Код програми:**

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

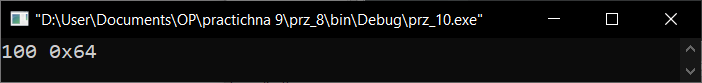
cout << 100 << ' ' << resetiosflags(ios::dec) <<

setiosflags(ios::hex | ios::showbase) << 100 << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №9 :**

В незавершенной программе имеется класс strtype. Для вывода строки на экран создайте пользовательскую функцию вывода:

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class strtype {

char\* p;

int len;

public:

strtype(char\* ptr);

~strtype() { delete[] p; }

friend ostream& operator <<(ostream& stream, strtype& obj);

};

strtype::strtype(char\* ptr) {

len = strlen(ptr) + 1;

p = new char(len);

if (!p) {

cout << "Allocation error\n";

exit(1);

}

strcpy(p, ptr);

}

// Create operator << inserter function here.

int main() {

strtype s1("This is a test."), s2("I hate C++!");

cout << s1 << '\n' << s2;

return 0;

}

**Код програми:**

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class String {

char \*p;

int len;

public:

String(char \*ptr);

~String() { delete [] p; }

friend ostream &operator <<(ostream &stream, String &obj);

friend istream &operator >>(istream &stream, String &obj);

};

String::String(char \*ptr) {

len = strlen(ptr) + 1;

p = new char(len);

if (!p)

exit(1);

strcpy(p, ptr);

}

istream &operator >>(istream &stream, String &obj){

cout << "Enter data" << endl;

char data[100];

stream >> data;

if(strlen(data) >= obj.len){

delete []obj.p;

obj.len = strlen(data) + 1;

obj.p = new char [obj.len];

if(!obj.p)

exit(1);

}

strcpy(obj.p, data);

}

ostream &operator <<(ostream &stream, String &obj){

cout << obj.p << endl;

return stream;

}

int main() {

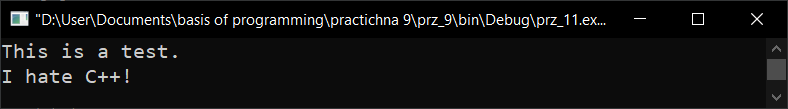
String s1((char \*)"This is a test."), s2((char \*)"I hate C++!");

cout << s1 << s2;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №10 :**

Замените в следующей программе функцию show() пользовательской функцией вывода:

#include <iostream>

using namespace std;

class planet {

protected:

double distance; // расстояние в милях от Солнца

int revolve; // полный оборот в днях

public:

planet(double d, int r) {

distance = d;

revolve = r;

}

};

class earth : public planet {

double circumference; // окружность орбиты

public:

earth(double d, int r) : planet(d, r) {

circumference = 2 \* distance \* 3.1416;

}

/\* Rewrite this so that it displays the information using an inserter function.

\*/

void show() {

cout << "Distance from sum: " << distance << '\n';

cout << "Days in orbit: " << revolve << '\n';

cout << "Circumference of orbit: " << circumference << '\n';

}

};

int main() {

earth obj(93000000, 365);

cout << obj;

return 0;

}

**Код програми:**

#include <iostream>

using namespace std;

class Planet {

protected:

double distance;

int revolve;

public:

Planet(double d, int r) {

distance = d;

revolve = r;

}

};

class Earth : public Planet {

double circumference;

public:

Earth(double d, int r) : Planet(d, r) {

circumference = 2 \* distance \* 3.1416;

}

friend ostream &operator <<(ostream &stream, Earth &obj);

};

ostream &operator <<(ostream &stream, Earth &obj){

cout << "Distance from sum: " << obj.distance << endl;

cout << "Days in orbit: " << obj.revolve << '\n';

cout << "Circumference of orbit: " << obj.circumference << '\n';

return stream;

}

int main() {

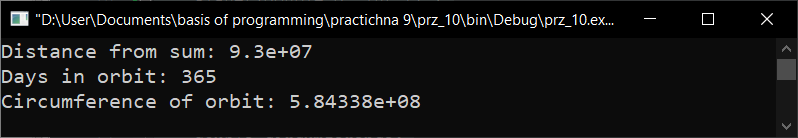
Earth obj(93000000, 365);

cout << obj;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №11 :**

Добавьте пользовательскую функцию ввода в класс strtype Задачи 9.

**Код програми:**

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class String {

char \*p;

int len;

public:

String();

String(const char \*ptr);

~String() {

delete [] p;

}

friend ostream &operator <<(ostream &stream, String &obj);

friend istream &operator >>(istream &stream, String &obj);

};

String::String() {

len = 0;

p = new char[1];

p[0] = '\0';

}

String::String(const char \*ptr) {

len = strlen(ptr) + 1;

p = new char(len);

if (!p)

exit(1);

strcpy(p, ptr);

}

istream &operator >>(istream &stream, String &obj) {

cout << "Enter string:" << endl;

char data[100];

stream >> data;

int size\_data = strlen(data) + 1;

if(obj.len < size\_data) {

delete []obj.p;

obj.p = new char [size\_data];

obj.len = size\_data;

if(!obj.p)

exit(1);

}

strcpy(obj.p, data);

return stream;

}

ostream &operator <<(ostream &stream, String &obj) {

cout << "Entered string:" << endl << obj.p << endl;

return stream;

}

int main() {

String s1;

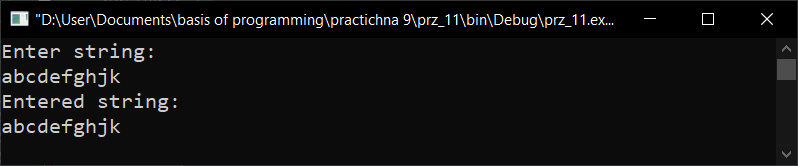
cin >> s1;

cout << s1;

return 0;

}

**Результати роботи програми (скриншоти):**



**Завдання №12 :**

Создайте класс для хранения целого и его наименьшего делителя. Создайте для этого класса пользовательские функции ввода и вывода.

**Код програми:**

#include <iostream>

using namespace std;

class Integer {

private:

int num;

int divisor;

public:

Integer () { num = 0; divisor = 0; }

friend istream &operator >> (istream &stream, Integer &obj);

friend ostream &operator << (ostream &stream, Integer &obj);

};

istream &operator >> (istream &stream, Integer &obj){

cout << "Enter integer" << endl;

stream >> obj.num;

if (obj.num == 0){

cout << "divisor of 0: " << obj.num << "\*b, b!= 0" << endl;

exit(1);

}

else if(obj.num % 2 == 0)

obj.divisor = 2;

else

obj.divisor = obj.num;

return stream;

}

ostream &operator << (ostream &stream, Integer &obj){

cout << "Integer: " << obj.num << ','

<< " its the smallest divisor: " << obj.divisor << endl;

return stream;

}

int main()

{

Integer a;

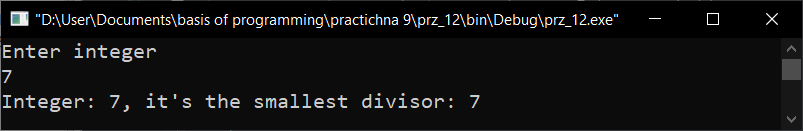
cin >> a;

cout << a;

return 0;

}

**Результати роботи програми (скриншоти):**



**Завдання №13 :**

Напишите программу для вывода числа 100 в десятичной, шестнадцатеричной и восьмеричной системе счисления. (Используйте флаги формата класса ios.)

**Код програми:**

#include <iostream>

using namespace std;

int main () {

cout << "dec: " << 100 << endl;

cout.unsetf (ios::dec);

cout.setf(ios::hex);

cout << "hex: " << 100 << endl;

cout.unsetf(ios::hex);

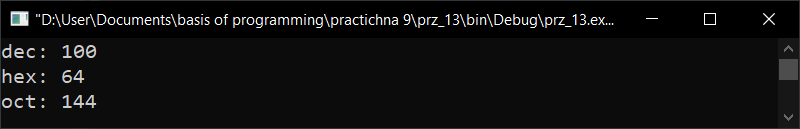
cout.setf(ios::oct);

cout << "oct: " << 100 << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №14 :**

Напишите программу для вывода значения 1000.5364 в 20-символьном поле, с выравниванием влево, с двумя позициями после запятой и символом \* в качестве символа заполнения. (Используйте флаги формата класса ios.)

**Код програми:**

#include <iostream>

using namespace std;

int main()

{

cout.width(20);

cout.setf(ios::left);

cout.precision(2);

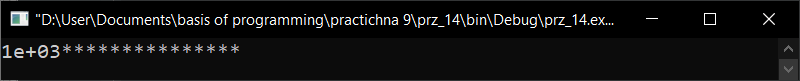
cout.fill('\*');

cout << 1000.5354 << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №15 :**

Перепишите ваши ответы на Задачи 13 и 14 так, чтобы использовались манипуляторы ввода/вывода.

**Код програми:**

#include <iostream>

#include <iomanip>

using namespace std;

int main () {

cout << 100 << ' ' << hex << 100 << ' '<< oct << 100 << endl;

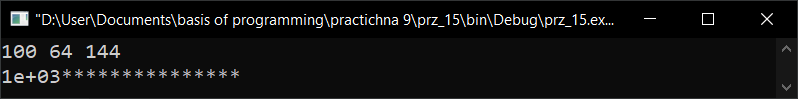
cout << setw(20) << setiosflags(ios::left) <<

setprecision(2) << setfill('\*') << 1000.5354 << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №16 :**

Покажите, как записать и как отобразить для класса cout флаги формата. Используйте функции-члены либо манипуляторы.

**Код програми:**

#include <iostream>

using namespace std;

void show();

int main() {

cout << "Some number here " << 12773.8676 << endl;

cout.setf(ios:: hex | ios:: uppercase | ios:: showpos| ios:: left);

cout << "Some number here " << 12773.8676 << endl;

show();

return 0;

}

void show() {

ios::fmtflags state = cout.flags();//getting state of set flags

if(state & ios::skipws)

cout << "skipws set" << endl;

if(state & ios::left)

cout << "left set" << endl;

if(state & ios::right)

cout << "right set" << endl;

if(state & ios::dec)

cout << "dec set" << endl;

if(state & ios::hex)

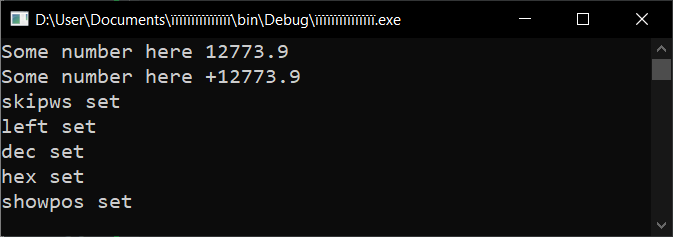
cout << "hex set" << endl;

if(state & ios::showpos)

cout << "showpos set" << endl;

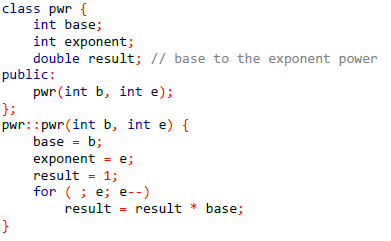
}

**Результати роботи програми (скриншоти)**



**Завдання №17 :**

Создайте для следующего класса пользовательские функции ввода и вывода:



**Код програми:**

#include <iostream>

using namespace std;

class Power {

int base;

int exponent;

double result; // base to the exponent power

public:

Power(int b, int e);

friend istream &operator >>(istream &stream, Power &obj);

friend ostream &operator << (ostream &stream, Power &obj);

};

Power::Power(int b, int e) {

base = b;

exponent = e;

result = 1;

for ( ; e; e--)

result = result \* base;

}

istream &operator >> (istream &stream, Power &obj){

int b, e;

stream >> b >> e;

Power new\_obj(b, e);

obj = new\_obj;

return stream;

}

ostream &operator << (ostream &stream, Power &obj){

cout << "Base: " << obj.base << ", result: " << obj.result << endl;

return stream;

}

int main() {

Power a(2, 3);

cout << a;

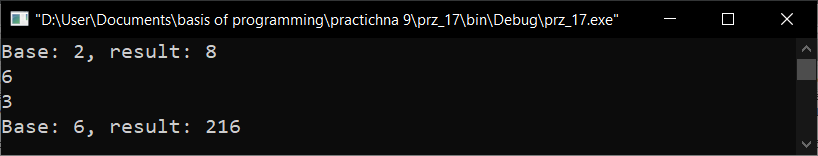
cin >> a;

cout << a;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №18 :**

Создайте класс box для хранения размеров квадрата. Для вывода изображения квадрата на экран создайте пользовательскую функцию вывода. (Способ изображения выберите любой.)

**Код програми:**

#include <iostream>

using namespace std;

class Box {

private:

int size;

public:

Box() { size = 1; }

Box(int s) { size = s; }

friend istream &operator >> (istream &stream, Box &obj);

friend ostream &operator << (ostream &stream, Box &obj);

};

istream &operator >> (istream &stream, Box &obj){

cout << "Enter square size:" << endl;

int s; stream >> s;

obj.size = s;

return stream ;

}

//ostream &operator << (ostream &stream, Box &obj){

// stream << "Your square:" << endl;

// for (int i = 0; i < obj.size; ++i)

// stream << "\* ";

// stream << endl;

// for(int i = 1; i < obj.size - 1; ++i){

// stream << "\* ";

// for(int j = 1; j < obj.size - 1; ++j)

// stream << " ";

// stream << "\* " << endl;

// }

// for(int i = 0; i < obj.size; ++i)

// stream << "\* ";

//return stream;

//}

ostream &operator << (ostream &stream, Box &obj){

stream << "Your square:" << endl;

for(int i = 0; i < obj.size; ++i){

stream << "\* ";

for(int j = 0; j < obj.size - 1; ++j)

stream << "\* ";

stream << endl;

}

return stream;

}

int main()

{

Box a;

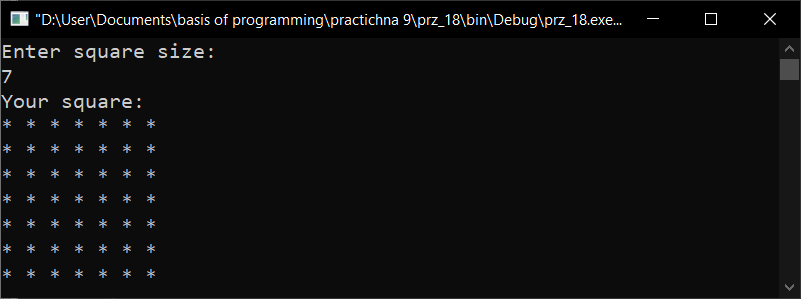
cin >> a;

cout << a;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №19 :**

Используя показанный здесь класс stack, создайте пользовательскую функцию вывода для вставки в поток содержимого стека. Покажите, что функция работает.

#include <iostream >

using namespace std;

#define SIZE 10

class stack {

char stck[SIZE]; // holds the stack

int tos; // index of top -of - stack

public:

stack();

void push(char ch); // push character on stack

char pop(); // pop character from stack

};

// Initialize the stack:

stack::stack() {

tos = 0;

}

// Push a character

void stack::push(char ch) {

if (tos == SIZE) {

cout << "Stack is full\n";

return 0;

}

stck[tos] = ch;

tos++;

}

// Pop a character:

char stack::pop() {

if (tos == 0) {

cout << "Stack is empty\n";

return 0;

}

tos--;

return stck[tos];

}

**Код програми:**

#include <iostream>

#define SIZE 10

using namespace std;

class stack {

char stck[SIZE]; // holds the stack

int tos; // index of top -of - stack

public:

stack(){ tos = 0; }

void push(char ch); // push character on stack

char pop(); // pop character from stack

friend ostream &operator << (ostream &stream, stack &obj);

// friend istream &operator >> (istream &stream, stack &obj);

};

void stack::push (char ch) {

if (tos == SIZE)

return;

stck[tos] = ch;

++tos;

}

char stack::pop() {

if (tos==0)

return 0;

--tos;

return stck[tos];

}

ostream &operator << (ostream &stream, stack &obj){

if (obj.tos == 0){

stream << "array stack is empty"<< endl;

exit(1);

}

else {

stream << "Stack: " << endl;

while(char i = obj.pop())

stream << i;

stream << endl;

}

return stream;

}

int main()

{

stack a;

a.push('a');

a.push('b');

a.push('c');

a.push('d');

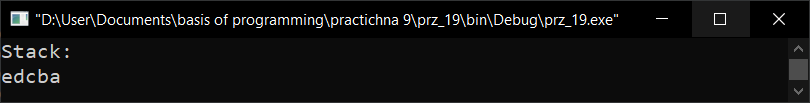
a.push('e');

cout << a;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №20:**

Напишите программу с классом watch, который бы играл роль обычных часов. Используя стандартные функции времени, создайте конструктор класса, который должен считывать и запоминать системное время. Для вывода этого времени на экран создайте пользовательскую функцию вывода.

**Код програми:**

#include <iostream>

#include <ctime>

using namespace std;

class Watch {

private:

time\_t timer;

public:

Watch (){

timer = time(NULL);

}

friend ostream &operator << (ostream &stream, Watch &obj);

};

ostream &operator << (ostream &stream, Watch &obj){

stream << asctime(localtime(&obj.timer)) << endl;

return stream;

}

int main()

{

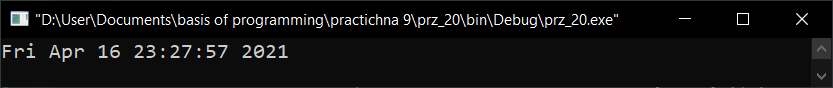
Watch a;

cout << a;

return 0;

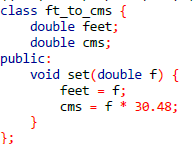
}

**Результати роботи програми (скриншоти)**



**Завдання №21 :**

На основе класса, созданного для преобразования футов в сантиметры, напишите пользовательскую функцию ввода, формирующую строку-приглашение для записи числа футов. Кроме этого, напишите пользовательскую функцию вывода для отображения на экране как числа футов, так и числа сантиметров. Включите указанные функции в программу и продемонстрируйте их работоспособность.



**Код програми:**

#include <iostream>

using namespace std;

class ft\_to\_cms {

double feet;

double cms;

public:

void set(double f) {

feet = f;

cms = f \* 30.48;

}

friend istream &operator >>(istream &stream, ft\_to\_cms &obj);

friend ostream &operator << (ostream &stream, ft\_to\_cms &obj);

};

istream &operator >> (istream &stream, ft\_to\_cms &obj){

cout << "Enter feets:" << endl;

double data;

stream >> data;

obj.set(data);

return stream;

}

ostream &operator << (ostream &stream, ft\_to\_cms &obj){

cout << "Feets: " << obj.feet << ", cms: " << obj.cms;

return stream;

}

int main() {

ft\_to\_cms a;

cin >> a;

cout << a;

return 0;

}

**Результати роботи програми (скриншоти)**

