**Лабораторна робота №2**

Тема: Робота з лінійними списками. Конструктор і деструктор класу Мета: Навчитись використовувати конструктори і деструктори класів, створювати класи для опису лінійних списків

**ПОРЯДОК ВИКОНАННЯ РОБОТИ 1**

Реалізувати один простий клас згідно варіанту індивідуального завдання, що містить закриті данні, а саме два типа даних: числове та рядкове, реалізоване через покажчик на char. Потрібно створити декілька екземплярів класу статично і динамічно, а також масив та продемонструвати дію всіх конструкторів і методів. Реалізувати методи:  Scan - ввід даних з клавіатури у поля класу;  Print - констатний метод виводу даних на екран;  Конструктор по замовчуванню;  Конструктор ініціалізації Клас(сhar\*, int);  Конструктор копіювання Клас(const Клас&);  Деструктор  Методи доступу та закритих данних Get та Set.

**Код програми:**

//ЗАВДАННЯ 1

//Реалізувати один простий клас згідно варіанту індивідуального завдання, що містить закриті данні,

//а саме два типа даних : числове та рядкове, реалізоване через покажчик на char.

//Потрібно створити декілька екземплярів класу статично і динамічно, а також масив та продемонструвати дію всіх конструкторів і методів.

//Реалізувати методи :

// Scan - ввід даних з клавіатури у поля класу;

// Print - констатний метод виводу даних на екран;

// Конструктор по замовчуванню;

// Конструктор ініціалізації Клас(сhar\*, int);

// Конструктор копіювання Клас(const Клас&);

// Деструктор

// Методи доступу та закритих данних Get та Set.

#include "stdafx.h"

#include<iostream>

#include<math.h>

using namespace std;

struct Node

{

int x;

Node \*Next, \*Prev;

};

class Computer

{

char \*Owner = new char[40];

int Processor;

public:

Computer(char\* owner, int processor) { Owner = owner; Processor = processor; };

Computer() :Computer((char\*)("Other"), 0) {};

Computer & operator=(const Computer & Computer){}

void SetOwner(const char\*);

char\*GetOwner();

void SetProcessor(int);

int GetProcessor();

void Print();

void Input();

~Computer() { cout << "delete" << endl;};

};

void Computer::SetOwner( const char\* owner)

{

Owner = new char[40];

for (int i = 0; i < 40; i++)

Owner[i] = owner[i];

}

char\* Computer ::GetOwner(){ return Owner;}

void Computer::SetProcessor(int n) {Processor = n;}

int Computer::GetProcessor() {return Processor;}

void Computer::Print()

{

cout << "Processor " << Processor << endl;

cout << "Owner " << Owner << endl<<endl;

}

void Computer::Input()

{

cout << "Write processor ";

cin >> Processor;

cout << "Write owner ";

char \* str = new char[40];

cin >> str;

SetOwner(str);

cout << endl;

}

int main()

{

Computer PC\_1;

Computer PC\_2((char\*)("AMD"), 21);

Computer \*PC\_3 = new Computer;

Computer \*PC\_4 = new Computer((char\*)"AMD", 21);

PC\_1.Input();

PC\_2.Input();

PC\_3->Input();

PC\_4->Input();

PC\_1.Print();

PC\_2.Print();

PC\_3->Print();

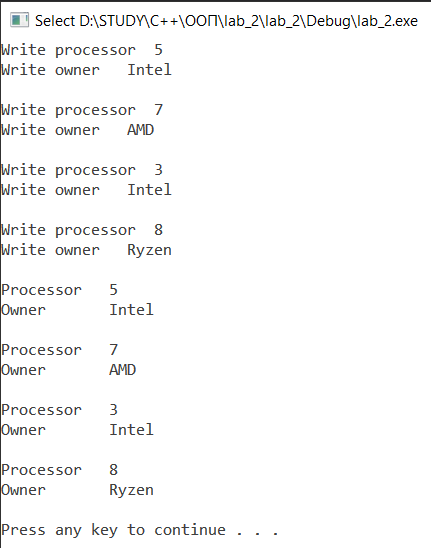
PC\_4->Print();

system("pause");

return 0;

}

**Результат виконання:**

****

**2 Виконати завдання згідно варіанту індивідуального завдання:**

Створити двозв’язний список та заповнити його n-ою кількістю простих чисел.