

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни

«Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 27

Виконав студент ІП-15, Пономаренко Маргарита Альбертівна  
(шифр, прізвище, ім'я, по батькові)

Перевірів \_\_\_\_\_  
( прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота 6

### Дослідження рекурсивних алгоритмів

**Мета** - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

### Індивідуальне завдання

#### Варіант 27

#### Постановка задачі

Перетворення значення у двійковій системі числення в шістнадцятирічне значення.

#### Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Значення num	Натуральне	num	Вхідні дані
Переведення з двоїчної системи числення в шістнадцяткову	Функція	hex_num	Проміжні дані
Переведення числових представлень у буквенні	Функція	convert	Проміжні дані
Значення локальної змінної в convert()	Натуральне	res	Результат

Для виконання даного завдання будуть використовуватися дві підпрограми. Підпрограма `hex-num` – шляхом цілочисельного та ділення з остачею заданого числа `num` на 16 буде знаходити числові представлення знаків шістнадцяткової системи. Підпрограма `convert()` при умові, якщо `res > 9`, буде переводити числові представлення у буквенні, тобто переводити значення отримані першою підпрограмою “10”, “11”, “12”, “13”, “14”, “15” у їх шістнадцяткові представлення: “A”, “B”, “C”, “D”, “E”, “F”.

В програмі будуть використовуватися функції цілочисельного ділення, що позначатимуться “ / ” та функція знаходження остачі від ділення - “ % ”. Також буде застосований принцип порівняння двох змінних або значень “ == ”.

### Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Переведення `num` із двоїчної системи числення в шістнадцяткову за допомогою функції `hex_num`.

Крок 3. Перевірка значення `res` та конвертація числових значень у буквенні за допомогою функції `convert ()`.

### Псевдокод

*Крок 1*

**початок**

**ввід** `num`:

Переведення `num` із двоїчної системи числення в шістнадцяткову за допомогою функції `hex_num`.

**кінець**

*Крок 2*

**початок**

**ввід** num:

hex\_num ( num )

**кінець**

*Підпрограма 1:*

hex\_num ( num )

**початок**

**якщо** num / 1000 == 0

**то**

hex\_num ( num / 16 )

convert ( num % 16 )

**повернути** ( num mod 16 )

**інакше**

convert ( num )

**повернути** ( num )

**кінець**

*Підпрограма 2:*

**convert** ( res )

**початок**

**якщо** res == 10 **то**

виведення “ A ”

**інакше**

**якщо** res == 11 **то**

виведення “ B ”

**інакше**

**якщо** res == 12 **то**

виведення “ C ”

**інакше**

**якщо** res == 13 **то**

виведення “ D ”

**інакше**

**якщо** res == 14 **то**

виведення “ E ”

**інакше**

**якщо** res == 15 **то**

виведення “ F ”

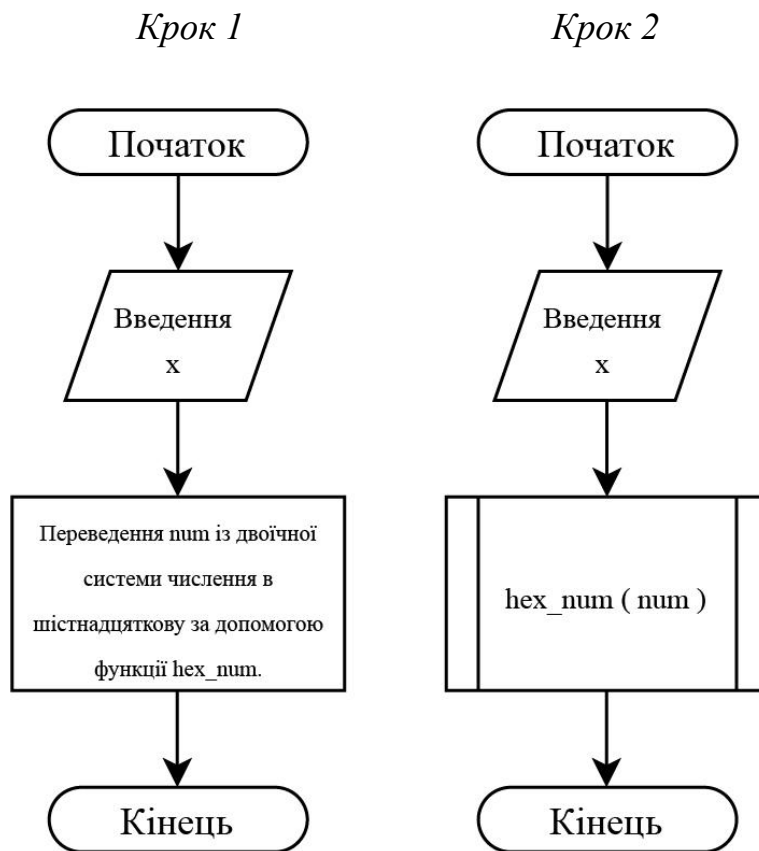
**інакше**

виведення res

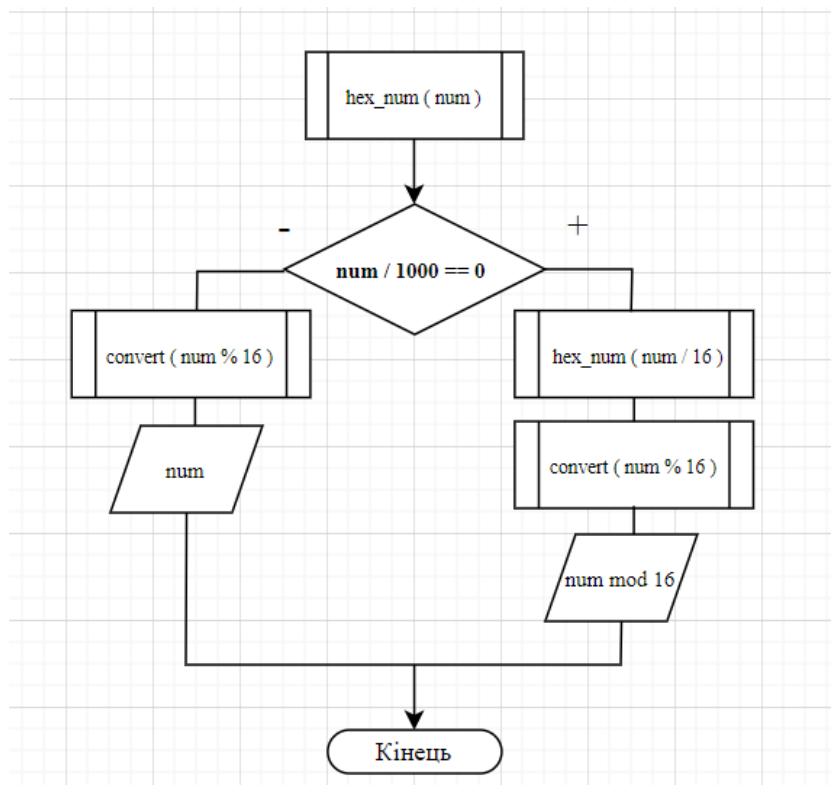
**кінець**

## Блок схема

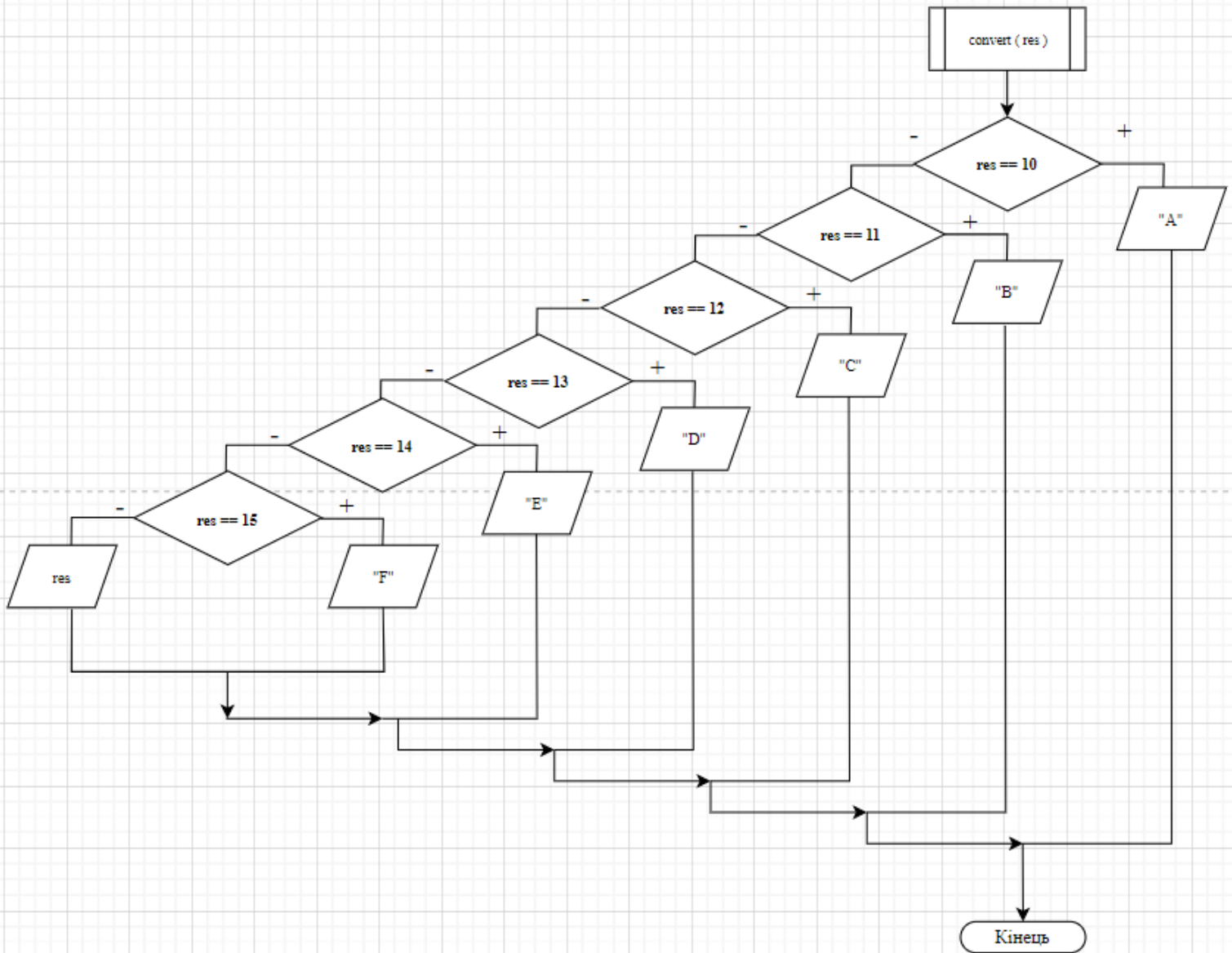
Основна програма:



Підпрограма 1:



**Підпрограма 2:**



*Kod:*

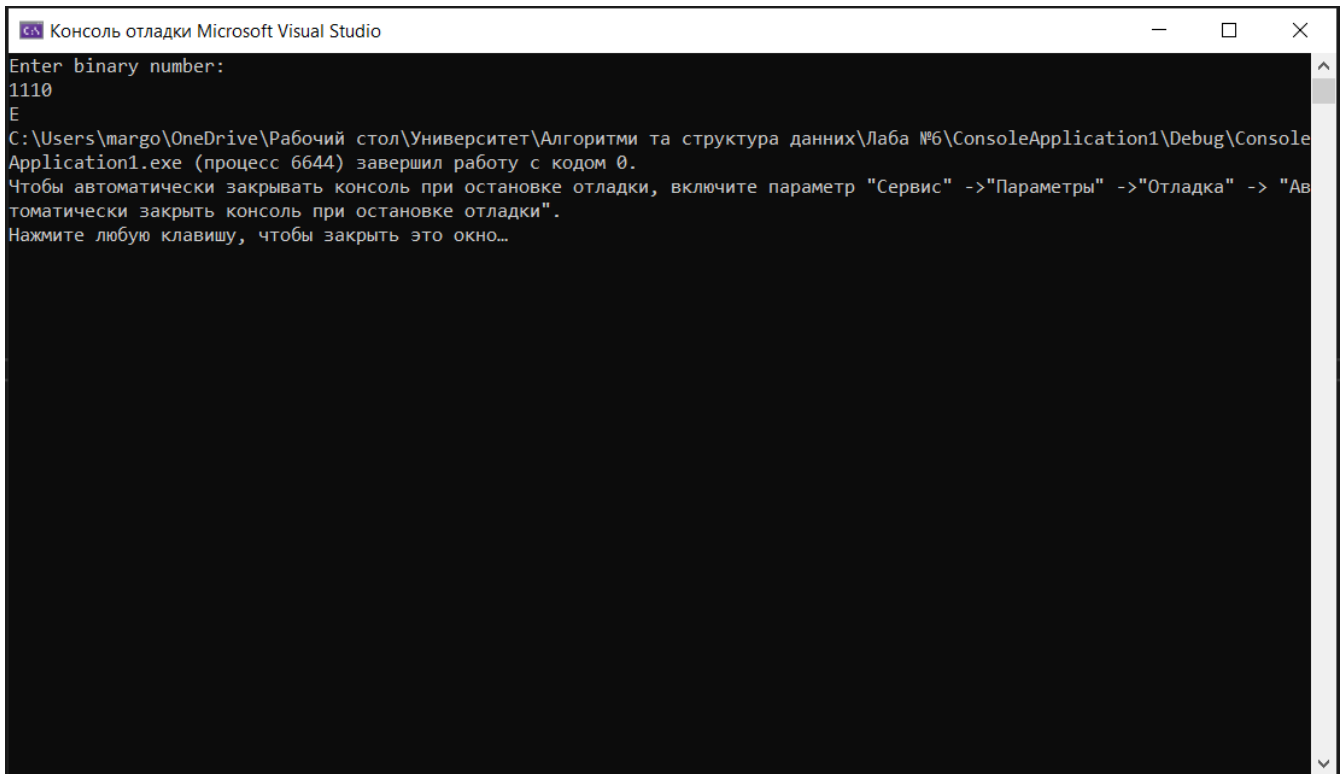
```

1  #include<iostream>
2  using namespace std;
3  void convert(int res)
4  {
5      if (res == 10) {
6          cout << "A";
7      }
8      else if (res == 11) {
9          cout << "B"; }
10     else if (res == 12) {
11         cout << "C"; }
12     else if (res == 13) {
13         cout << "D"; }
14     else if (res == 14) {
15         cout << "E"; }
16     else if (res == 15) {
17         cout << "F"; }
18     else {
19         cout << res; }
20 }
21 int hex_num(int num)
22 {
23     int hx_nm;
24     if (num / 1000 == 0) {
25         convert(num % 16);
26         return num % 16;
27     }
28     else {
29         convert(num);
30         return num;
31     }
32 }
33 int main()
34 {
35     int num;
36     cin >> num;
37     hex_num(num);
38 }

```



### *Тестування:*



```
Консоль отладки Microsoft Visual Studio
Enter binary number:
1110
E
C:\Users\margo\OneDrive\Рабочий стол\Университет\Алгоритми та структура даних\Лаба №6\ConsoleApplication1\Debug\ConsoleApplication1.exe (процесс 6644) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

### **Висновки**

Ми дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм. В результаті виконання лабораторної роботи вдалося створити алгоритм, що перетворює значення з двійкової системи числення в шістнадцятирічне значення.