 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №8**

з дисципліни «Алгоритмізація та програмування»

на тему: «Рекурсія»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2018

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

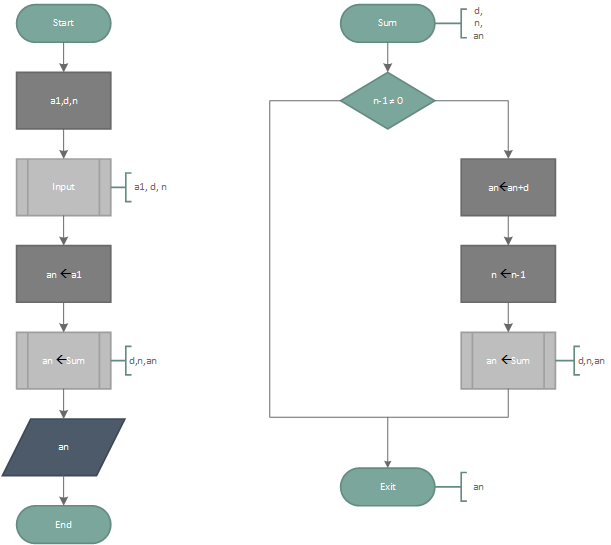
**Завдання:**

1. Ознайомитися з теоретичними відомостями створення функцій користувача.
2. Побудувати блок-схему алгоритму для вирішення задання відповідно до свого варіанту.
3. Розробити програмний застосунок, що включає реалізацію функції для обробки беззнакового цілого числа за допомогою бітових операцій відповідно до свого варіанту(глобальні змінні не використовувати, перевіряти на правильність введення даних користувача.):

*Написати рекурсивну функцію знаходження* ***n****–го члена арифметичної прогресії за першим членом прогресії, різницею прогресії та номером члена прогресії* ***n****.*

1. Скласти та захистити звіт по роботі.

**Блок-схеми програми та функції:**

****

**Лістинг:**

#include <iostream>

#include <conio.h>

using namespace std;

float Sum(float d, int n, float an)

{

if (n-1)

{

an = an + d;

n--;

an = Sum(d, n, an);

}

return an;

}

void Input(float &a1, float &d, int &n)

{

cout << "a1 = ";

cin >> a1;

cout << "d = ";

cin >> d;

cout << "n = ";

cin >> n;

}

int main()

{

float a1;

float d;

int n;

Input(a1, d, n);

float an = a1;

an = Sum(d, n, a1);

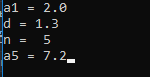
cout << "a" << n << " = " << an;

\_getch();

return 0;

}

**Результати:**

****

**Контрольні питання:**

1. Дайте визначення рекурсії та наведіть приклади її використання в різних галузях знань.

У програмуванні рекурсія — виклик функції чи процедури з неї самої (звичайно з іншими значеннями вхідних параметрів) безпосередньо чи через інші функції (наприклад, функція А викликає функцію B, а функція B — функцію A).

Приклади її використання: з рекурсією тісно пов'язана математична індукція, числа Фібоначчі визначаються за допомогою рекурсії, практично всі геометричні фрактали задаються у формі нескінченної рекурсії, класичним прикладом нескінченної рекурсії є два поставлені одне проти одного дзеркала.

1. В яких випадках доцільно використовувати рекурсивні функції?

Рекурсія використовується, коли можна виділити самоподібність задачі(коли наступне значення величини обчислюється через попереднє.).

1. Що спільного та в чому полягає різниця між циклічними та рекурсивними способами організації розрахунків?

Спільним є те що, ці методи дозволяють обчислити задачі з однаковим завданням , але різними даними.

Різниця полягає в тому, що в рекурсії програма викликає саму себе, тобто запису функції є компактнішим, також рекурсивний метод займає більше часу та пам’яті на повторні виклики функції і передачу параметрів.

1. Визначити основні особливості роботи рекурсивних функцій в мові програмування С++.

Особливістю роботи рекурсивних функцій в мові С++ є те, що оператори, записані перед рекурсивним викликом, виконуються в тому ж порядку, в якому відбуваються виклики функцій; оператори, розташовані після рекурсивного виклику, виконуються в зворотному порядку відносно рекурсивних викликів даної функції.

Надто велика кількість рекурсивних викликів функції може призвести до переповнення стека. Оскільки місцем зберігання параметрів і локальних змінних функції є стек, і кожен новий виклик створює нову копію змінних, простір стека може бути вичерпаний, це викличе помилку переповнення і призведе до аварійної зупинки програми.

1. Що таке пряма та непряма (опосередкована) рекурсії?

Рекурсія буває двох типів: пряма, коли функція викликає сама себе, та непряма (опосередкована), коли функція викликає іншу функцію, а та, в свою чергу, потім викликає першу.

1. Коли виконується завершення рекурсивних викликів? Що таке рекурсивне зациклювання та до чого воно призводить?

Завершення рекурсивних викликів відбувається тоді, коли досягнута умова повернення функцією значення. Якщо в тілі функції такої умови не вказано, або вона вказана неадекватно, то виникає ситуація рекурсивного зациклення – необмеженого самозвернення функції, що може призвести до переповнення сегмента стека і неможливості створення наступних примірників локальних об’єктів функції, що призводить до аварійного завершення роботи програми.

1. Що таке область дії змінних?

Область дії змінних – це частина програми, у якій ці змінні можна використовувати для операцій.

1. Як необхідно виконати об’яву змінної, щоб доступ до неї був можливим з будь-якої функції поточного файлу програми?

Необхідно оголосити змінну поза межами всіх функцій – тобто створити глобальну змінну, яку можна використовувати у будь-якому місці програми, починаючи від точки її оголошення, і доступ до неї буде можливим з будь-якої функції поточного файлу програми.

1. Як змінюються значення локальних та глобальних змінних при використанні рекурсивних функцій?

При використанні рекурсивних функцій глобальні та зовнішні змінні не змінюються. А локальні змінні та інші автоматичні змінні, що об’явлені у функції, можуть бути змінені. Таким чином, змінні, що розташовані поза функцією виклику, не змінюються.

1. Визначте головне обмеження при використанні рекурсії.

Головним обмеженням при використанні рекурсії є глибина рекурсії (кількість викликів). А допустима кількість викликів визначається доступним для використання об’ємом пам’яті ЕОМ.