 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №2**

з дисципліни «Алгоритмізація та програмування»

на тему: «Блок-схеми алгоритмів виконання програм»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

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

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

(підпис викладача)

Київ-2019

**Завдання:**

1. Вивчити теоретичні основи опису алгоритмів за допомогою блок-схем.
2. Відповідно до свого варіанту побудувати блок-схему алгоритму вирішення задачі:

***Побудувати блок-схему алгоритму для визначення номера трикутника з найбільшою гіпотенузою при умові, що для двох прямокутників задано їх катети, які дорівнюють A₁, A₂, B₁, B₂, C₁, C₂ відповідно.***

***Додаткове завдання\*.*** *Організувати додатковий блок в схемі для виконання перевірки на правильність введення даних користувачем. За розробленою схемою перехід до блоку виконання завдання має відбуватися лише за умови введення коректних даних та передбачувати можливість багаторазового введення хибних даних*

1. Скласти і захистити звіт по роботі.

**🞏 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:**

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

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

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

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

**Блок-схема:**



**Контрольні питання:**

1. Що таке алгоритм?

Алгоритм – це послідовність команд, призначена виконавцю, у результаті виконання якої він повинен вирішити поставлене завдання.

1. Які існують способи опису алгоритмів?

Існують такі способи подання алгоритмів:

* словесний опис;
* графічний опис алгоритму у вигляді блок-схеми;
* опис алгоритму за допомогою математичних формул;
* опис алгоритму за допомогою псевдокоду;
* комбінований спосіб зображення алгоритму з використанням словесного, графічного і ін. способів.

1. Використовуючи словесний запис опису алгоритмів, скласти алгоритм переходу вулиці на світлофор.
   1. Підійти до вулиці.
   2. Подивитись на світлофор:
   3. Червоний колір – повернутись на пункт 2.
   4. Зелений колір – подивитись наліво та направо, якщо машин немає чи вони зупинились – перейти вулицю.
2. Перелічити основні графічні елементи блок-схем.
   1. Блок початок-кінець (круглий прямокутник): початок-кінець функції, вхід-вихід з алгоритму.
   2. Блок вводу-виводу даних (нахилений прямокутник): введення даних з клавіатури, вивід результату на екран.
   3. Обчислювальний блок (прямокутник): обчислювання або послідовність обчислювання.
   4. Визначний процес (прямокутник з двома лініями): виконання підпрограми (функції).
   5. Логічний блок (ромб): перевірка умови.
   6. Цикл (шестикутник): початок циклу
3. Що визначають лінії, які з’єднують графічні блоки на блок-схемах алгоритмів?

Лінії визначають послідовність дій у алгоритмі, з’єднуючись між блоками відповідно до правил побудови блок-схем.

1. Визначити призначення графічного блоку «Визначений процес»

Елемент відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

1. Чи може в блок-схемі, в якій використано «Логічний блок», бути два графічних блоки «Кінець»?

Ні, у будь-якій блок-схемі повинен бути лише один початок і кінець.

1. Навести приклад використання графічного блоку «Перехід» для з’єднання блок-схеми між двома сторінками.



1. Правила оформлення блок-схем.

Блок-схеми розташовуються зверху вниз. Лінії з’єднання окремих блоків показують напрямок процесу обробки в схемі. Кожний такий напрям називається гілкою. Незалежно від структури алгоритму, він завжди має по одному блоку «Початок» і «Кінець». Гілки алгоритму повинні врешті-решт зійтися. Незалежно з якої гілки було розпочато рух, він завжди має призвести до блоку «Кінець». На блок-схемі кожен пункт алгоритму зображується відповідно геометричною фігурою (Див. питання 4).

1. Чому блок-схема не має містити частин коду в графічних блоках?

Блок-схема повинна бути зрозуміла не тільки програмістам, а й спеціалістам з різних областей, тому використання частин коду у блоках не допустима і повинна замінятись на математичні чи словесні форми вираження.