МЕТОДИЧНІ ВКАЗІВКИ

ДО КУРСОВОЇ РОБОТИ З ДИСЦИПЛІН

”Системне програмування”,

”Системне програмування та операційні системи"

для студентів спеціальностей 6.091500, 6.080400

(Частина 1)

Весь цифровий і фактичний матеріал та бібліографічні

відомості перевірено. Зауваження рецензента враховано

Методичні вказівки до виконання курсової роботи з дисциплін ”Системне програмування", "Системне програмне забезпечення”, ”Системне програмування та операційні системи" для студентів спеціальностей 6.091500, 6.080400

Змiст

Вступ

1. Завдання та порядок виконання курсовоі роботи

1.1 Завдання курсової роботи

2. Теоретичні відомості

2.1 Формати числових даних

2.2 Виконання арифметичних операцій

2.2.1 Арифметичні команди над двійковими цілими числами

2.2.2 Арифметичні команди над двійково-десятковими неупакованими числами

2.2.3 Арифметичні команди над двійково-десятковими упакованими числами

3. Порядок виконання курсової роботи

4. Змiст курсової роботи

4.1 Титульний лист

4.2 Список авторів

4.3 Зміст

4.4 Вступ

4.5 Суть звіту

4.6 Висновки

5. Вимоги до оформлення роботи

5.1 Загальні вимоги

5.2 Нумерація аркушів

5.3 Нумерація розділів, підрозділів, пунктів, підпунктів

5.4 Ілюстрації

6. Захист курсової роботи

Додаток А - ТИТУЛЬНА СТОРІНКА

Додаток Б - Зміст

Додаток В - Приклад оформлення та розміри елементів блок-схеми

Додаток Г - Перелік посилань

## Вступ

Вивчення курсу закінчується виконанням курсової роботи, яка є самостійною роботою студента.

Метою курсової роботи є розробка системних програм мовою асемблера або мовою високого рівня з вставками мовою асемблера. Конкретне завдання визначається відповідно до варіанту.

Допускається виконання курсової роботи групою не більш як з трьох студентів (варіант завдання та склад групи погоджується з керівником курсової роботи).

До здачі та захисту курсової роботи допускаються студенти, що повністю виконали відповідний навчальний план.

Для виконання курсової роботи рекомендується застосовувати учбово-довідкові посібники.

## 1. Завдання та порядок виконання курсовоі роботи

В разі виконання курсової роботи бригадою варіант завдання погоджується з керівником курсової роботи.

## 

## 1.1 Завдання курсової роботи

В залежності від складності завдання алгоритму програми присвоюється один із трьох рівнів складності.

Рівень виконання обирається студентом.

Завдання

Скласти програму моделювання операції додавання, віднімання і множення двійково-десяткових чисел з фіксованою точкою з заданою кількістю байтів. Введення операндів виконувати з клавіатури. Результати виводити на екран.

Програма повинна задовольняти такі умови:

1 рівень - програма виконує операції підсумовування і віднімання цілих чисел.

2 рівень - програма виконує операції підсумовування і віднімання чисел з фіксованою крапкою і виконує перевірку правильності введення чисел.

3 рівень - програма виконує операції підсумовування, віднімання і множення чисел з фіксованою крапкою, виконує перевірку правильності введення чисел і має "дружній" інтерфейс.

ВАРІАНТИ

|  |  |  |
| --- | --- | --- |
| № варіанта | Кільк. байтів в цілої і дробової частинах числа | Спосіб зберігання числа в пам'яті |
| 1 | 3 | упакований |
| 2 | 4 | неупакований |
| 3 | 5 | упакований |
| 4 | 6 | неупакований |
| 5 | 7 | упакований |
| 6 | 8 | неупакований |
| 7 | 8 | упакований |
| 8 | 3 | неупакований |
| 9 | 4 | упакований |
| 10 | 5 | неупакований |
| 11 | 6 | упакований |
| 12 | 7 | неупакований |
| 13 | 8 | упакований |
| 14 | 9 | неупакований |
| 15 | 10 | упакований |

## 2. Теоретичні відомості

## 2.1 Формати числових даних

Числа, які вводяться з клавіатури, надходять у процесор в ASCІІ - кодах. ASCІІ - код кожної десяткової цифри числа дорівнює значенню цієї цифри, збільшеному на 30h.

В свою чергу, процесор дозволяє виконувати арифметичні операції додавання, віднімання, множення і ділення над цілими числами, які можуть зберігатись у пам'яті в одному з трьох форматів: двійковий формат, двійково-десятковий упакований формат і двійково-десятковий неупакований формат. Число в двійковому форматі зберігається в пам'яті у вигляді одно - чи двобайтового числа в доповнювальному коді. Число в двійково-десятковому упакованому форматі зберігається в пам'яті у вигляді послідовності байтів. Послідовність цифр у байтах природна: в молодших байтах зберігаються старші цифри. В кожному байті зберігаються дві двійково-кодовані десяткові цифри в коді BCD (Bіnary-Coded Decіmal).

|  |  |  |  |
| --- | --- | --- | --- |
| Цифра | Двійковий код | Цифра | Двійковий код |
| 0 | 0000 | 5 | 0101 |
| 1 | 0001 | 6 | 0110 |
| 2 | 0010 | 7 | 0111 |
| 3 | 0011 | 8 | 1000 |
| 4 | 0100 | 9 | 0101 |

Приклад запису 6-розрядного десяткового числа 653201 в пам'яті в коді BCD:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | байт | 2 | байт | 3 | байт |  |
| 6 | 5 | 3 | 2 | 0 | 1 | - десяткове представлення |
| 0110 | 0101 | 0011 | 0010 | 0000 | 0001 | - двійкове представлення |

Число в двійково-десятковому неупакованому форматі зберігається у вигляді послідовності байтів. У кожному байті зберігається одна двійково-кодована десяткова цифра. Її значення міститься в чотирьох молодших бітах байта.

Перед виконанням операції множення таких чисел у старших чотирьох бітах повинні бути записані нулі. Для операції додавання і віднімання зміст цих бітів не є суттєвим.

Приклад запису 3-розрядного десяткового числа 653 у пам`яті в двійково-десятковому неупакованому форматі:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | байт | 2 | байт | 3 | байт |  |
| 0 | 6 | 0 | 5 | 0 | 3 | - десяткове представлення |
| 0000 | 0110 | 0000 | 0101 | 0000 | 0011 | - двійкове представлення |

## 2.2 Виконання арифметичних операцій

Процесори ІBM PC мають у системі команд такі команди, які дозволяють виконувати арифметичні операції додавання, віднімання, множення і ділення над цілими числами в двійковому форматі, а також у двійково-десятковому упакованому і неупакованому форматах.

## 

## 2.2.1 Арифметичні команди над двійковими цілими числами

Для виконання додавання цілих двійкових чисел можуть використовуватись команди:

ADD a,b - Додавання без урахування перенесення. Команда додає значення двох операндів а та b і розміщує результат замість першого операнда. Операнд може знаходитися в пам'яті, в регістрі або бути безпосереднім параметром у команді. Не допускається додавання двох операндів, які знаходяться в пам'яті. Першим операндом не може бути безпосередній параметр. Установлюються прапорці AF, CF, OF, FF, SF, ZF.

ADC a,b - Додавання з перенесенням. Команда додає значення двох операндів, а також значення прапорця перенесення CF (a+b+CF), і розміщує результат замість першого операнда. Обмеження на операнді такі ж, як і в команді ADD. Установлюються прапорці AF, CF, OF, FF, SF, ZF. Ця команда використовується для програмування арифметичних операцій над довгими цілими числами.

ІNC a - Збільшення на 1. Команда додає 1 до значення операнда а. Операнд може знаходитися в пам'яті або в регістрі. Установлюються прапорці AF, OF, PF, SF, ZF.

Для виконання віднімання можуть використовуватись команди:

SUB a,b - Віднімання. Команда віднімає вид значення першого операнда а значення другого операнда b і розміщує результат замість першого операнда. Операнд може знаходитися в пам'яті, в регістрі або бути безпосереднім параметром у команді. Не допускається віднімання, коли обидва операнди знаходяться в пам'яті. Першим операндом не може бути безпосередній параметр. Установлюються прапорці AF, CF, OF, PF, SF, ZF.

SBB a,b - Віднімання з позикою. Команда віднімає із значення першого операнда а значення другого операнда b, зменшує результат на значення прапорця CF (a-b-CF) і розміщує результат замість першого операнда. Обмеження операндів такі ж, як і в команді SUB. Установлюються прапорці AF, CF, OF, PF, SF, ZF.

DEC a - Зменшування на 1. Команда зменшує значення операнда а на 1. Операнд може знаходитися в пам'яті або в регістрі. Установлюються прапори AF, OF, PF, SF, ZF.

Для виконання множення цілих двійкових чисел можуть використовуватись команди:

MUL a - Множення. Команда виконує множення вмісту акумулятора (AL або AX) на значення операнда а. Результат розміщується відповідно в акумуляторі AX або в парі регістрів DX: AX. Множники розглядаються як числа без знаків. Установлюються прапорці CF і OF. Прапорці AF, PF, SF і ZF не визначені.

ІMUL a - Множення із знаком. Команда виконує множення вмісту акумулятора (AL або AX) на значення операнда а. Результат розміщується відповідно в акумуляторі AX або в парі регістрів DX: AX. Множники розглядаються як числа із знаками. Прапорці CF і OF скидаються в 0, коли старша частина добутку (регістр AH або DX) є поширенням знака, який знаходиться в молодшій частині добутку (AL або AX), інакше ці прапорці установлюються в 1.

Для виконання ділення цілих двійкових чисел можуть використовуватися команди:

DІV a - Ділення. Команда виконує ділення вмісту акумулятора на операнд a. Коли операнд а - байт, тоді ділене розміщується в AX, ціла частина в AL, а залишок - в AH. Коли операнд a - слово, тоді ділене розміщується в парі DX: AX (подвійне слово), ціла частина розміщується в AX, а залишок в DX. Операнди розглядаються як цілі числа без знака. При діленні на 0 виконується переривання типу 0.

ІDІV a - Ділення із знаком. Команда виконує ділення операндів так само, як у команді DІV. Операнди розглядаються як числа із знаками. Прапорці не визначені. При діленні на 0 виконується переривання типу 0.

## 

## 2.2.2 Арифметичні команди над двійково-десятковими неупакованими числами

Арифметичні операції над двійково-десятковими неупакованими числами виконуються порозрядно, починаючи з молодших розрядів операндів.

Слід нагадати, що для зберігання однієї цифри такого числа треба один байт.

Виконання операції над черговими розрядами операндів, тобто над черговими байтами, починається з того, що над цими байтами виконується відповідна двійкова команда додавання (ADD, ADC), віднімання (SUB, SBB), множення (MUL) або ділення (DІV).

Після того виконується корекція отриманого результату, щоб отримати правильне значення неупакованої десяткової цифри результату:

ААА - Команда коригує в AL результат операції ADD, ADC. У чотирьох молодших бітах регістра AL записується значення десяткової цифри, а в чотирьох старших бітах записуються нулі. Коли в AL результат попередньої операції перевищує 9, тоді CF=1 і AH=AH+1.

AAS - Команда коригує в AL результат операції SUB, SBB. У чотирьох молодших бітах регістра AL записується значення десяткової цифри, а в чотирьох старших бітах записуються нулі. Коли потрібно мати позику із старшого розряду, тоді CF=1 і AH=AH+1.

AAM - Корекція в AX результату множення. Команда коригує результат операції MUL. У регістрі AH записується старша цифра добутку, а в AL - молодша цифра.

AAD - Корекція ділення. Команда коригує ділене в регістрі AL перед виконанням ділення, так що наступне ділення DІV дає неупаковану десяткову частку: в AL - результат, в AH - нуль.

## 

## 2.2.3 Арифметичні команди над двійково-десятковими упакованими числами

Арифметичні операції над двійково-десятковими упакованими числами виконуються по байтах, тобто задана операція додавання або віднімання виконується одночасно над двома розрядами числа, які знаходяться в одному байті, починаючи з молодших розрядів.

Виконання операції додавання або віднімання здійснюється в два етапи. Спочатку над цим байтом виконується команда додавання (ADD, ADC) або віднімання (SUB, SBB), а потім виконується корекція результату, щоб отримати правильні значення упакованих десяткових цифр.

Для корекції результату використовуються такі команди:

DAA - Корекція результату додавання. Команда коригує в AL результат операції ADD, ADC. У регістрі AL створюються дві правильні двійково-десяткові цифри результату. Коли виникає перенесення в наступний байт, тоді CF=1.

DAS - Корекція результату віднімання. Команда коригує в AL результат операції SUB, SBB. У регістрі AL створюються дві правильні двійково-десяткові цифри результату. Коли виникає позика із наступного байта, тоді CF=1.

У процесорі ІBM PC не передбачені команди корекції множення і ділення упакованих двійково-десяткових чисел. Тому для виконання цих операцій треба спочатку розпакувати черговий байт, виконати відповідну операцію над неупакованими цифрами, а потім упакувати результат.

## 3. Порядок виконання курсової роботи

1. Ознайомитись з завданням на роботу, теоретичними відомостями та методичними вказівками до виконання роботи.

2. Проаналізувати завдання та визначити можливі напрями його реалізації.

3. Обравши спосіб виконання завдання ознайомитись з додатковими можливостями мови програмування, які можуть знадобитися при реалізації завдання.

4. Застосувати отримані навички та знання для безпосередньої реалізації завдання.

5. Оформити звіт.

## 4. Змiст курсової роботи

Звіт з курсової роботи є документом, на підставі якого студент стримує оцінку за курсову роботу. Звіт повинен мати обсяг 20-25 сторінок.

Звіт умовно розділяють на: вступну частину, основну частину, додатки, матеріал наприкінці звіту.

Вступна частина містить наступні структурні елементи: титульний лист, список авторів, зміст.

Основна частина містить наступні структурні елементи: вступ, суть звіту, висновки, перелік посилань.

До суті звіту відносять: теоретичні відомості, алгоритм, блок-схему алгоритму, лістинг, результати роботи.

## 4.1 Титульний лист

Звіт починається з титульної сторінки, яка є першою сторінкою звіту, номер сторінки на ній не ставлять. Оформлення титульної сторінки має відповідати державному стандарту ДСТУ 3008-95.

Титульна сторінка містить дані, які розміщують в наступній послідовності:

відомості про виконавця роботи - юридичну особу (організацію):

найменування міністерства, в систему якого входить організація;

повна і скорочена назва організації;

назва кафедри;

повне найменування документа включає в себе вид і найменування роботи (у тому числі й умовне, якщо воно мається), за результатами якої підготовлений звіт:

назву документа - слово “ЗВІТ" (друкують прописними літерами посередині сторінки);

вид і найменування роботи (у тому числі й умовне, якщо воно мається), за результатами якої підготовлений звіт.

Вид роботи друкують прописними буквами, найменування роботи - рядковими з першої прописний.

назву звіту - друкують прописними літерами;

підписи відповідальних осіб, включаючи керівника роботи;

рік складання (утвердження) звіту.

Переноси слів в заголовках титульного листа не допускаються.

Підписи відповідальних осіб, включаючи керівника роботи, оформляють наступним чином: зліва вказують посаду, учені ступені, учені звання відповідальних осіб, потім залишають вільне місце для особистих підписів і справа від них у відповідних рядках розміщують перші букви імен з крапкою і прізвища осіб які підписали звіт; нижче особистих підписів проставляють дати підписання.

Також на титульній сторінці розміщують посаду виконавця роботи, потім залишають вільне місце для особистий підписі і справа від неї розміщують першу букву імені з крапкою і прізвище виконавця роботи.

Рік складання звіту розміщують посередині рядка в нижній частині титульної сторінки.

Зразок оформлення титульної сторінки наводиться в додатку А.

## 4.2 Список авторів

Список авторів поміщають безпосередньо за титульним листом на новій сторінці. В списку в загальному випадку приводять імена (чи перші букви імен) і прізвища авторів, їхньої посади з указівкою частини звіту, підготовленої конкретним автором. Якщо звіт виконаний одним автором, зведення про нього вказують на титульному листі.

## 4.3 Зміст

У загальній кількості аркушів це друга сторінка. Зміст містить: вступ; послідовно перелічені найменування всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають назви) суті звіту; висновки; перелік посилань, номера сторінок, на яких міститься початок матеріалу.

Найменування розділів мають бути короткими, відповідати змісту та записуватися у вигляді заголовків великими літерами. Перенос слів у заголовках не допускається. Крапку в кінці заголовка не ставлять.

Рекомендований зміст звіту:

Вступ

Теоретичні відомості

Алгоритм роботи

Блок-схема алгоритму

Лістинг програми

Результати роботи

Висновки

Перелік посилань

Приклад оформлення змісту наведено в додатку Б.

## 4.4 Вступ

В цьому розділі необхідно стисло викласти:

мету цієї роботи;

завдання і варіант завдання.

Вступ розташовують на окремій сторінці.

## 4.5 Суть звіту

Суть звіту - це викладення відомостей про предмет дослідження чи розробки, які необхідні і достатні для розкриття сутності даної роботи (опис: теорії; методів роботи; характеристик та/або властивостей створеного об’єкта; принципів дії об’єкта та ін.) та її результатів.

Суть звіту викладають, поділяючи матеріал на розділи. Розділи можуть поділятися на пункти або на підрозділи та пункти. Пункти, якщо це необхідно, поділяють на підпункти. Кожний пункт та підпункт повинен мати закінчену інформацію.

Схема - це графічне зображення алгоритму, доповнене елементами словесного запису. Кожен пункт алгоритму відображається на схемі певною геометричною фігурою-блоком, при чому різним діям відповідають різні геометричні фігури.

Найчастіше схеми алгоритмів роблять у вигляді блок-схем. Блок-схеми будуються на основі стандартних графічних елементів.

Правила виконання схем алгоритмів регламентуються ДСТ 19.002-80. Застосовувані графічні символи, що відображають основні операції процесу обробки даних, установлені ДСТ 19.003-80.

Елементи блок-схем мають стандартизовані розміри, яких треба дотримуватися при створенні технічної документації, оформленні розрахунково-графічних робіт, курсових робіт, дипломних проектів. Основний напрямок потоку інформації в блок-схемі іде згори донизу і зліва направо.

При великій кількості ліній, які перетинаються, великій їх довжині та багатократних змінах напрямків схеми стають мало наочними. У цих випадках припускається розривати лінії потоку інформації, розміщаючи на обох кінцях розірвання спеціальні символи „з'єднувач”. Якщо схема розташовується на декількох аркушах, перехід ліній потоку з одного аркуша на інший позначається за допомогою символів „з'єднувач між сторінками".

При виконанні схем алгоритмів необхідно дотримуватись мінімальної відстані 3 мм між рівнобіжними лініями потоків та 5 мм - між іншими словами.

Приклад оформлення та розміри елементів блок-схеми наведено в додатку В.

## 4.6 Висновки

Висновки розміщують безпосередньо після викладення суті звіту на новій сторінці.

У висновках наводять оцінку отриманих результатів роботи.

Перелік посилань

Перелік посилань повинен бути наведений в кінці тексту звіту, починаючи з нової сторінки. У відповідних місцях тексту повинні бути посилання.

Бібліографічні описи в переліку посилань наводять в черзі, в якій вони вперше згадуються в тексті. Порядкові номера описів в переліку є посиланнями в тексті.

Приклад оформлення переліку посилань наведено в додатку Г.

## 5. Вимоги до оформлення роботи

## 5.1 Загальні вимоги

Кожен аркуш повинен мати поля на відстані від краю: праворуч 10 мм та по 20 мм з трьох інших сторін.

Текст має бути написаний охайно та чітко. Помилки та графічні зображення допускається виправляти підчищенням або зафарбовуванням білою фарбою і нанесенням на тому ж місці або між рядками виправленого зображення. Виправлення повинно бути чорного кольору. Перекреслювання та нестандартні скорочення слів не допускаються. При потребі сторінку з умовними скороченнями розташовують відразу за листом “Зміст".

Термінологія та визначення повинні бути єдиними і відповідати встановленим стандартам.

Допускається виконання записки в тестовому редакторі MSWord і друкування на принтері, на аркушах формату А4. При цьому повинні виконуватися таки вимоги:

шрифт - Times New Roman, розмір - 14;

міжрядковий інтервал - полуторний;

інтервал між словами - один пробіл;

таблиці потрібно створювати за допомогою табличних функцій редактора MSWord;

оформлення тексту повинно бути однаковим (виділення заголовків, підзаголовків, підмалюнкових написів) протягом усього документа.

Структурні елементи "СПИСОК АВТОРІВ", "Зміст", "Вступ", "ВИСНОВКИ", "ПЕРЕЛІК ПОСИЛАНЬ", не нумерують, а їх найменування служать заголовками структурних елементів.

Розділи і підрозділи "СУТІ ЗВІТУ" повинні мати заголовки. Пункти і підпункти можуть мати заголовки.

Заголовки структурних елементів звіту і заголовки розділів розміщуються в середині рядка і друкуються прописними буквами без крапки наприкінці, не підкреслюючи.

Заголовки підрозділів, пунктів і підпунктів звіту починаються з абзацного відступу і друкуються малими літерами, крім першої прописний, не підкреслюючи, без крапки наприкінці.

Абзацний відступ повинний бути однаковим у всьому тексті звіту і рівним п'ятьом знакам.

Якщо заголовок складається з двох чи більш речень, їх розділяють крапкою.

Переноси слів у заголовку розділу не допускаються.

Відстань між заголовком і наступним чи попереднім текстом повинне бути не менш двох рядків.

Не допускається розміщати \_найменування розділу, підрозділу, а також пункту і підпункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту.

## 5.2 Нумерація аркушів

Аркуші звіту потрібно нумерувати арабськими цифрами, дотримуючи наскрізної нумерації зі всьому тексту звіту. Номер сторінки проставляють у правому верхньому куті без крапки наприкінці.

Нумерація аркушів пояснювальної записки має бути наскрізною, починаючи з титульної сторінки й закінчуючи переліком посилань.

Титульний лист включають у загальну нумерацію сторінок звіту. Номер сторінки на титульному листі не проставляють.

## 5.3 Нумерація розділів, підрозділів, пунктів, підпунктів

Розділи звіту повинні мати порядкову нумерацію в межах викладу суті звіту і позначатися арабськими цифрами без крапки, наприклад 1, 2 і т.д. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад, 1.1, 1.2 і т.д.

## 5.4 Ілюстрації

Ілюстрації (креслення, малюнки, графіки. схеми, діаграми) потрібно розташовувати в звіті безпосередньо після тексту, у якому вони згадуються вперше, або на наступній сторінці. На всі ілюстрації повинні бути дані посилання в звіті.

Креслення, малюнки, графіки, схеми, діаграми, яки розміщені в звіті, повинні відповідати вимогам стандартів "Єдиної системи конструкторської документації" і "Єдиної системи програмної документації".

Ілюстрації можуть мати назви, що поміщають під ілюстрацією.

При необхідності під ілюстрацією поміщають дані, що роз’яснюють ілюстрацію (підмалюнковий текст).

Ілюстрації позначається словом "Малюнок", що разом з назвою ілюстрації поміщають після даних, що роз’яснюють ілюстрацію, наприклад, “Малюнок 1.2.1 Буфери, зв`язані між собою в циклічний список".

Ілюстрації потрібно нумерувати арабськими цифрами порядковою нумерацією в межах розділу.

Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, відокремлених крапкою, наприклад, Малюнок 3.2 - другий малюнок третього розділу.

Таблицю слід розташовувати безпосередньо після тексту, в якому вона вперше згадується, або на наступній сторінці. Слово “Таблиця ” зазначають лише один раз зліва. Таблиці нумеруються арабськими цифрами. Номер таблиці складається з номера розділу і порядкового номера таблиці, відокремлених крапкою, наприклад, таблиця 2.1 - перша таблиця другого розділу. Таблиця може мати назву, яку розміщують над таблицею. Назва повинна бути стислою і відображати вміст таблиці.

## 6. Захист курсової роботи

Роботу над курсовою роботою студент повинен планувати таким чином, щоб закінчити її в термін, визначений навчальним планом.

Перед захистом курсової роботи студент має продемонструвати роботу програми керівникові та отримати допуск до захисту.

В процесі захисту студент пояснює призначення і принцип дії реалізованої ним програми (задачі) та відповідає на запитання керівника.

Оцінку по роботі керівник визначає з урахуванням усіх її складових: якості виконання та рівня складності роботи, змісту та оформлення звіту, знань студента при захисті і термін виконання роботи відповідно до навчального плану.

Якщо студент одержує незадовільну оцінку, він або опрацьовує роботу за зауваженнями, або отримує нове завдання. Це питання вирішує керівник роботи.

## Додаток А - ТИТУЛЬНА СТОРІНКА

Міністерство освіти та науки України

Черкаський державний технологічний університет (ЧДТУ)

Кафедра системного програмування

ЗВІТ

ПРО КУРСОВУ РОБОТУ З ДИСЦИПЛІНИ

“Системне програмне забезпечення”

РОЗРОБКА ПРОЦЕДУР ВИКОНАННЯ КОМАНД Введення/ВИВЕДЕННЯ

Перевірив к. т. н., доц. - В. Платонов

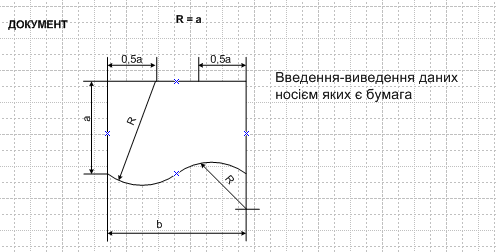
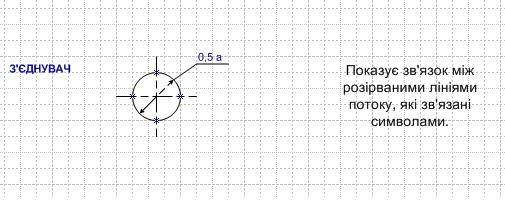
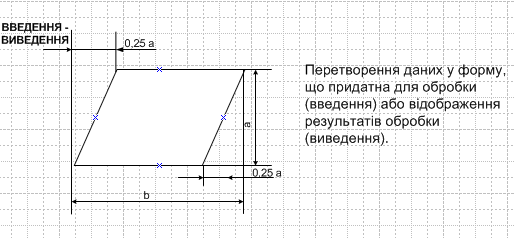
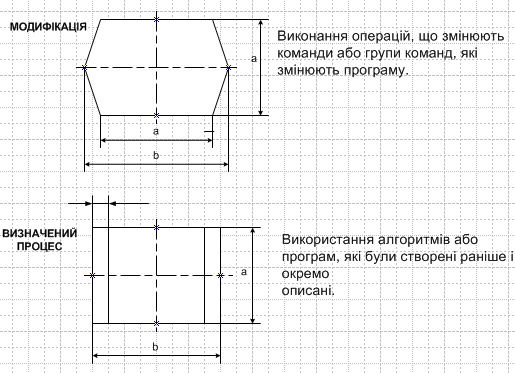
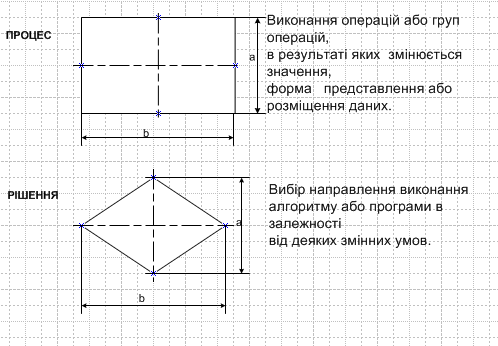
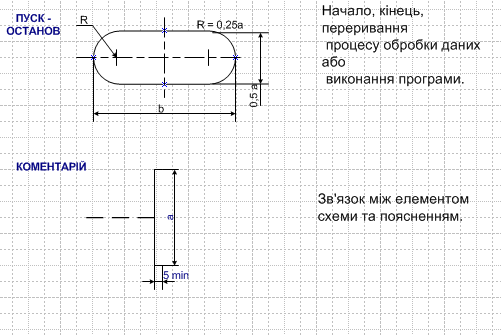
Виконав ст. групи КМ-111 Р. Кононенко

Черкаси 2004

## Додаток Б - Зміст

|  |
| --- |
| Вступ |
| 1. Керування введенням/виведенням по запитам |
| 1.1 Керування при відсутності і при наявності буферів введення/ виведення, буферний свопинг |
| 1.2 Метод циклічних буферів і сопрограм |
| 1.3 Керування по перериванню при введенні/виведенні |
| 1.4 Керування по перериванню за методом циклічних буферів для операції введення/виведення |
| 1.5 Метод пула буферів |
| 2 Алгоритм роботи |
| 3 Блок-схема алгоритму |
| 4 Лістинг програми |
| 5 Результати роботи |
| Висновки |
| Перелік посилань |

## Додаток В - Приклад оформлення та розміри елементів блок-схеми



Розмір *a* повинен обиратися з ряду 10, 15,20 мм. Допускається збільшувати розмір *a* на число, яке кратне 5. Розмір *b* дорівнює 1,5*a*.

При виконанні умовних графічних позначень автоматизованим методом розміри геометричних елементів символів округляються до значень, обумовлених технічними можливостями пристроїв, які використовуються.

## Додаток Г - Перелік посилань

Перелік посилань:

1. Методичні вказівки до виконання лабораторних робіт з дисципліни "Системне програмне забезпечення" для студентів денної форми навчання. Частина 2/Укл. Платонов В.А. - Черкаси ЧІТІ, 1999. - 38с.
2. Брябрин В.М." Программное обеспечение персональных ЭВМ ", М., "Наука", 1988.
3. Рудаков П.И. “ Язык ассемблера ", М., Диалог-МИФИ, 2001
4. Зубков С.В. ASSEMBLER для DOS, Windows и Unix. М. 2004
5. Абель П. Язык ассемблера для IBM PC и программирования. Пер. с англ. М., Высшая школа, 1992
6. Чижов А.А. Системные программные средства ПЭВМ.М. Финансы и статистика. 1989
7. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT, AT. Пер. с англ., М., Финансы и статистика, 1991
8. Майко Г.В. Ассемблер для IBM PC. Одесса, 1998
9. Юров В. ASSEMBLER. Учебник. СПб, "Питер", 2001