

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ**

Кафедра інформаційних технологій та кібербезпеки факультету №4

**МЕТОДИЧНІ МАТЕРІАЛИ
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ**

**з навчальної дисципліни "Системне програмування"
вибіркових компонент
освітньої програми першого рівня вищої освіти**

125 "Кібербезпека" (Безпека інформаційних та комунікаційних систем)

Харків 2021

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 25.03.2021 № 3

СХВАЛЕНО

Вченою радою факультету №4
Протокол від 17.03.2021 № 3

ПОГОДЖЕНО

Секцією науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 22.03.2021 № 3

Розглянуто на засіданні кафедри інформаційних технологій та
кібербезпеки
(протокол від 16.03.2021 № 6)

Розробник:

старший викладач кафедри інформаційних технологій та кібербезпеки
ХНУВС Калякін С.В.

Рецензенти:

доцент кафедри інформаційних технологій та кібербезпеки факультету №4
Харківського національного університету внутрішніх справ к.т.н., доцент
Тулупов В.В.

доцент кафедри інформаційних технологій і систем управління ХарPI
НАДУ при Президентіві України к.т.н. доцент Кобзев І.В.

1. Розподіл часу навчальної дисципліни за темами

Номер та назва навчальної теми	Кількість годин відведених на вивчення навчальної дисципліни					Вид контролю
	Всього	з них:				
		Лекції	Практичні заняття	Лабораторні заняття	Самостійна робота	
Семестр №5						
Тема № 1. Особливості системного програмування в ОС Windows.	6	2	0	0	4	екза-мен
Тема № 2. Мова програмування C/C++. Типи даних, вирази й операції.	14	2	4	0	8	
Тема № 3. Препроцесор мови C/C++.	10	2	0	0	8	
Тема № 4. Управляючі структури мови C/C++.	18	2	4	4	8	
Тема № 5. Обробка строк. Unicode.	14	2	0	4	8	
Тема № 6. Робота з динамічною пам'яттю.	14	2	4	0	8	
Тема № 7. Динамічні структури даних.	14	2	0	4	8	
Тема № 8. Обробка файлів засобами мови C/C++.	16	4	0	4	8	
Тема № 9. Обробка файлів засобами Windows API.	14	2	0	4	8	
Всього за семестр №5	120	20	12	20	68	
Всього за дисципліною	120	20	12	20	68	

2. Методичні вказівки до лабораторних занять

Лабораторна робота №1.

Тема: «Управляючі структури мови C/C++».

Навчальна мета заняття: ознайомитись із використанням циклічних конструкцій для реалізації ітераційних процесів.

Кількість годин: 4 год.

Навчальні питання

1. Оператори циклу.
2. Реалізація ітераційного процесу.

Матеріально-технічне забезпечення: комп'ютерний клас із встановленим інтегрованим середовищем Visual C ++.

План проведення заняття

1. Загальні відомості.

Перед початком роботи ознайомтесь з текстом лекції № 4.

В C та інших сучасних мовах програмування цикли дозволяють виконувати набір операторів, поки не виконається деяка умова. Наприклад:

```
#include <stdio.h>
int readnum(), prompt();
int sqrnum(int num);

main()
{
    int t;
    for (prompt(); t = readnum(); prompt())    sqrnum(t);
}

int prompt()
{
    printf(": ");
}

int readnum()
{
    int t;
    scanf("%d", &t)
    return t;
}

int sqrnum(int num)
{
    printf("%d\n", num*num);
}
```

Якщо уважно подивитися на цикл for в main (), то можна побачити, що кожна частина циклу містить виклики функцій, які здійснюють підказку і

читання вводиться з клавіатури числа. Якщо введене число дорівнює нулю, цикл закінчується, оскільки умова помилкова, інакше число зводиться в квадрат. Отже, в даному циклі частини ініціалізації і збільшення використовуються не традиційно, але абсолютно коректно.

Ще один приклад використання циклу:

```
void func1(void)
{
    int  working;
    working = 1;          /* 1 - істина */
    while (working) {
        working = process1();
        if (working)
            working = process2();
        if (working)
            working = process3();
    }
}
```

І третій приклад:

```
void menu(void)
{
    char  ch;

    printf("1. Перевірка орфографії\n");
    printf("2. Корегування помилок\n");
    printf("3. Вивести на екран\n");
    printf("        Зробіть ваш вибір:  ");
    do {
        ch = getche();          // читання символу з клавіатури
        switch (ch) {
            case '1' :
                check_spelling();
                break;
            case '2' :
                correct_errors();
                break;
            case '3' :
                display_errors();
                break;
        }
    } while (ch!='1' && ch!='2' && ch!='3');
}
```

2. Постановка завдання

Скласти таблицю значень функції, використовуючи її розкладання в ряд. Значення функції підрахувати з точністю ε . Результат вивести на екран.

Варіанти завдання обираються згідно останньої цифри номеру в журналі (якщо 0 – обираємо 10 варіант)

1) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,001$.

$$\frac{1}{(1+1)^2} + \frac{1}{4(2+1)^2} + \dots + \frac{1}{n^2(n+1)^2} + \dots$$

2) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,005$.

$$\frac{1}{3} + \frac{1}{8} + \dots + \frac{1}{n^2 - 1} + \dots$$

3) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,0001$.

$$1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n} + \dots$$

4) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,005$.

$$1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots \pm \frac{1}{2^n} \mp \dots$$

5) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,00005$.

$$\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \dots + \frac{1}{(2n-1)(2n+1)} + \dots$$

6) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,0001$.

$$\frac{1}{3 \cdot 5} + \frac{1}{7 \cdot 9} + \dots + \frac{1}{(4n-1)(4n+1)} + \dots$$

7) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,001$.

$$\frac{1}{1^2} + \frac{1}{3^2} + \dots + \frac{1}{(2n+1)^2} + \dots$$

8) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,005$.

$$1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots + \frac{1}{n^4} + \dots$$

9) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,0005$.

$$1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots \pm \frac{1}{n^4} \mp \dots$$

10) Обчислити наближене значення нескінченної суми з точністю до $\varepsilon = 0,001$.

$$\frac{1}{1^4} + \frac{1}{3^4} + \dots + \frac{1}{(2n+1)^4} + \dots$$

3. Зміст звіту

- 3.1. Тема та мета роботи.
- 3.2. Постановка завдання.
- 3.3. Текст програм.
- 3.4. Результати виконання програм.
- 3.5. Аналіз помилок та висновки по роботі.

Лабораторна робота №2

Тема: «Обробка символів та рядків»

Навчальна мета заняття: Засвоїти методи обробки символьних даних.

Кількість годин: 4 год.

Навчальні питання

1. Функції мови C для обробки символьних даних.
2. Реалізація власних функцій для обробки символьних даних.

Матеріально-технічне забезпечення: комп'ютерний клас із встановленим інтегрованим середовищем Visual C ++.

План проведення заняття

1. Загальні відомості

Перед початком роботи ознайомтесь з текстом лекції № 5.

Далі розглянемо приклади функцій, які використовуються для роботи з символьними даними.

Приклад 1. Функція `isalpha_cyr` виконує перевірку на малу кіриличну літеру (в кодуванні Windows)

```
#include <stdio.h>
int isalpha_cyr (unsigned char c) {
    return ((c >= 'а') && (c <= 'я'));
}
void main () {
    unsigned char alpha;
    scanf ( "% c", & alpha);
    printf ( "% d", isalpha_cyr (alpha));
    fflush (stdin); getchar ();}
```

Приклад 2. Функція strrev - перевертання рядку в зворотній вигляд.

```
#include <stdio.h>
#define byte unsigned char
unsigned strlen (byte * s) { // визначення довжини
рядку
    unsigned l = 0;
    while (* s ++ != '\0') l ++;
    return l;
}

void strrev (byte * s) { // перевертання рядку
    byte c, * t;
    for (t = s + strlen (s) - 1; s < t; s ++, t --) {
        c = * s; * s = * t; * t = c;
    }
}

void main () { // головна програма
    byte s [80];
    scanf ( "% s", & s [0]);
    strrev (s);
    printf ( "% s", s);
    fflush (stdin); getchar ();
}
```

2. Завдання:

Завдання 1. Засобами мови C++ реалізувати функцію перевірки і перетворення символів. Порівняти роботу функції зі стандартною функцією Cі, якщо вона існує (див. Файл ctype.h) Ваша функція повинна враховувати особливості кодування кирилиці в DOS і Windows!

Варіанти (обираються згідно останньої цифри номеру в журналі):

1. ismaild - перевірка на символ, допустимий в адресі E-mail (латинські літери, цифра, точка, знак @)
2. isdigit - перевірка на десяткову цифру
3. isurl - перевірка на символ, допустимий в адресі Web-ресурсу
4. ispunct - перевірка на знак пунктуації
5. isxdigit - перевірка на символ 16-ричного числа
6. tolower - перетворення в малу літеру, якщо передана прописна буква
7. toupper - перетворення в прописну букву, якщо передана мала літера
8. isodigit - перевірка на символ 8-ричного числа
9. isnamed - перевірка на символ, допустимий в іменах змінних (латинські літери, цифри, знак підкреслення)
10. isquotable - перевірка на символ, який є апострофом або лапками

Завдання 2. Засобами мови C++ реалізувати функцію роботи з рядками. Порівняти роботу функції зі стандартною функцією C (файл string.h) Ваша функція повинна враховувати особливості кодування кирилиці в DOS і Windows!

Варіанти (обираються згідно останньої цифри номеру в журналі)

1. strcat - зчеплення рядків
2. strchr - знаходження першого входження символу в рядок
3. strrchr - знаходження останнього входження символу в рядок
4. strcmp - порівняння двох рядків, ігноруючи регістр символів
5. strspn - пошук позиції, починаючи з якої рядки різняться
6. strlwr - перетворення рядка в нижній регістр
7. strstr - знайти входження першого рядка в другий
- 8.strupr - перетворення рядка в верхній регістр
9. strcmp - порівняння двох рядків, з огляду на регістр символів
10. strncat - зчеплення рядків з обмеженням максимальної довжини

отриманого рядка

3. Зміст звіту

- 3.1. Тема та мета роботи.
- 3.2. Постановка завдання.
- 3.3. Текст програм.
- 3.4. Результати виконання програм.
- 3.5. Аналіз помилок та висновки по роботі.

Лабораторна робота № 3

Тема: «Робота з динамічними структурами»

Навчальна мета заняття: Засвоїти методи обробки динамічних структур даних.

Кількість годин: 4 год.

Навчальні питання

1. Особливості використання мови C++ для обробки динамічних структур даних.
2. Реалізація власної програми для обробки динамічних структур даних.

Матеріально-технічне забезпечення: комп'ютерний клас із встановленим інтегрованим середовищем Visual C ++.

План проведення заняття

1. Загальні відомості

Перед початком роботи ознайомтесь з текстом лекції № 7.

Далі розглянемо приклад програми, яка використовує для роботи динамічні структури даних.

Приклад. Написати програму, що виконує наступні дії:

- уведення із клавіатури даних у масив BAZA, що складається із шести елементів типу AVTO;
- вивести на екран інформацію про власників автомобіля, марка якого вводится з клавіатури;
- якщо таких немає, то на екран дисплея вивести відповідне повідомлення.

Програма рішення цієї задачі буде мати вигляд:

```
#include <iostream.h>
#include <string.h>
#include <windows.h>
char buf[256];
// функція для виводу тексту, що написано кирилицею
char *Cyr(char *text)
{CharToOem(text,buf);
return buf;}
struct avto {
    char fam[25];
    char marka[20];
    char nomer[10];};
void output_BAZA(avto *,int );
void main(void)
{  int i,n;
    bool flag;
    cout<<Cyr("Уведіть кількість записів: ");
    cin>>n;
    avto *BAZA = new avto[n], temp;
    char m[20];
    for(i=0;i<n;i++)
    { // Уведення даних
        cout<<Cyr("Уведіть прізвище власника ")<<i+1;
        cout<<Cyr("-го автомобіля: ");
        cin >> BAZA[i].fam;
        cout<<Cyr("Уведіть марку ")<<i+1;
        cout<<Cyr("-го автомобіля: ");
        cin >> BAZA[i].marka;
        cout<<Cyr("Уведіть номер ")<<i+1;
        cout<<Cyr("-го автомобіля: ");
        cin >> BAZA[i].nomer; }
    cout << endl << Cyr("Уведіть марку автомобіля: ");
    cin >> m;
    cout << endl << Cyr("Шукані автомобілі") << endl;
    flag = true;
    for(i=0;i<n;i++)
        if(!strcmp(m,BAZA[i].marka))
        { cout.setf(ios::left);
          cout.width(15);
          cout<<BAZA[i].fam;// *(BAZA+i)->fam;
          cout.setf(ios::right);
```

```

        cout.width(5);
        cout<<BAZA[i].nomer<<endl;
        flag = false; }
        if(flag)
            {cout<<Cyr("\nмарки ")<<m;
            cout<<Cyr("немає в списках");}
        delete [] BAZA;
    }

```

2. Варіанти завдань (обираються згідно останньої цифри номеру в журналі)

Варіант 1. Описати структуру з ім'ям STUDENT, що містить наступні поля:

- NAME - прізвище та ініціали;
- GROUP - номер групи;
- SES - успішність (масив з п'яти елементів).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список STUD1, що складається з десяти структур типу STUDENT;
- виводить на дисплей прізвищ і номерів груп для всіх студентів, які мають хоча б одну оцінку меншу 3;
- якщо таких студентів немає, вивести відповідне повідомлення.

Варіант 2. Описати структуру з ім'ям AERO, що містить наступні поля:

- NAZN - назва пункту призначення рейсу;
- NUMR - номер рейсу;
- TIP - тип літака.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список AIRPORT, що складається з семи елементів типу AERO;
- виводить на екран номерів рейсів і типів літаків, що вилітають в пункт призначення, назва якого співпало з назвою, введеним з клавіатури;
- якщо таких рейсів немає, видати на дисплей відповідне повідомлення.

Варіант 3. Описати структуру з ім'ям WORKER, що містить наступні поля:

- NAME - прізвище та ініціали працівника;
- POS - назва займаної посади;
- YEAR - рік оформлення на роботу.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список TABL, що складається з десяти структур типу WORKER;
- виводить на дисплей прізвищ працівників, чий стаж роботи в організації перевищує значення, введене з клавіатури;
- якщо таких працівників немає, вивести на дисплей відповідні повідомлення.

Варіант 4. Описати структуру з ім'ям TRAIN, що містить наступні поля:

- NAZN - назва пункту призначення;
- NUMR - номер поїзда;
- TIME - час відправлення.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список RASP, що складається з восьми елементів типу TRAIN;
- виводить на екран інформації про поїзди, що відправляються після введеного з клавіатури часу;
- якщо таких поїздів немає, видати на дисплей відповідне повідомлення.

Варіант 5. Описати структуру з ім'ям MARSH, що містить наступні поля:

- BEGST - назва початкового пункту маршруту;
- TERM - назва кінцевого пункту маршруту;
- NUMER - номер маршруту.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список TRAFIC, що складається з восьми елементів типу MARSH;
- виводить на екран інформації про маршрут, номер якого введено з клавіатури;
- якщо таких маршрутів немає, видати на дисплей відповідне повідомлення.

Варіант 6. Описати структуру з ім'ям NOTE, що містить наступні поля:

- NAME - прізвище, ім'я;
- TEL - номер телефону;
- BDAY - день народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список BLOCKNOTE, що складається їх восьми елементів типу NOTE;
- виводить на екран інформації про людей, чиї дні народження припадають на місяць, значення якого введено з клавіатури;
- якщо таких немає, видати на дисплей відповідне повідомлення.

Варіант 7. Описати структуру з ім'ям ZNAK, що містить наступні поля:

- NAME - прізвище, ім'я;
- ZODIAC - знак Зодіаку;
- BDAY - день народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список BOOK, що складається з восьми елементів типу ZNAK;
- виводить на екран інформації про людину, чиє прізвище введена з клавіатури;
- якщо такого немає, видати на дисплей відповідне повідомлення.

Варіант 8. Описати структуру з ім'ям PRICE, що містить наступні поля:

- TOVAR - назва товару;
- MAG - назва магазину, в якому продається товар;
- STOIM - вартість товару в рублях.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список SPISOK, що складається з восьми елементів типу PRICE;
- виводить на екран інформації про товар, назва якого введено з клавіатури;
- якщо такого товару немає, видати на дисплей відповідне повідомлення.

Варіант 9. Описати структуру з ім'ям ORDER, що містить наступні поля:

- PLAT - розрахунковий рахунок платника;
- POL - розрахунковий рахунок одержувача;
- SUMMA - перераховується сума в рублях.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список SPISOK, що складається з восьми елементів типу ORDER;
- виводить на екран інформації про суму, знятої з розрахункового рахунку платника, введеного з клавіатури;
- якщо такого розрахункового рахунку немає, видати на дисплей відповідне повідомлення.

Варіант 10. Описати структуру з ім'ям BOOK, що містить поля:

- FAM – прізвище та ім'я автора;
- NAZV – назва книги;
- YEAR – рік видання.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в динамічний список LIBRARY, що складається з десяти елементів типу BOOK;
- виводить на екран інформацію про книги, написаних автором, прізвище якого вводиться з клавіатури;
- якщо таких книг немає, то вивести на екран дисплея відповідне повідомлення.

3. Зміст звіту

- 3.1. Тема та мета роботи.
- 3.2. Постановка завдання.
- 3.3. Текст програм.
- 3.4. Результати виконання програм.
- 3.5. Аналіз помилок та висновки по роботі.

Лабораторна робота № 4

Тема: «Обробка файлів стандартними засобами мови C/C++»

Навчальна мета заняття: Засвоїти методи обробки текстових файлів стандартними засобами мови C/C++.

Кількість годин: 4 год.

Навчальні питання

1. Особливості використання мови C для обробки текстових файлів.
2. Реалізація власної програми для обробки текстових файлів.

Матеріально-технічне забезпечення: комп'ютерний клас із встановленим інтегрованим середовищем Visual C ++.

План проведення заняття

1. Загальні відомості

Перед початком роботи ознайомтесь з текстом лекції № 8.

2. Варіанти завдань (обираються згідно останньої цифри номеру в журналі)

Варіант 1. Створити текстовий файл, що складається з 6 рядків не більше ніж по 15 символів. Написати програму, що виконує наступні функції:

- підрахунок загальної кількості символів;
- підрахунок числа цифрових і нецифрових символів;
- підрахунок частоти знаходження послідовності символів "abc".

Варіант 2. Дано два символічних файли S1 і S2, що містять слова, розділені символом '/'. Створити файл S3, кожне слово якого утворюється зчепленням слів з файлів S1 і S2, у яких збігаються перші символи. Для поділу слів використовувати пробіл. Якщо не відбулося жодного зчеплення, видати повідомлення: "Файл S3 порожній".

Варіант 3. Дан символічний (текстовий) файл T. Групи символів, розділені пробілами утворюють слова. Підрахувати кількість слів у файлі, знайти найдовше слово, підрахувати кількість слів, що складаються з одного, двох, трьох і т. д. символів.

Варіант 4. Дан текстовий файл (файл складається з рядків) F, що містить програму на мові C. Перевірити цю програму на невідповідність числа

відкриваються і закриваються круглих дужок. Вважати, що кожен оператор програми займає не більш одного рядка файлу F.

Варіант 5. Дан текстовий файл F. Записати в перевернутому вигляді рядки файлу F в файл G. Порядок рядків у файлі G повинен бути зворотним по відношенню до порядку рядків вихідного файлу F.

Варіант 6. Дан файл F, компоненти якого є цілими числами. Жодна з компонент файлу F не дорівнює нулю. Числа в файлі йдуть в такому порядку: десять позитивних, десять негативних, десять позитивних, десять негативних і т. д. Число компонент файлу має бути кратно 40. Переписати компоненти файлу F в файл G, щоб у файлі G числа йшли в наступному порядку: п'ять позитивних, п'ять негативних, п'ять позитивних, п'ять негативних і т. д.

Варіант 7. Дано текстовий файл F і рядок s. Отримати всі рядки файлу, що містять в якості фрагмента рядок s.

Варіант 8. Дано два текстові файли F і G. Визначити, чи збігаються компоненти (рядки) файлу F з компонентами файлу G. Якщо немає, то вивести номер першого рядка і позицію першого символу в цьому рядку, в яких файли F і G відрізняються між собою.

Варіант 9. Дан текстовий файл F, кожен рядок в якому складається з одного слова, розмір якого не більше 20 символів. Переписати цей файл, розмістивши слова в алфавітному порядку.

Варіант 10. Створити текстовий файл, що складається з 5 рядків, кожен з яких має не більше ніж по 10 малих літер кириличного алфавіту. Оскільки в мові C немає функції, яка перетворює символи кириличного алфавіту з малих на прописні (є тільки для латинських букв), розробити таку функцію і використати для перетворення символів з малих в прописні в цьому файлі. Перетворений текст записати в новий текстовий файл.

3. Зміст звіту

3.1. Тема та мета роботи.

3.2. Постановка завдання.

3.3. Текст програм.

3.4. Результати виконання програм.

3.5. Аналіз помилок та висновки по роботі.

Лабораторна робота № 4

Тема: «Обробка файлів засобами Windows API»

Навчальна мета заняття: Засвоїти методи обробки файлів стандартними засобами Windows API.

Кількість годин: 4 год.

Навчальні питання

3. Особливості використання Windows API для обробки файлів.
4. Реалізація власної програми для обробки файлів засобами Windows API.

Матеріально-технічне забезпечення: комп'ютерний клас із встановленим інтегрованим середовищем Visual C ++.

План проведення заняття

1. Загальні відомості

Перед початком роботи ознайомтесь з текстом лекції № 9.

2. Варіанти завдань (обираються згідно останньої цифри номеру в журналі)

Варіант 1. Створити файл цілих чисел (не менш 20 компонент). Для даного файлу виконати наступні операції:

- упорядкувати файл по зростанню значень компонентів;
- знайти суму парних чисел серед компонентів;
- підрахувати кількість повних квадратів серед компонентів.

Результати записати в текстовий файл.

Варіант 2. Створити файл цілих чисел (не менш 25). Створити на базі цього файлу три файли - F1, F2 і F3. У файл F1 помістити в порядку зростання позитивні компоненти, у файл F2 помістити в порядку убутання негативні компоненти, у файл F3 помістити всі нульові компоненти.

Число компонентів у кожному файлі вивести в текстовий файл.

Варіант 3. Створити три файли F1, F2 і F3, що складаються кожного не менш чим з 8 компонентів цілого типу. Упорядкувати кожний файл по убутанню значень компонентів. Об'єднати файли F1, F2 і F3 в один файл у послідовності: компонент файлу F2, компонент файлу F1, компонент файлу F3, далі знову черговий компонент файлу F2 і т.д. Відсутні компоненти файлів F1, F2 і F3 замінюються відповідно значеннями: 1, -1 і 0 відповідно.

Варіант 4. Створити файл речовинних чисел (не менш 10 компонент).
Для даного файлу виконати наступні операції:

- знайти суму компонентів файлу;
- знайти добуток компонентів файлу;
- знайти суму квадратів компонентів файлу;
- знайти середній арифметичний і середнє геометричне компонентів файлу;
- знайти третю від кінця компоненту файлу.

Всі результати записати в текстовий файл.

Варіант 5. Створити файл речовинних чисел (не менш 15 компонент).
Для даного файлу виконати наступні операції:

- знайти найбільше зі значень компонентів з парними номерами;
- знайти найменше зі значень компонентів з непарними номерами;
- суму позитивних компонентів;
- різниця найбільшої по модулі компонента й найменшої по модулі;
- одержати новий файл, де компонента будуть розташовуватися у зворотному порядку.

Всі результати записати в текстовий файл.

Варіант 6. Дано файли F1, F2, F3, F4, F5, компоненти яких є речовинними числами. Організувати обмін компонентами між файлами у відповідності з наступною схемою: F1(F3, F2 (F4, F3 (F5, F4 (F2, F5 (F1, тобто компоненти файлу F1 листуються у файл F3, компоненти файлу F2 - в F4 і т.д. Дозволяється використати тільки один допоміжний файл G.

Варіант 7. Дано файл F, компоненти якого є цілими числами. Ніяка з компонентів файлу F не дорівнює нулю. Числа у файлі йдуть у наступному порядку: десять позитивних, десять негативних, десять позитивних, десять негативних і т.д. Число компонентів файлу повинне бути кратне 40. Переписати компоненти файлу F у файл G, щоб у файлі G числа йшли в наступному порядку: п'ять позитивних, п'ять негативних, п'ять позитивних, п'ять негативних і т.д.

Варіант 8. Дано файл F, компоненти якого є цілими числами. Ніяка з компонентів файлу F не дорівнює нулю. Числа у файлі йдуть у наступному порядку: десять позитивних, десять негативних, десять позитивних, десять негативних і т.д. Число компонентів файлу повинне бути кратне 40. Переписати компоненти файлу F у файл G, щоб у файлі G числа йшли в наступному порядку: вісім позитивних, вісім негативних, вісім позитивних, вісім негативних і т.д.

Варіант 9. Дано файл F, компоненти якого є цілими числами. Ніяка з компонентів не дорівнює нулю. Файл F містить рівна кількість позитивних і негативних чисел. Використовуючи допоміжний файл H, переписати компоненти файлу F у файл G так, щоб у файлі G не було двох сусідніх чисел з однаковим знаком.

Варіант 10. Дано файл F, компоненти якого є цілими числами. Ніяка з компонентів не дорівнює нулю. Файл F містить рівна кількість позитивних і негативних чисел. Використовуючи допоміжний файл H, переписати компоненти файлу F у файл G так, щоб у файлі G спочатку йшли позитивні, а потім негативні числа.

3. Зміст звіту

- 3.1. Тема та мета роботи.
- 3.2. Постановка завдання.
- 3.3. Текст програм.
- 3.4. Результати виконання програм.
- 3.5. Аналіз помилок та висновки по роботі.

3. Рекомендована література (основна, допоміжна), інформаційні ресурси в Інтернеті

Основна

1. Галісеєв Г.В. Системне програмування // Видавництво Університету “Україна”, 2018. – 253 с.
2. С.В. Єфіменко, О.В. Сугакова. Програмування: мови C і C++. - К.: ВПЦ "Київський університет", 2006 р.

Допоміжна

3. Шпак З. Я. Програмування мовою C.- Львів: Видавництво Львівська політехніка, 2011. – 436 с.
4. Ivor Horton's Beginning Visual C++. - Wiley Publishing, Inc., 2010 – 1272 p.

Інформаційні ресурси

1. <http://cppreference.com/>
2. <http://www.learncpp.com/>
3. <https://www.fluentcpp.com/>