

**Загальні зауваження**(читати уважно!).

NZ — це назва завдання — в кожного різна.

Проект на Сі та Сі++ складається з 3 (мінімум) програмних файлів для Сі-програми, та 4 для С++-програми: заголовочного файлу **NZ.h** та файлом модуля **NZ.c(cpp)**, де описані всі функції та типи даних та тестового файлу **NZ\_test.c(cpp)**, в якому викликаються всі функції модуля та перевіряються їх робота. При цьому дані повинні вводитися *вручну з консолі* **ТА** з *тестового текстового файлу або бінарного файлу* **NZ\_test.dat(txt)**, в якому в зрозумілій формі містяться дані, що тестують функції. При запуску тестового файлу пропонується обрати режим тесту – введення з консолі або введення даних з тестового файлу, після цього демонструється робота програми з виводом на консоль та виводом результату у файл. В Сі++ - програмі також створюється тестовий файл **NZ\_cross\_test.cpp** в якому викликаються відповідні функції на Сі та Сі++ та перевіряються, що результат у них ідентичний.

За потребою можна підключати інші файли в проект.

**( NZ- це назва завдання – це формат назви файлу)**

**Таким чином, для кожного завдання створюється дві директорії NZ\_C — для Сі проекту та NZ\_CPP для Сі++ проекту.**

**На початку кожного файлу — обов'язково коментар з вказанням задачі, групи та вашого прізвища, а також що саме знаходиться в цьому файлі.**

Для всіх створюваних структур та класів створити функції введення/виведення та можливість запису в файл читання з файлу.

Оцінюється окрім коректності та відповідності коду задачі та описаним вище умовам також: *акуратність коду, наявність потрібних коментарів, зрозумілість вводу/виводу, повнота тестових випадків, оптимальність алгоритмів.*

Передбачається реферативна частина (2-3 ст. або 5-20 в задачах на дослідження бібліотеки) з описом програмного доданку.

Курсивом – додаткова(необов'язкова підзадача).

Репозиторій:

<https://github.com/mecchmatProjects/C-Projects>

Створити та прогнати файл з тестовими даними для перевірки коректності роботи створених функцій. Коментувати код згідно зразків.

Оцінюватися будуть наступні фактори:

1. Файли, класи та методи підготовлені згідно описаного стилю та повністю відповідають умові
2. Коментарі та код написані згідно вказаного зразка
3. Тести присутні для кожної функції та методу класу
4. Код працює коректно для кожного описаного тесту
5. Повнота описаних тестових випадків

## Математика

1. (DTensor) створити клас – аналог тензору NumPy, за таким самим набором базових функцій (вирізка, решейп, модифікація групи елементів) в наступному вигляді:

number\_dim – 32-бітове натуральне число – кількість N розмірностей тензору;  
далі — вказівник на натуральні числа, - це масив з N чисел, величини першої, другої ... N-тої розмірності даного тензору.  
далі — вказівник на дійсні числа - масив даних тензору

Написати функцію, що змінює порядки координат в тензорі `changeCoord(int* coords, in num);` - масив номерів координат від 0 до N-1, та кількість координат (аналог транспонування)

Реалізувати для цієї структури арифметичні операції (виконуються коректно якщо коректні дані, в інших випадках — повідомляють помилку) та матричні дії (детермінант, обернена матриця).

## 2. Smatrix

Структура РОЗРІДЖЕНИЙ ВЕКТОР(SVector) зі стандартними операціями та РОЗРІДЖЕНА МАТРИЦЯ(Smatyrix) – розміри матриці n,m, та відповідний список чисел як словник(відображення) де ключ – x та y координати числа в матриці, а його значення це значення словника. Ті координати, що не ввійшли в словник вважаються як ті що містять нулі. Реалізувати стандартні операції (додавання, множення) для матриць та векторів, детермінант, знаходження оберненої матриці, множення на SVector, обмін рядків та колонок.

Передбачити введення векторів та матриць як рядок довільної довжини (відповідним списком даних) та через файл:

Int input(<Task>\* x); // через scanf по значенням коефіцієнтів – результат кількість коректно введених коефіцієнтів

Передбачити введення матриці через консоль як таблицю через пробіли  
int input(<Task>\* x, char\* str); // через рядок по значенням коефіцієнтів – результат кількість коректно введених символів

Виведення: виведення в на екран та в текстовий файл як виведення у рядок. Виведення в бінарний файл – як список коефіцієнтів зліва направо, зверху донизу.Smatrix

Передбачити функцію виведення що буде керувати точністю виведення та кількістю елементів у стовпчику та рядку, також довільним розміром рядків та стовпчиків.

Dtype det(SMatrix x);  
SMatrix inverse(SMatrix x);  
SVector solve(SMatrix x, SVector y);

Реферативна частина містить опис програми та алгоритмів.

### 3. OMG8.

Для кодування символів різних алфавітів використовується кодування UTF-8. На зміну цього кодування Вася Пупкін запропонував кодування OMG8. Кодування OMG8 кодує символ s, що може бути символ будь-якого алфавіту наступним чином:

- кожен символ може представлятися одним або декількома байтами (unsigned char) a[0],...,a[n];
- якщо останній біт з числа a[0] дорівнює 0, то s=a[0];
- інакше, якщо останній біт a[1] дорівнює 0, то s=(a[0])\*128+a[1];
- інакше, якщо останній біт a[2] дорівнює 0, то s=a[2]\*128^2+a[1]\*128+a[0] і т.д.

Таким чином ми можемо закодувати будь яку заздалегідь вказане натуральне число (а отже й будь-який символ).

Створити структуру OMG8String яка реалізує операції для роботи з рядком записаним в такій кодировці (стандартні операції крім ділення та множення на інший рядок: **введення\виведення, доступ до i-го**

**символу, видалення підрядку, вставка підрядку на дане місце, копіювання підрядку).**

Реалізувати функції над ціми рядками для пошуку підрядків.

Реалізувати функцію конкатенації рядків, функції визначення того, чи є цей рядок числом (у двійковій, десятковій та 16-річній системі числення) та переведення з кожної з цих форм до іншої. Структура повинна мати функції **для виводу та зміни значень** (заміна символів, додавання до кінця та початку нових символів, перевірка на коректність і т.п.)

Реферативна частина містить опис програми та алгоритмів.

#### 4. Interval

Створити структуру ІНТЕРВАЛ(Interval) – поля(type\_interval,a,b): два дійсних числа ( $a < b$ ), при цьому за допомогою значення поля type\_interval ми можемо вказати, що одно або обдва значення a,b є плюс- або мінус-нескінченостям.

Реалізувати методи: введення/виведення, перетин (як стандартне множення), об'єднання (як додавання) та різницю інтервалів.

Створити структуру(клас) SetIntevals (поля – список(або вектор чи масив) інтервалів та кількість інтервалів), що є об'єднанням декількох інтервалів (що не перетинаються) як масив ІНТЕРВАЛІВ. Реалізувати перетин(множення), об'єднання (додавання) та різницю цих списків, також довжину списку(сумарна довжина інтервалів).

Створити структуру (клас) КВАДРАТНА НЕРІВНІСТЬ (три дійсних числа та тип перерахування (більше, менше, рівне, нерівне, більше-дорівнює, менше-дорівнює)).

Розв'язати задачу – дано система квадратних нерівностей, отримується відповідь як SetIntevals.

Реферативна частина містить опис програми та алгоритмів.

#### 5. Напишіть клас або структуру які дозволяють ввести з консолі та файлу (текстового та бінарного) масив комплексних чисел.

Реалізуйте обернене та звичайне перетворення Фур'є. За допомогою нього розв'язати систему рівнянь  $\sum_{i=1}^{16383} A_{m-i} x_i = b_m$  для  $m=0..16383$ . Значення  $A_i$ ,  $i=0..16383$  для додатніх індексів дані у файлах (заповнюються випадковим чином але є можливість вводу), а для від'ємних обчислюються як  $A_i = A_{i+T}$ . Проведіть перевірку отриманих розв'язків. Вказівка: використовуйте перетворення Фур'є.

В реферативній частині - опис реалізація та методу швидкого перетворення Фур'є.

6. Створіть структури (класи) ТОЧКА(Point2D), ВІДРІЗОК(Segment2D), ПРЯМА(Line2D), ЛАМАНА(Polyline2D), БАГАТОКУТНИК(Polygone2D). Точка – це структура двох дійсних чисел( $x, y$ ), відрізок – дві точки( $a, b$ ), пряма задається 3 числами(Dtype  $a, b, c$ ), ЛАМАНА – масив точок, Багатокутник – це масив точок. Перезавантажити стандартні оператори (та створити аналогічні функції: тут додавання – це додавання векторів, множення – векторний та скалярний добуток (mult,multS). Реалізувати функції перетину відрізків (повертає точку або нічого), відрізка та прямих, прямих. Реалізувати функції визначення чи є багатокутник опуклим, чи немає в ламаній самоперетинів. Обчислити площу багатокутника. Тріангуляція багатокутника – виводить масив Triangle3D, які покривають трикутник.

#### 7. (Curve:Ellipse,Hyperbole, Parabole)

Створити структуру(клас ) ЕЛІПС – задається вектором 7 дійсних чисел(матрична форма) та/або координатами фокусів еліпса та 2 радіусами. Функції введення,виведення перетворення з однієї форми в іншу, площа еліпсу, знаходження координат центру еліпсу, чи знаходиться дана точка всередині еліпсу. Точки перетину Еліпсу з прямою та іншим еліпсом. Створити структуру(клас ) ПАРАБОЛА – задається вектором 3 дійсних чисел. Функції введення,виведення, знаходження координат центру параболі, чи знаходиться дана точка зверху над параболою, точки перетину параболі та ПРЯМОЇ, точки перетину 2-х ПАРАБОЛ, площі цих перетинів. Створити структуру(клас ) ГІПЕРБОЛА – задається вектором 5 дійсних чисел. Функції введення,виведення, знаходження координат центру гіперболи, чи знаходиться дана точка зверху над параболою, точки перетину параболі та ГІПЕРБОЛИ, точки перетину 2-х ГІПЕРБОЛ, площі цих перетинів.

### Дослідження стандартної бібліотеки

В даному циклі задач передбачається реферативна частина та створення програмного доданку що використовує функціонал описаний в реферативній частині.

Реферативна частина — від 5 сторінок, структурована по частинам, з повним описом стандартної бібліотеки, що описується (опис всіх відповідних класів, структур та функцій, які в цій бібліотеці є).

Крім того, в цій частині повинні міститись також опис програмного завдання та результати виконання їх програмного коду, порівняння швидкодії функціоналів, якщо це вказано в завданні та невеличкі висновки з цих результатів.

Програмний доданок містить заголовочний файл, де згідно прикладу описані функції й/або класи та методи, що потрібні у даному завданні та файл з релізацією та головну(драйвер) програму з тестами та кодом для виміру швидкості використання методів.

## 8. Комплексні числа на Cі та Cі++

- а) Створити власний клас Complex для комплексних чисел зі стандартними операціями на ньому (перевантаження операторів).
- б) Використати на Cі++ стандартну бібліотеку <complex>.
- в) Використати на Cі заголовочний файл <complex.h> та <tgmath.h> та стандартні комплексні типи Cі.

За допомогою формули для дискримінанта, формули Кардано та формули Ферарі розв'яжить, квадратне, кубічне та рівняння четвертого порядку з комплексними коефіцієнтами. Постарайтеся зробити якомога більш універсальне введення та виведення для даних задач.

Порівняйте швидкодію для цих задач для варіантів а)-в)

Реферативна частина — від 5 сторінок: опис роботи бібліотек <complex.h> та <tgmath.h> на Cі, опис бібліотеки <complex> на Cі++, результати порівнянь реалізацій.

## 9. Дата та час

- а) Реалізуйте задачу на Cі за допомогою структур або на Cі++ за допомогою класів.
- б) Реалізуйте задачу за допомогою **модуля time.h на Cі**
- в) Реалізуйте задачу за допомогою бібліотеки chrono на C++11**

Date

Створити тип Date - (рік, місяць, число) з функціями:

- Введення (як 3 цілих числа)

Наприклад, процес вводу:

"Рік: 1975

Місяць: 10

Число: 1"

При цьому дані перевіряються на коректність вводу.

- Виведення (формат rrrr.мм.чч та число(число) місяць (рядок) рік(число))

Наприклад, "1975.10.01" та "1 October 1975"



- Попереднє та наступне число для даної дати ( `Next(1976.12.31) = 1977.01.01`)
- Порівняння двох дат(яке раніше)
- Функція визначення дня тижня: наприклад, `week_day(2014.07.01) = "Wednesday"`
- Функція додавання та віднімання кількості днів до дати: `add_day(2014.07.21,11) = 2014.08.01`
- Функція визначення кількості днів між двома датами, та кількості днів від 0-го року до дати: `days(2014.01.21,2014.04.21) = 10 + 28 + 31 + 20 = 89`, `days(01.01.21) = 365 + 31 + 21 = 417` (будьте уважними з вісокосними роками!!!)
- Додавання дат та віднімання дат: `add_data(2014.07.01, 01.01.21) = add_day(2014.07.21,417)`

Створити тип `Time` - (доба,година, хвилина, секунда) з функціями:

- Введення (як 4 цілих числа)

Наприклад, процес вводу:

Доба: 5

Година: 10

Хвилина: 45

Секунда: 51"

При цьому дані перевіряються на коректність вводу.

- Виведення (формат `дд.гг.хх.сс` )

Наприклад, "5.10.45.51"

- Визначення кількості днів у `Time`, та перетворення з типу `Data` у `Time`
- Порівняння типу `Time`
- Функція додавання та віднімання `Time`: `add_time(1.7.21.11,0.8.3.52) = 1.15.25.3`
- Додавання та віднімання `Time` до дати: `add_timedata(2014.07.01, 01.01.21.32) = (2014.07.02)`

**Перевірте коректність а) б) в) порівнюючи результати між собою.**

Реферативна частина — від 5 сторінок: опис роботи бібліотеки `<time.h>` на Cі, опис бібліотеки `<chrono>` на Cі++, результати порівнянь реалізацій.

## 10. Бітові поля

а)Під час виконання цієї роботи для кожного завдання потрібно написати програму на мові Cі в якій виконано розробку структури з бітовими полями відповідно до постановкою завдання, введення даних елементів структури, їх обробка і виведення на екран в зазначеному форматі.

При цьому бітові поля можуть використовуватися тільки в якості полів структури. Введення даних здійснюється з клавіатури з урахуванням вимог до вхідних даних, що містяться в постановці завдання. Обмеженнями на вхідні дані є максимальна величина рядкових даних, діапазони числових типів полів структури, максимально дозволений розмір масиву структур в мові Сі

б) Замість бітової структури використовуйте колекцію `vector<bool>` зі стандартної бібліотеки

в) Замість бітової структури використовуйте псевдоконтейнер `bitset` зі стандартної бібліотеки

Задачі:

1) Розробіть структуру, використовуючи бітові поля для позначення логічних (бітових) значень. Структура Магазин: Найменування товару, Ціна, Країна виробництва, Наявність в магазині, Відповідність стандарту ЕУ, є товаром місяця. Оголосіть і заповніть масив з п'яти структур. Роздрукуйте інформацію про всі введені товари. Роздрукуйте інформацію про наявні в магазині товари, які не відповідають стандартам ЕУ.

2) Скористайтеся об'єднання з бітовими полями, визначте парність беззнакового цілого числа з аналізу старшого біта (операцію ділення із залишком не використовувати).

3) Використовуючи об'єднання з бітовими полями, визначте кратність 8 беззнакового цілого числа з аналізу старших бітів (операцію ділення із залишком не використовувати).

4) Використовуючи бітові поля створити тип для представлення дати та часу, що використовує мінімальну кількість пам'яті та реалізуйте метод різниця для часу.

**Порівняйте дані реалізації а), б), в) по швидкодії.**

Реферативна частина — від 5 сторінок: опис роботи бітових полів на Сі, опис псевдоконтейнерів `bitset` та `vector<bool>` на Сі++, результати порівнянь реалізацій.

## 11. Перерахування

а) Використайте в даних задачах структуру `enum` в стандартному Сі

б) Використайте в даних задачах клас `enum` з Сі++ (`enum class` з Сі++11) (в реферативній частині опишіть різницю між `enum` та `enum class`)

1) Створіть та реалізуйте за допомогою перерахування базові функції вводу-виводу для наступних сутностей:

- а) день тижня;
- б) місяць у році;
- в) колір спектру;



d) шахова фігура.

2) Опишіть тип – структуру Card для карти з колоди для преферансу. Для цього створіть перерахування Мاستь= {Піка, Трефи, Бубна, Чирва} та Ранг = {7,8,9,10, 'Jack', 'Queen', 'King', 'Ace'}. Реалізуйте логічну функцію beat(Card x, Card y, Масть z), що вказує чи бє перша карта другу, а третій параметр вказує яка масть є козирною.

3) Створіть перелік величин довжини (мм, см, дм, м, км) та реалізуйте функцію яка за введеною довжиною та величиною виміру виводить довжину в метрах.

4) Створіть перерахування Відмінок= {ім, бат, дат, ... } та за вказаним відмінком провідмінійте задані слова – програмування, мова, компютер.

5) Створіть перерахування Голосні, яке містить всі англійські(українськи) голосні та за допомогою цього типу визначить яка кількість складів в даному реченні (вважаючи, що склад містить лише одну голосну).

6) Створіть перерахування Course={N,S,W,E} та Order={Forward, Back, Left, Right}. В нас задано початковий курс корабля та масив команд як він рухався. Виведіть кінцевий напрямок корабля. Введіть також швидкість судна та масив дійсних чисел, що відповідає часу – скільки воно рухалося за даним курсом та за допомогою цих даних визначте на яку абсолютну відстань від початкової змістився корабель.

**Порівняйте дані реалізації а), б) по швидкодії.**

## 12. Об'єднання

а) Розв'яжить дані задачі на Сі за допомогою union

б) Розв'яжить дані задачі на Сі++17 за допомогою variant

1) Визначити універсальний тип, що дозволяє представляти точку на площині в декартовій та полярних координатах. Введіть дві точки та обчисліть довжину відрізка на даних точках.

2) Визначити універсальний грошовий тип, що може представляти вартість або в гривнях та копійках, або лише в копійках з методом, що дозволяє при цьому правильно відображати ті самі вартості.

- 3) Визначити універсальний тип, що дозволяє представляти вектор в як дві точки та як точку та вектор до другої точки. Введіть три вектори та з'ясуйте чи колінеарні вони.
- 4) Визначити універсальний тип, що дозволяє представляти точку в просторі в декартовій, полярній та сферичних координатах. Введіть дві точки та обчисліть довжину відрізка на даних точках.
- 5) Визначити тип Плоска Фігура, що включає Круг, Квадрат, Трикутник, Прямокутник, Трапеція. Реалізуйте функції обчислення периметру та площі фігури.
- 6) Визначте тип, що дозволяє зберігати число або будь-якого числового типу (double, int, unsigned,...) або рядки «Нескінченість» та «Невизначеність». Реалізуйте арифметичні операції для цього типу які коректно працюють з діленням та іншими операціями для всіх можливих комбінаціях значень та типів.

**Порівняйте дані реалізації а), б) по швидкодії.**

Реферативна частина — від 5 сторінок: опис об'єднань на Cі, опис псевдоконтейнеру variant

### 13. Командний рядок

а) Опишіть роботу з командним рядком Cі. Порівняйте за можливістю особливості цієї роботи для різних операційних систем (Windows/Linux/Android/MacOS). Зробіть парсинг командного рядочку функціями Cі.

б) Зробіть парсинг командного рядочку функціями стандартної бібліотеки STL Cі++.

1) Напишіть програму, що приймає з командного рядку 1 цілий аргумент та виведить його квадрат. Якщо аргументів 2 або більше, або жодного – виведіть повідомлення про помилку.

2) Напишіть програму, що приймає з командного рядку 3 дійсних аргументи та виводить їх середнє гармонічне. Якщо аргументів більше трьох, або менше – виведіть повідомлення про помилку. Якщо серед них є нуль — інше повідомлення про помилку.

3) Введіть з командного рядочку ім'я текстового файлу та підрахуйте кількість рядків в цьому файлі. Виведіть повідомлення про помилку якщо щось негаразд.

4) Введіть з командного рядочку ім'я декількох текстових файлів (їх повинно бути більше одного) та підрахуйте середню щільність символів на рядок в цих файлах.

5) Введіть через командний рядочок наступного вигляду:

`-filename name -rows rows ,`

імя файлу (name) та кількість рядків (rows),

параметри `-filename` та `-rows` – це обов'язкові літерали в рядку.

Якщо формат команди не такий як приведений вище – виведіть повідомлення про помилку та підказку. Якщо все вірно, створіть відповідний бінарний файл, що містить вказану кількість цілих чисел від 0 до rows.

6) Введіть через командний рядочок рядок наступного вигляду:

`-filename name -rows rows -cols cols`

імя файлу (name) та кількість рядків (rows) та стовпчиків (cols),

параметри `-filename` та `-rows, cols` – це обов'язкові літерали в рядку.

Якщо формат команди не такий як приведений вище – виведіть повідомлення про помилку та підказку. Якщо все вірно, створіть відповідний текстовий файл, що містить вказану кількість рядків заповнену cols нулями через табуляцію.

7) Введіть через командний рядочок рядок наступного вигляду:

`-filename1 name1 -filename2 name2 -rows,`

rows імя файлу (name) та кількість рядків (rows) параметри `-filename1` та `-filename2`, це обов'язкові літерали в рядку.

А параметр `-rows rows` може бути необов'язковий.

Якщо формат команди не такий як приведений вище – виведіть повідомлення про помилку та підказку. Якщо все вірно, порівняйте чи співпадає в даних двох файлах перші rows рядків з точністю до пробілів, якщо rows не вказаний – файли порівнюються повністю всі рядки.

Реферативна частина — від 5 сторінок: бібліотеки для роботи з командним рядком на C/C++

## 14. Змінні оточення

а) Розв'яжить дані задачі використовуючи бібліотеку `<env>` на C.

б) Розв'яжить дані задачі використовуючи бібліотеку `filesystem` (<https://en.cppreference.com/w/cpp/filesystem>) з C++17.

Задачі:

1) Напишіть функцію, яка визначає тип операційної системи даного комп'ютера.

2) Напишіть функцію, яка записує вміст даного файлу в новий файл, що знаходиться в системній директорії.

- 3) Напишіть функцію, що визначає чи існує в системі змінна PYTHON, та якщо немає, то встановлює цю змінну коректним шляхом.
- 4) Напишіть функцію, що визначає чи існує в системі змінна PYTHON3, а в поточній директорії файл file1.py (назва файлу вводиться з консолі) та якщо є, то запускає в консолі команду 'PYTHON3 file1.py'
- 5) За допомогою бібліотеки <fenv> зробіть можливість введення дійсних чисел в форматі з плаваючою крапкою, але замість крапки використовується кома.

Реферативна частина — від 5 сторінок: бібліотеки для роботи з системним оточенням на Cі/Cі++

## 15. Робота з valarray

Докладно опишіть класи та методи бібліотеки valarray.

- а) Реалізуйте задачу на Cі за допомогою виділення пам'яті вказівником
- б) Реалізуйте задачу за допомогою визначення типів, використовуючі valarray.
- в) Реалізуйте задачу за допомогою контейнера vector.

За допомогою бібліотеки псевдоконтейнера valarray реалізувати роботу зі структурою МАТРИЦЯ (див. проект Smatrix).

Порівняти швидкодії програм з реалізаціями, що виконані на базі вектору та звичайного масиву.

*За допомогою бібліотеки псевдоконтейнера valarray реалізувати роботу конволюції*

## 16. Регулярні вирази

Опишіть використання регулярних виразів на Cі++11.

- а) Розв'яжіть дані задачі на Cі за допомогою функцій бібліотеки string.h (без регулярних виразів)
- б) Розв'яжіть дані задачі на Cі за допомогою функцій бібліотеки <string>(без регулярних виразів)
- в) Розв'яжіть дані задачі на Cі за допомогою функцій бібліотеки <regex> (с++11)

1) В даному текстовому файлі знайти всі гіперпосилання та вивести їх разом зі словом (слово відокремлюється пробілами та знаками пунктуації) що стоїть перед ним.

2) Знайти в даному текстовому файлі всі дати в форматах \_\_.\_\_.\_\_ та \_\_/\_\_/\_\_ (в кінці можуть бути 2 або 4 цифри, а також \_\_\_\_ (рік).\_\_.\_\_) та перевести їх всі до єдиного формату \_\_ (рік) назва місяцю \_\_ (число).

3) Знайти всі цілі та дійсні числа (в науковому та звичайному форматі з крапкою/комою) та привести їх до єдиного формату (дійсне число з фіксованою крапкою, точність 4 розряду) та оточити круглими дужками.

4) Змінити всі дати та записи часу у файлі (див. Задачу 2) на поточні дату та час.

## 17. Робота з файлами 2

а) Розв'яжить дані задачі використовуючи бібліотеку `<env>` та функції роботи з файлами на Cі.

б) Розв'яжить дані задачі використовуючи бібліотеку `filesystem` (<https://en.cppreference.com/w/cpp/filesystem>) з Cі++17.

1. В даній директорії з підкаталогами відшукати всі файли з розширенням `*.c` та поміняти їх на відповідні файли з розширенням `*.cpp`

2. В даній директорії з підкаталогами відшукати всі файли з розширенням `*.cpp` та поміняти там коментарі вигляду `//` (до кінця рядку) на коментарі де в початку рядку `/* ...` та в кінці рядку `*/`

3. В даній директорії з підкаталогами відшукати всі файли з розширенням `*.txt` які модифіковані раніше заданої дати та видалити їх

4. В даній директорії з підкаталогами відшукати всі файли з розширенням `*.txt` які створені раніше ніж рік тому та перенести їх в іншу (задану) директорію

5. В даній директорії з підкаталогами відшукати всі файли з розширеннями Word які менше 100 кб та видалити їх.

6. Створити програму яка дозволяє ввести шлях до директорії (включаючи кириличні символи та враховуючі формати директорій різних ОС) та виводить середній розмір текстових файлів у цій директорії.

Реферативна частина — від 5 сторінок: бібліотеки для роботи з файлами та директоріями на Cі/Cі++.

## 18. Локалізація

Опишіть можливості локалізації на Cі (широкі символи, юнікод та бібліотеку `locale`), також бібліотеки `<locale>` та `<codecvt>` на Cі++.

а) Розв'яжить дані задачі на Cі використовуючі широкі символи та бібліотеку локалізації `<locale>`, `<wchar>`, `<uchar>`.

б) Розв'яжить дані задачі на Cі++ використовуючі бібліотеку локалізації `<locale>` та `<codecvt>`

Перевірте результат щонайменше на 2-х різних платформах.

1) Обчислити результати наступних виразів та вивести на екран напис українською мовою «Результат дорівнює:»:

$2+3$ ;  $4.5*56$ ;  $2/3.0$ .

2) Виведить напис : «Введіть ім'я:»

Введіть з нового рядка ваше ім'я (наприклад, «Вася» ) та виведіть привітання вигляду «Привіт, Вася!»

3) Введіть два цілих числа, що позначають грошовий тип — гривні та копійки та виведіть значення як грошовий тип в англійському, американському та українських локалізаціях.

4) Введіть два дійсних числа, які записані за допомогою десяткової коми та виведіть їх середнє геометричне в такому ж форматі.

5) Введіть дату в форматі (число, місяць, рік) та виведіть значення в німецьких, американських та українських локалізаціях.

## Робота з файлами різних форматів

### 19. CSV

Написати бібліотеку на Сі та клас на Сі++, що працюють з форматом CSV. Вони повинні містити функції(методи): створення файлу, читання файлу рядок за рядком (в класі — за допомогою власного ітератору), читання поточного рядку та рядку за заданим номером, читання поля за номером, заміна поточного поля, видалення поточного (заданого) рядку. Якщо вважати, що перший рядок це назва полів — то додати можливість визначення полів, читання поля за його назвою та номером рядку та виділення заданого поля з таблиці і додавання нового поля і даних для нього в таблицю. Зробіть також можливість додавати та модифікувати типи роздільників для CSV.

Реферативна частина містить опис формату CSV, шляхи відкриття її на Сі++ та опис вашої програми.

### 20. JSON/ XML

Написати бібліотеку на Сі та клас на Сі++, що працюють з форматами JSON та XML. Вони повинні містити функції(методи): створення файлу, знаходження всіх ключів, читання даних по данному ключу, ітерація по ключам та даним (за допомогою власного ітератору в класі), знаходження ланцюгу ключів при знаходженні даних, додавання даних у даний ланцюг, видалення та редагування даного запису.

Реферативна частина містить опис форматів JSON та XML, шляхи відкриття їх на Сі++ та опис вашої програми.

### 21. BMP/PNG



Написати бібліотеку на Cі та клас на Cі++, що працюють з форматами BMP та PNG. Вони повинні містити функції(методи): створення файлу, зміна кольорів даних пікселів, відображення (хуман-рідебл) інформації про файл, відображення (хуман-рідебл) вмісту файлу, модифікація кольору заданої групи символів (кола, прямокутника, відрізка) в файлі.

Реферативна частина містить опис форматів BMP/PNG, шляхи відкриття їх на Cі++ та опис вашої програми.

## 22. HSF

Файл бази даних з малюнками містить на початку ціле 32-бітне число 2051, потім ціле 32-бітне число – кількість малюнків, а наступні два 32-бітних числа – кількість пікселів висоту та ширину кожного малюнку у пікселях. При цьому ці числа задані в форматі high-indian (MSB first). Наступний вміст файлу – беззнакові натуральні байти ( $K \cdot n \cdot m$  байтів), кожен з яких – значення яскравостей пікселів (число від 0 до 255) кожного з цих малюнків, що проходяться у порядку зліва-направа та вгору-вниз.

Напишіть функцію, що перевіряє даний файл (заданий ім'ям) на відповідність даному формату, та виводить масив яскравостей малюнка з заданим номером, якщо такий номер та сам файл коректно задані. В протилежному випадку — вивести змістовне повідомлення про помилку (на Cі++ створіть виключення для цього).

Створіть клас та структуру на Cі, які можуть створювати та читати файли у такому форматі, при цьому користувач має можливість змінювати магічне число, тип порядку байтів, кількість малюнків та розміри, імена файлів і т. п. Зробіть можливість виведення малюнків в графічному вигляді.

Реферативна частина містить опис high-indian та інших архітектур та засоби їх обробки на Cі++.

## 23. IDX

Для представлення баз даних, що містять тензори часто використовують формат IDX (IDX file format), який має наступну форму:

magic\_number – 32-бітне число у форматі high-indian (MSB first), в якому перші 2 байти нулі, третій байт описує тип даних: якщо 0x08 -unsigned byte, 0x09 – signed byte, 0x0B – short(2 bytes), 0x0C – int (4 bytes), 0x0D – float (4 bytes), 0x0E – double (8 bytes), четвертий байт – кількість N розмірностей тензору;

size 1 - 32-бітне число у форматі high-indian (MSB first) величина першої розмірності;

size 2 - 32-бітне число у форматі high-indian (MSB first) величина другої розмірності;

\*\*

size N - 32-бітове число у форматі high-indian (MSB first) величина N-ої розмірності;

далі йдуть дані вказаного у першому числі формату:

data - Сі-масив даних у форматі high-indian (MSB first).

Напишіть функцію, що перевіряє даний файл (заданий ім'ям) на відповідність даному формату, та виводить координату тензору, що задана аргументом функції. В випадку, коли це не можливо, вивести змістовне повідомлення про помилку (на Сі++ створіть для цього власне виключення).

Створіть клас та структуру на Сі, які можуть створювати, читати та модифікувати дані з файлів (зміна значень заданих координатами, розмірностей, дописування або видалення заданих значень — не створюючи при цьому зайвих копій тензору в пам'яті, а лише для модифікованих даних) у такому форматі, при цьому користувач має можливість змінювати магічне число, тип порядку байтів, кількість малюнків та розміри, імена файлів і т. п. На Сі дані будуть вводиться динамічним масивом, а на Сі++ за допомогою vector та array.

Зробіть можливість виведення заданого своїм номером тензору у зрозумілому вигляді.

Реферативна частина містить опис роботи з тензорами на Сі++.

Статистика

24. Random1

Випадкові числа

Маємо натуральні  $a$ ,  $c$ ,  $m$ ,  $s_0$  такі, що  $m$  — найбільше з них. Визначемо послідовність натуральних чисел  $s_0, s_1, \dots, s_n$  наступним чином:  $s_n$  дорівнює  $s_n = a \cdot s_{n-1} + c \pmod{m}$ , та розглянемо послідовність  $r_n = (s_{n+1}) / (m+1)$ .

Послідовність  $r_i$  буде імітувати рівномірно розподілені в інтервалі  $(0, 1)$  випадкові числа, якщо:

- а)  $m=2^{**k}$ , де  $k$  — натуральне число;
- б) при діленні  $a$  на 8 — остаток дорівнює 5. Крім того,  $\sqrt{m} < a < m - \sqrt{m}$ ;
- в) Число  $c$  — непарне, при цьому бажано щоб  $\frac{s}{m} \approx 0.5 - \frac{\sqrt{3}}{6}$ ;
- г) Число  $s_0$  можна обрати довільно в діапазоні від 0 до  $m-1$ .

Завдання:

а) Створити функцію, що буде генерувати числа  $a$ ,  $c$ ,  $m$ ,  $s_0$ , що задовольняють вказаним умовам

б) створити на базі цієї послідовності генератор випадкових цілих чисел та генератор випадкових дійсних чисел

в) створити цей генератор таким чином, щоб він генерував майже завжди різні числа при першому виклику (визначаєте нове  $s_0$ ) та враховував попередні виклики при нових викликах (використуйте статичні глобальні змінні та хеш часу)

г) Отримайте цим датчиком 1000 чисел та оцініть рівномірність розподілу: розбийте інтервал  $(0, 1)$  на  $N$  інтервалів рівної довжини та знайдіть варіацію серед чисел, що туди потрапили.

Описані методи повинні бути описаними в заголовочному файлі на Cі та як методи класу на Cі++(клас відповідно містить приватні члени для  $a$ ,  $c$ ,  $m$ ,  $s_0$ ). В тестовій програмі перевірте зокрема коректність ГВЧ за критерієм Хі-квадрат.

За допомогою цього генератора побудувати послідовність випадкових величин, що задовольняють розподіл:

а) Пуассона

б) Гауса

в) експоненційного

г) Стюдента

д) Фішера

На Cі++ порівняйте отримані генератори з відповідними ГВЧ модуля `<random>`

Реферативна частина містить огляд бібліотек ГВЧ для Cі++.

## 25. Random-2

Маємо натуральні  $a$ ,  $c$ ,  $m$ ,  $s_0$  такі, що  $m$  — найбільше з них. Визначимо послідовність натуральних чисел  $s_0, s_1, \dots, s_n$  наступним чином:  $s_n$  дорівнює  $s_n = a \cdot s_{n-1} + c \pmod{m}$ , та розглянемо послідовність  $r_n = (s_n + 1)/(m + 1)$ .

Послідовність  $r_i$  буде імітувати рівномірно розподілені в інтервалі  $(0, 1)$  випадкові числа, якщо:

а)  $m = 2^k$ , де  $k$  — натуральне число;

б)  $c$  та  $m$  — взаємно прості числа;

в)  $(a-1)$  ділиться на всі прості числа  $p$ , що є дільниками  $m$

г)  $b \cdot (a-1)$  ділиться на 4, якщо  $m$  ділиться на 4.

Завдання:

а) Створити функцію, що буде генерувати числа  $a$ ,  $c$ ,  $m$ ,  $s_0$ , що задовольняють вказаним умовам

б) створити на базі цієї послідовності генератори випадкових цілих чисел та генератор випадкових дійсних чисел

в) створити цей генератор таким чином, щоб він генерував майже завжди різні числа при першому виклику (визначаєте нове  $s_0$ ) та враховував попередні виклики при нових викликах (використуйте статичний член структури або класу та *та хеш часу*)

г) на базі даного ГВЧ створить метод, що генерує  $n$ -вимірні випадкові вектори дійсних чисел, кожні дві координати яких є некорельовані.

Напишіть код, який перевіряє на достатньо великій вибірці, що вони дійсно некорельовані.

д) На базі методу Монте-Карло підрахуйте об'єм  $n$ -вимірної сфери.

Описані методи повинні бути описаними в заголовочному файлі на Сі та як методи класу на Сі++(клас відповідно містить приватні члени для  $a$ ,  $c$ ,  $m$ ,  $s_0$ ). В тестовій програмі перевірте зокрема коректність ГВЧ за критерієм Хі-квадрат.

За допомогою цього генератору побудувати послідовність випадкових величин, що задовольняють розподіл:

а) Пуассона

б) Гауса

в) експоненційного

г) Стюдента

д) Фішера

На Сі++ порівняйте отримані генератори з відповідними ГВЧ модуля `<random>`.

Реферативна частина містить опис критеріїв перевірки ГВЧ.

Різне

## 26. Server

Реалізувати клієнт-серверну систему, що дозволяє клієнтам посилати повідомлення на сервер (поток байтів та можливість посилати широкі символи `wchar`). Повідомлення також містять ідентифікатори інших клієнтів, і сервер якщо бачить підєднаншого відповідного клієнта пересилає йому ці повідомлення. Проект передбачає текстовий опис роботи програми (документація) та реферат по створенню клієнт/серверу на Сі/Сі++ .

## 27. Plant

Створити версії програми на Сі (за допомогою структур та класів) та Сі++(за допомогою шаблонів та класів):

Створити на базі абстрактного класу ЛЮДИНА класи ПРАЦІВНИК, спадкоємці КЕРІВНИК, СУМІСНИК. Створити на базі даних класів

систему обліку зарплати та обладнання на заводі. Зарплата працівників залежить від базової ставки професії працівника, стажу, видобутку ним продукції. Класи ВИРОБ, ПРОФЕСІЯ та ОБЛАДНАННЯ також створити. Методи введення/виведення зробити як з/на екран так і в/з файлу (формат файлу CSV). Обчислити вартість обладнання працівників, що відносяться до даного керівника.

Реферативна частина містить опис програми з діаграмами UML класів.

## 28. University

Створити версії програми на Сі (за допомогою структур та класів) та Сі++(за допомогою шаблонів та класів):

Створити на базі абстрактного класу ЛЮДИНА класи ПРАЦІВНИК, ВИКЛАДАЧ, СТУДЕНТ. Створити на базі даних класів систему обліку зарплати та стипендії в ВУЗі. Зарплата працівників, викладачів залежить від базової ставки професії працівника, стажу, ступеня. Стипендія студента залежить від його середнього балу. Класи ЗАЛІКОВКА, ПРОФЕСІЯ також створити. Методи введення/виведення зробити як з/на екран так і в/з файлу (формат JSON). Дані про викладачів зберігають в цих файлах та доданок має інтерфейс для введення модифікації цих файлів та їх аналізу. Обчислити середній бал студентів, що навчаються у даного викладача.

Реферативна частина містить опис програми з діаграмами UML класів.

## 29. Shop

Створити версії програми на Сі (за допомогою структур та класів) та Сі++(за допомогою шаблонів та класів):

Створити сім'ю класів ЛЮДИНА – ПОКУПЕЦЬ, ПРОДАВЕЦЬ – КЕРІВНИК МАГАЗИНУ. Про моделювати процес продажу-купівлі класу ТОВАР за допомогою випадкового процесу купівлі для даного МАГАЗИНУ. Закупівля товару також йде випадковим чином. Врахувати обмеженість складу товару та обсягу обороту. Дані по складу МАГАЗИНУ та по працівникам і покупцям зберігаються в бінарних файлах певного формату. Для роботи з цим форматом потрібно створити окрему бібліотеку, що передбачає для користувача можливість гарно працювати з цією БД (модифікація, видалення, додавання даних). Введення/виведення даних з екрану та з файлу.

Зарплата Продавця залежить від кількості проданого ним товару та отриманого ним прибутку. Оптовим та постійним Покупцям є можливі знижки, що повинні враховуватися. Для деяких видів товару вартість знижується (уцінка) з часом. Обчислити прибуток/збиток магазину за деякий час. Показати стан МАГАЗИНУ (список товарів в наявності, продано та отримано за невеличкий проміжок часу, список продавців та їх здобутки).

Реферативна частина містить опис програми з діаграмами UML класів.

### 30. Bank

Створити версії програми на Сі (за допомогою структур та класів) та Сі++(за допомогою шаблонів та класів):

Промоделювати роботу БАНКу, що приймає ВКЛАДи від КЛІЄНТІВ та за купує й продає ЦІННІ ПАПЕРИ. ЦІННІ ПАПЕРИ змінюють свої ціну випадковим чином (але в середньому лінійно зростають кожен рік). Процес продажу паперів випадковим чином робиться. Типи паперів та види їх продажу різні. Клієнти так само випадково роблять різні Вклади та знімають з них гроші (процент, тривалість вкладу, можливість та дати зняття грошей залежать від типу вкладу). Дані зберігаються в файлах формату CSV та користувач має можливість модифікувати ці дані. Оцінити роботу банку (прибуток) за певний період часу. Введення/виведення на екран та в файл. Показати стан БАНКУ (гроші та акції в наявності, скільки видано та отримано грошей, паперів за невеличкий проміжок часу).

Реферативна частина містить опис програми з діаграмами UML класів.

### 31. Лінгвістика

Як показують численні експерименти, розбиття українського слова на частини для переносу з одного рядка на іншу з великою ймовірністю виконується правильно, якщо користуватися наступними простими прийомами:

**1.** Дві підряд голосні можна розділити, якщо першій з них передують приголосна, а за другою йде хоча б одна буква (буква «й» при цьому розглядається разом з попередньою голосною як єдине ціле).

**2.** Дві йдуть підряд приголосні можна розділити, якщо першій з них передують голосна, а в тій частині слова, яка йде за другою приголосною, є хоча б одна `голосна (буква ь разом з попередньою приголосною розглядаються як єдине ціле).

**3.** Якщо не вдається застосувати пункти 1), 2), то слід спробувати розбити слово так, щоб перша частина містила більш ніж одну букву і закінчувалася на голосну, а друга містила хоча б одну голосну. Ймовірність правильного розбиття збільшується, якщо попередньо використати хоча б неповний список приставок, що містять голосні, і спробувати перш за все виділити зі слова таку приставку. Дано текст, який є українським словом. Виконати поділ його на частини для переносу.

Створити клас СЛОВО, який містить наступні члени:

*Тип (дієслово, іменник, прикметник, і т.п.)*

*Рід,*

*Число,*



## Особа

### Відмінок

Префікс (може містити від 0 до 3 префіксів)

Корінь(може містити від 1 до 2 коренів)

Суфікс (може містити від 0 до 3 суфіксів)

Закінчення (може містити від 0 до 1 закінчень)

Методи: введення/виведення в/з консолі та в/з файл.

Розбиття слова на слоги.

Злиття слогів у слово.

Методи введення слова по частинах: префікс, корінь, суфікс та закінчення

Методи виведення частин слова.

Установка властивостей слова (тип, рід, число, відмінок).

Зміна властивості – тобто перевод слова в інше число та відмінок.

Наслідники класу СЛОВО: ДІЄСЛОВО, ІМЕННИК, ПРИКМЕТНИК, ЧИСЕЛЬНИК і т.п.

Використовувати тут роботу з широкими символами(wide char).

## 32 Convolution

На базі структури (класу) Tensor реалізувати методи конволюцій та оберених конволюцій, що використовуються при машинному аналізі зображень та розпізнаванні. При цьому ці конволюції влючають роботу з щонайменше чотирма варіантами різних паддінгів (Якщо не вистачає знань по теорії — звертатись до викладача).

## Ігри

### 33. SeaBattle-1

Створити гру МОРСЬКИЙ БІЙ у наступній варіації. Поле 10\*10 (можна задати і інші розміри) заповнюється стандартним чином. Постріли робляться залпами по N (N задається на початку гри та замовченням рівно 3, N=1 – класичний морський бій) та відповідь отримується по всьому залпу (Приклади: a1,a2,a3 - K1,W1 (1 поранено, 1 вбито); b1, c5,c6 – W2 (2 поранено), d3,d4,d5 – All fail (всі мимо) ). Гравці кожен час мають бачити всі результати та робити нотатки на власній копії карті (можна псевдографікою). Розробіть варіанти гри двох суперників на 1-му комп'ютері, гри з комп'ютером, що стріляє випадково (але не повторюючись) та гри з розумним комп'ютером, що має непогану стратегію гри. Плюс за клієнт-серверний варіант.

Реферативна частина містить опис програми, алгоритмів та стратегій гри.

#### 34. SeaBattle-2

Створити гру МОРСЬКИЙ БІЙ у наступній варіації. Поле 10\*10 (можна задати і інші розміри) заповнюється лише одним однопалубним кораблем. Постріли робляться одиничні та у якості відповіді повідомлюється відстань (манхеттонська, тобто L1) до корабля. Після пострілу супротивника гравець повинен/може рухати корабель на сусідню позицію (N4 або N8 – за домовленістю) (але не може бути там, де вже був постріл супротивника). Перемагає той, хто перший потопив корабель супротивника. Гравці кожен час мають бачити всі результати та робити нотатки на на власній копії карти (можна псевдографікою). Розробіть варіанти гри двох супротивників на одному комп'ютері, гри з комп'ютером, що стріляє випадково (але не повторюючись) та гри з розумним комп'ютером, що має непогану стратегію гри. Плюс за клієнт-серверний варіант.

Реферативна частина містить опис програми, алгоритмів та стратегій гри.

#### 35. FLAT

клас КВАРТИРА складається зі списку КІМНАТ та КОРИДОРІВ (ПРИМІЩЕННЯ)

КІМНАТА визначається координатами СТІН та ДВЕРЕЙ.

СТІНА - (x1,y1),(x2,y2), isExit (bool, number of the doors), ширина

ДВЕРІ - (x1,y1),(x2,y2), матеріал

МЕБЕЛЬ - список координат

ДІЯ - список координат шагів та поворотів

РОБОТ - поточні координати (x,y), action

Target - координати або ПРИМІЩЕННЯ.

ШЛЯХ - послідовність ДІЙ.

Знайти найкоротший шлях від початку до ЦІЛІ. Визначити в якому Приміщенні знаходиться робот та ЦІЛЬ.

Скласти розподіл розмірів оптимальних шляхів до цілі в залежності від координат

Нехай РОБОТ випадково обирає шлях, але при цьому слідкує, щоб кожен крок був якнаймога далі від тих місць де він недавно був. Знайдіть середній результат успіху в залежності від кроків.

**Приклад оформлення коду:**

Задача 1. Реалізувати список точок та певні функції для них:

Файли в директорії CPointsTask:

TPoints.h

TPoints.c

TPoints\_test.c

TPoints\_testfile.dat

**TPoints.h:**

```
/* File TPoints.h
Realization of Points structure and list for it (Task 1)
Done by Ivanov Ivan (group 3.2)
Date 10.12.2014
Headers for Point struct and list
*/
/**** headers
.....

/* Base structure of point */

typedef struct TPoint { double x; double y};

// .... More structures

/* functions for the Points */
/* Input point
input point from console
@param[out] - OUT – Point for return
@result – int :
    0 - in the case of success,
    error_code – number of error
*/
extern int InputPnt( TPoint *x);
```

```

/* Output point:
   write point to console
   @param - IN Point
   */
extern void OutputPnt( TPoint x);

.....

// list element
typedef struct list_node {
    struct list_node *next;
    TPoint data;
} *list_node;

// realization of list
typedef struct list {
    list_node head;
} *list;

/*****/
/* functions for this list */

/* first function – delete element
   you should have a corresponding acc_list_node_create() function
   @param - IN element of list
   @result - void
   */
extern void list_node_destroy(list_node node)

/* 2 function – create element
   @param - IN element of list
   @result – 0 if element created, 1 -if no
   */

extern int list_node_create(list_node node)
{
    node = malloc(sizeof(list_node));
    return (node!=NULL);
}

```

```
}  
***** more functions declarations
```

## **TPoints.c**

```
/* File TPoints.c  
Realization of Points List structure functions  
Done by Ivanov Ivan (group 3.2)  
Date 10.12.2014  
Implementations of TPoint and List  
*/
```

```
// function of removing list element  
void list_node_destroy(list_node node) {  
    free(node);  
}
```

```
// function of creating list element  
int list_node_create(list_node node){  
    node = malloc(sizeof(list_node));  
    return (node!=NULL);  
}
```

```
***** more functions realization
```

## **TPoints\_test.c**

```
/* File TPoints_test.c  
Realization of tests for Points List structure functions  
Done by Ivanov Ivan (group 3.2)  
Date 11.12.2014
```

```
Tests for Tpoint and List  
*/
```

```
/* test 1 – create List */  
int list_test1 ()  
{
```

```
    list lst;
```

```

list_allocate(&lst);
list_init(lst);
lst = list_add(lst,(char*)"aaaa\0");
lst = list_add(lst,(char*)"bbbb\0");

    If (lst.get() != "bbbb\0") return 1;
    Lst.pop();
    If (lst.get() != "aaaa\0") return 2;
    Lst.pop();
    return (!(lst==NULL));
}

```

*/\* test 2 – create List with file \*/*

```

int list_test2 ()
{
    list lst;
    FILE* fp;
    TPoint x;
    double s1=0;
    double s2=0;

    list_allocate(&lst);
    list_init(lst);

    fopen(fp,"Pts_testfile.dat","rt");

    while (!feof(fp))
        { fread(fp,&x,1,sizeof(TPoint));
          Lst.push(x);}
//check the equality of the sum
    While(lst){
        S1 +=lst.get().x;
        S2 +=lst.get().y;
        lst.pop();
    }
// we know the sum of coordinates in file and check it */
    If ((s1==SUM1)&&(s2==SUM2)) return 0

```



```
Else return 1;
```

```
}
```

```
/** more tests on functions .....
```

```
int main()
```

```
{ unsigned i,n; TPoint x; list lst;
```

```
.....
```

```
  If (list_test1() ==0) printf("1 test passed");
```

```
  If (list_test2() ==0) printf("2 test passed");
```

```
...
```

```
/// and here the possibility to check input and output
```

```
printf("Enter the number of points");
```

```
pcanf("%u", &n);
```

```
for(i=0;i<n;i++){
```

```
    input(&x);
```

```
    lst.push(x);
```

```
}
```

```
while (lst){
```

```
    output(lst.get());
```

```
    lst.pop();
```

```
}
```

```
return 0;
```

```
}
```