

Лабораторна робота №7.Тема: **Контейнери стандартної бібліотеки шаблонів мови C++. Зрізи.****Методичні вказівки****1. Контейнер.**

Контейнер — це об'єкт, який вміщує інші об'єкти. Як правило, в контейнер можна добавляти та вилучати об'єкти, а також переглядати вміст контейнера. Контейнери стандартної бібліотеки мови C++ були побудовані у відповідності з такими критеріями (умовами): забезпечення максимуму свободи при побудові власних контейнерів, в той же час надання користувачеві загального інтерфейсу роботи зі стандартними контейнерами, що дозволяє вивільнити програміста від повторного написання існуючого коду. Це дозволяє оптимізувати як час розробки проекту, також оптимізувати його код.

Приклад 1. Використовуючи контейнер **list** розробити програму, яка вводить інформацію про студентів групи, яка складається з полів (прізвище, ім'я та рік народження), впорядковує за прізвищем та записує у файл StudRec.dat в прямому порядку, а у – RStudRec.dat оберненому порядку та дублюючи виведенням на екран.

```
#include "stdafx.h"
#include <string>
#include <fstream>
#include <iostream>
#include <list>
using namespace std;
class MyRec
{
    string fname, name;
    int year;
public:
    MyRec() {};;
    string GetName() const { return name; };
    void SetName(string n) { name = n; }
    string GetFName() const { return fname; };
    void SetFName(string n) { fname = n; }
    int GetYear() const { return year; };
    void SetYear(int n) { year = n; }
    bool operator < (MyRec s)
    {
        return fname>s.fname;
    }
};

int main(int argc, char* argv[])
{
    cout << "Hello !!! Testing list \n" << endl;

    string yes = "Y";
    list<MyRec> mylist;
    do
    {
        MyRec temp;
        string fn, n;
        int y;
        cout << " Input fname "; cin >> fn; temp.SetFName(fn);
        cout << " Input name "; cin >> n; temp.SetName(n);
        cout << " Input year "; cin >> y; temp.SetYear(y);

        mylist.push_back(temp);

        cout << " If you want input press Y/N \n";
```

```

        cin >> yes;
    } while (yes == "Y");
    mylist.sort();
    list<MyRec>::iterator mIter;
    ofstream os("StudRec.dat");
    mIter = mylist.begin();
    cout << endl;
    while (mIter != mylist.end())
    {
        MyRec temp;
        temp = *mIter;
        os << " Fname =" << temp.GetFName() << "\t";
        os << " Name =" << temp.GetName() << "\t";
        os << " Year =" << temp.GetYear() << "\n";

        cout << " Fname =" << temp.GetFName() << "\t";
        cout << " Name =" << temp.GetName() << "\t";
        cout << " Year =" << temp.GetYear() << "\n";
        mIter++;
    }

    list<MyRec>::reverse_iterator rIter;

    rIter = mylist.rbegin();
    cout << " Reverse " << endl;
    os.close();
    os.open("RStudRec.dat");
    while (rIter != mylist.rend())
    {
        MyRec temp;
        temp = *rIter;
        os << " Fname =" << temp.GetFName() << "\t";
        os << " Name =" << temp.GetName() << "\t";
        os << " Year =" << temp.GetYear() << "\n";

        cout << " Fname =" << temp.GetFName() << "\t"; cout << " Name =" <<
temp.GetName() << "\t";
        cout << " Year =" << temp.GetYear() << "\n";

        rIter++;
    }

    cin >> yes;
    return 0;
}

```

2. Зрізи мови C++

Шаблон класів **valarray**

Цей шаблон розроблений для оптимізації обчислень, які виконуються над масивами чисел фіксованого розміру. Шаблон **valarray** схожий на контейнер, але він їм не є. Ви не можете динамічно та ефективно нарощувати його розмір. Він, як і контейнер, може змінювати свої розміри, використовуючи метод **resize**, але при цьому наявні дані руйнуються. Головною перевагою використання **valarray** є ефективність проведення операцій відразу над всіма елементами послідовності. Припустимо, ви хочете побудувати графіка функції $y = \sin(x)$ і маєте процедуру, що зробить це із урахуванням масштабування, цифрування осей і всяких інших зручностей. Вашою задачею є лише сформувати дані для графіка й подати їх на вхід цієї процедури. Використання **valarray** дасть перевага в легкості маніпулювання даними й ефективності виконання.

Зрізи масивів

Проблеми оптимізації роботи з матрицями давно хвилюють творців компіляторів. У той далекий час, коли рішення задач електродинаміки та взагалі крайових задач математичної фізики ще використовувались мови PL/I або FORTRAN, звичайно ж, зберігали та обробляли матриці в одномірних масивах. Справа в тому, що вибір одного елемента з більше природного для матриць двомірного масиву обходився дорого. Виробилася особлива техніка роботи з одномірними масивами, що зберігати матриці (звичайно розріджені). У мові C++ операція вибору елемента із двомірного динамічного масиву не набагато дорожче, ніж з одномірного(та й швидкості змінилися), тому гострота проблеми спала. Проте проблема економії часу при рішення складних крайових задач не пішла в минуле.

STL має пари допоміжних класів: **slice** та **gslice**, які створені для того, щоб було зручно працювати зі зрізами (перетинами) одномірних масивів. Якщо ви зберігаєте двомірну матрицю в послідовності типу **valarray**, то елементи одного рядка матриці або одного її стовпця можна представити у вигляді зрізу, тобто певної частини всієї послідовності. Конструктор класу *slice* визначає закономірність, відповідно до якої будуть вибиратися елементи послідовності, щоб утворити зріз. Наприклад, об'єкт *slice s(0, n, 2)*; являє собою перетин з *n* елементів послідовності. Елементи вибираються починаючи з нульового, через один, тобто із кроком 2. Якщо ви зберігаєте матрицю $n \times n$ у послідовності типу **valarray** і при цьому вона впорядкована по рядках (спочатку перший рядок, потім друга, і т.д.), то третій рядок матриці можна вибрати за допомогою зрізу – *slice s(2*n, n, 1)*. Дійсно, параметри вказують, що треба пропустити $2*n$ елементів, потім вибрати *n* елементів із кроком по одному. Якщо матриця зберігається а за правилами мови FORTRAN, тобто по стовпцях, то для вибору того ж рядка треба визначити перетин – *slice s(2, n, n)*, тобто пропускаються два елементи, потім вибирається *n* елементів із кроком *n*.

Приклад 2. Створити масив цілих чисел 1 до 20. Організувати вибрати з масиву 7 елементів починаю з 2 елемента з кроком 3.

```
#include "stdafx.h"
#include <valarray>
#include <iostream>

int main( )
{
    using namespace std;
    int i;

    valarray<int> va ( 20 ), vaResult;
    for ( i = 0 ; i < 20 ; i++ )
        va [ i ] = i + 1 ;

    cout << "The operand valarray va is:\n( ";
    for ( i = 0 ; i < 20 ; i++ )
        cout << va [ i ] << " ";
    cout << ")." << endl;

    slice vaSlice ( 1 , 7 , 3 );
    vaResult = va [ vaSlice ];

    cout << "\nThe slice of valarray va is vaResult:"
        << "\nva[slice( 1, 7, 3)] = ( ";
    for ( i = 0 ; i < 7 ; i++ )
        cout << vaResult [ i ] << " ";
    cout << ")." << endl;
}
```

The operand valarray va is:
(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20).

The slice of valarray va is vaResult:
 va[slice(1, 7, 3)] = (2 5 8 11 14 17 20).
 Press any key to continue . . .

Приклад 3. Створити масив цілих чисел 0 до 99. Здійснити узагальнений зріз масиву починаю з 0 елемента, організувавши 4 групи, які розташовані через 19 елементів, у кожній підгрупі, ще 5 під-підгруп, які розташовані через 6 елементів, а кожній під-підгруп 4 елементи, які розташовані з кроком 2.

```
#include "stdafx.h"
#include <valarray>
#include <iostream>
int main( )
{
    using namespace std;
    int i;

    valarray<int> va ( 100 ), vaResult;
    for ( i = 0 ; i < 100 ; i++ )
        va [ i ] = i;

    cout << "The operand valarray va is:" << endl << "(";
    for ( i = 0 ; i < 100 ; i++ )
        cout << va [ i ] << (!( (i+1)%25) || (i==99) )?"\n":" ");
    cout << ")" << endl;

    valarray<size_t> Len ( 3 ), Stride ( 3 );
    Len [0] = 4;
    Len [1] = 5;
    Len [2] = 3;
    Stride [0] = 19;
    Stride [1] = 6;
    Stride [2] = 2;

    gslice vaGSlice ( 0, Len, Stride );
    vaResult = va [ vaGSlice ];

    cout << "The valarray for vaGSlice is vaResult:" << endl
        << "va[vaGSlice] = (\n";

    for ( i = 0 ; i < vaResult.size() ; i++ )
    { cout << " " << vaResult [ i ];
      if((i+1)%Len [2]==0) cout<<endl;
      if((i+1)%(Len [2]*Len[1])==0) cout<<endl;
    }
    cout << ")" << endl;
}
```

The operand valarray va is:
 (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99)
 The valarray for vaGSlice is vaResult:
 va[vaGSlice] = (

```

0 2 4
6 8 10
12 14 16
18 20 22
24 26 28

19 21 23
25 27 29
31 33 35
37 39 41
43 45 47

38 40 42
44 46 48
50 52 54
56 58 60
62 64 66

57 59 61
63 65 67
69 71 73
75 77 79
81 83 85
)
Press any key to continue . . .

```

Приклад 4. Задано ціле число $k > 2$ та масив дійсних чисел $v = (v_0, v_1, \dots, v_{N-1})$ де

$N = 50k + 10k^2$. Обчислити $G = \sum_{j=0}^4 g_j$, де g вектор $(g_0, g_1, g_2, g_3, g_4)$, компоненти якого

визначаються наступним чином: $g_0 = \sum_{i=1}^k v_{4i}$, $g_j = \sum_{i=0}^{j-1} \sum_{s=0}^{k+j-1} \sum_{l=0}^{2k-j-1} \sin(v_{(3k+j)i+(k+1+j)s+l+3})$ $i=1,4$;

$g_1 = g_1 - g_3$; $g_2 = g_2 - g_4$.

```

#include "stdafx.h"
#include <valarray>
#include <iostream>
#include <complex>
using namespace std;
int main(int argc, char* argv[])
{
    int k,N,i;
    do {
        cout<<" Input k (k>2) \n";
        cin>>k;
    } while (k<=2);
    N=50*k+10*k*k;
    valarray< complex<double> > v(N);
    char rnd ='y';
    do {
        cout<<" Press - \'y \' for ramdom input v \n";
        cout<<" Press - \'k \' for keyboard input v \n";
        cin>>rnd;
    }
}

```

```

} while (! ( rnd == 'y' || rnd == 'k' ));

if(rnd == 'k')
{
    for(i=0;i<N;i++)
    {
        cout<<" Input Re Im ";
        double re,im;
        cin>>re>>im;
        v[i].real(re); v[i].imag(im);
    }

    } else for(i=0;i<N;i++)
        v[i]=complex<double>(rand()/3333.3,rand()/3333.3);

cout<<" V is \n";
cout.scientific;
for(i=0;i<N;i++)
{ cout<<v[i]<<" \t"; if((i+1)%3==0) cout<<endl; }

valarray< complex<double> > g(5);
    slice s(1,k,4);
    cout<<" \n t1 \n";
    valarray< complex<double> > g0 = v[s];
    for(i=0;i<g0.size();i++) { cout<<g0[i]<<" \t"; if((i+1)%3==0) cout<<endl;
}

    cout<<endl;

g[0] = g0.sum();
cout<<" \n Rezult \n g[0] "<<g[0];
// g[1]-g[4]
for (int j=1;j<5;j++)
{
    valarray <unsigned int> l(3),d(3);
    l[0] =j; l[1] =k+j; l[2] =2*k-j; // length
    d[0] =3*k+j; d[1] =k+1+j; d[2] =3; // step
    gslice gs(0,l,d);
    valarray< complex<double> > gj = v[gs];
    cout<<" \n gj.size() \n" <<gj.size()<<endl;
    for(i=0; i<gj.size() ; i++ ) { cout<<gj[i]<<"\t"; if((i+1)%3==0)
cout<<endl; }
    cout<<endl;
    // for valarray< double > or valarray< double > defined operator gj=sin(gj)
    complex<double> wr,wr2;
    for(i=0;i<gj.size();i++) { wr=gj[i]; wr2=sin(wr); gj[i]=wr2; }
    cout<<" \n sin(gj) \n" ;
    for(i=0; i<gj.size() ; i++ ) { cout<<gj[i]<<"\t"; if((i+1)%3==0) cout<<endl;
}

    cout<<endl;
    g[j] = gj.sum();
    cout<<" \n g["<<j<<"] "<<g[j]<<endl;
    cin>>rnd;
}

cout<<" g \n";
for (int j=0;j<5;j++)
    cout<<g[j]<<"\n";
    cout<<endl;

    slice s1(1,2,2),s2(2,2,2);
    g[s1]-=g[s2];

cout<<" g \n";
for (int j=0;j<5;j++)
    cout<<g[j]<<"\n";

```

```

        cout<<endl;

        complex<double> G = g.sum();

        cout<<" \n Rezult \n G "<<G<<endl;

cin>>rnd;
return 0;
}

```

Завдання до лабораторної роботи:

Розробити та реалізувати класи згідно варіантів використанням STL. Передбачити введення початкових даних: з клавіатури та файлу.

Варіант	Завдання 1	Завдання 2
1	Задача 1.1	Задача 11
2	Задача 1.2	Задача 12
3	Задача 2.1	Задача 13
4	Задача 2.2	Задача 14
5	Задача 3	Задача 15
6	Задача 4	Задача 16
7	Задача 5.1	Задача 17
8	Задача 5.2	Задача 18
9	Задача 6	Задача 19
10	Задача 7	Задача 20
11	Задача 1.3	Задача 21
12	Задача 1.4	Задача 22
13	Задача 8	Задача 11
14	Задача 9	Задача 12
15	Задача 10	Задача 13
16	Задача 1.6	Задача 14
17	Задача 1.7	Задача 15
18	Задача 2.3	Задача 16
19	Задача 2.4	Задача 17
20	Задача 2.5	Задача 18
21	Задача 2.6	Задача 19
22	Задача 2.7	Задача 20
23	Задача 2.8	Задача 21
24	Задача 2.9	Задача 22

Задачі.

- Використовуючи контейнер **list** розробити програму, яка вводять інформацію про студентів групи, яка складається з полів (прізвище, ім'я, по-батькові, рік народження та п'ять оцінок за останню сесію), впорядковує за *полем згідно варіанту* та записує у файл StudRec.dat в прямому порядку, а у – RStudRec.dat оберненому порядку та дублюючи виведенням на екран. Варіанти полів впорядкування:

- 1) ім'я;
- 2) по-батькові;
- 3) рік народження;

- 4) перша оцінка;
 - 5) третя оцінка;
 - 6) ім'я та перша оцінка;
 - 7) друга та четверта оцінка.
2. Використовуючи контейнер **vector** розробити програму, яка вводить інформацію про студентів групи, яка складається з полів (прізвище, ім'я, по-батькові, рік народження та п'ять оцінок за останню сесію), впорядковує за *полем згідно варіанту* та записує у файл StudRec.dat в прямому порядку, а у – RStudRec.dat оберненому порядку та дублюючи виведенням на екран. Варіанти полів впорядкування:
- 1) ім'я;
 - 2) по-батькові;
 - 3) рік народження;
 - 4) перша оцінка;
 - 5) третя оцінка;
 - 6) ім'я та перша оцінка;
 - 7) друга та четверта оцінка;
 - 8) прізвище;
 - 9) п'ята оцінка та ім'я.
3. Використовуючи контейнер **set** розробити програму. В програмі створити файл test.dat в який записати 100 випадкових цілих чисел з діапазону 1 до 100. Написати функцію, читає дані із файлу test.dat, будує множину чисел, знаходить добуток елементів множини, результат записує у файл test.out та дублює – на екран.
4. Використовуючи контейнер **set** розробити програму. В програмі створити файл test.dat в який записати 2000 випадкових цілих чисел з діапазону 200 до 400. Написати функцію, читає дані із файлу test.dat, будує множину чисел, знаходить суму елементів множини, результат записує у файл test.out та дублює – на екран.
5. Використовуючи контейнер **list** розробити програму. В програмі створити файл test.dat в який записати 100 випадкових цілих чисел з діапазону 1 до 100. Написати функцію, читає дані із файлу test.dat, перетворює за дані за певним законом (закон перетворення задається згідно варіанту) та записує перетворені дані у файл test.out та дублює – на екран. Перетворити за наступний чином:
- 1) у вихідний файл записати так числа, щоб чергувалися парні та непарні числа із вхідного файлу. Відносний порядок розташування парних чисел між собою зберегти. При побудові, може залишитися декілька чисел. Числа, які залишилися розташувати в кінець списку.
 - 2) у вихідний файл записати так числа, щоб чергувалися по два парні та три непарні числа із вхідного файлу. Відносний порядок розташування парних чисел між собою зберегти. При побудові, може залишитися декілька чисел. Числа, які залишилися розташувати в кінець списку.
 - 3) у вихідний файл записати так числа, щоб чергувалися по три числа з діапазону від 1 до 50 та по чотири числа з діапазону від 51 до 100. Відносний порядок розташування у діапазоні чисел між собою зберегти. При побудові, може залишитися декілька чисел. Числа, які залишилися розташувати в кінець списку.
6. Задано текст українською (англійською) мовою в форматі ASCII / UNICODE. Використовуючи контейнер **map** побудувати довідник кількості входження кожного слова в текст.
7. Задано текст українською (англійською) мовою в форматі ASCII / UNICODE. Використовуючи контейнер **set** побудувати множину слів, впорядкувати у алфавітному порядку та роздрукувати.
8. Задано текст українською (англійською) мовою в форматі ASCII / UNICODE. Використовуючи контейнер **set** побудувати множину речень, впорядкувати у алфавітному порядку за першим словом та роздрукувати.

9. Задано текст українською (англійською) мовою в форматі ASCII / UNICODE. Використовуючи контейнер **list** побудувати лінійний список елементами якого будуть слова списку. Впорядкувати у алфавітному порядку та роздрукувати.
10. Задано текст українською (англійською) мовою в форматі ASCII / UNICODE. Використовуючи контейнер **list** побудувати лінійний список елементами якого будуть речення тексту. Впорядкувати у алфавітному порядку речення тексту та роздрукувати.

11. Задано ціле число $k > 2$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 10k + 2k^2$, $k > 2$. Обчислити вектор $g = (g_1, g_2, g_3)$, де $g_1 = \sum_{i=1}^k v_{4i}$, $g_2 = \sum_{i=1}^k \sin(v_{4i+1}) + \cos(v_{4i})$, $g_3 = \sum_{i=1}^{k-2} \sum_{j=1}^{k-1} \sum_{l=1}^k v_{1+l+4i+3j}$.
12. Задано ціле число $k > 3$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 9k$. Обчислити вектор $g = (g_1, g_2, g_3, g_4, g_5)$, де $g_1 = \prod_{i=1}^{3k} v_{3+2i}$, $g_j = \sum_{i=1}^k \sum_{l=1}^k \sin(v_{j+i*6+l}) + \cos(v_{2*j+7i+l})$.
13. Задано ціле число $k > 2$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 7k$. Обчислити $G = \sum_{j=1}^k \sum_{i=1}^k g_i^T \cdot g_j$, де вектори $g_i = (v_{i+3k}, v_{i+2k}, v_{i+3k}, v_{i+k}, v_{i+4k}, v_{i+5k})$, $i = \overline{1, k}$.
14. Задано масив(вектор) дійсних / **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 4k$. Обчислити вектор $g = (g_1, g_2, g_3, g_4)$, де $g_1 = \sum_{i=1}^k v_{4i}$, $g_2 = \sum_{i=1}^k \sin(v_{4i+1}) + \cos(v_{4i})$, $g_3 = g_1 \cdot \sum_{i=1}^k \operatorname{tg}(v_{4i+2}) + \operatorname{ctg}(v_{4i})$, $g_4 = \sum_{i=1}^k v_{2i} + \sum_{i=1}^k v_{3i} + \sum_{i=1}^k v_{4i}$.
15. Задано ціле число $k > 2$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 9k$. Обчислити вектор $g = (g_1, g_2, g_3, g_4, g_5)$, де $g_1 = \prod_{i=1}^{3k} v_{3+2i}$, $g_j = \sum_{i=1}^k (v_{j+2k} + v_{j+3k} + v_{j+7k}) + \sum_{i=1}^k (\sin(v_{j \cdot i}) + \cos(v_{(9-j) \cdot i}))$.
16. Задано ціле число $k > 3$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 7k$. Знайти вектори $g_i = (v_{i+3k}, v_{i+2k}, v_{i+3k}, v_{i+k}, v_{i+4k}, v_{i+5k})$, $i = \overline{1, k}$ та обчислити $G = \prod_{j=1}^k \sum_{i=1}^k g_i^T \cdot g_j$.
17. Задано ціле число $k > 5$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 15k$. Побудувати вектори $g_0 = (v_3, \dots, v_{9k})$, $g_j = (v_j, v_{j+3}, \dots, v_{j+3k}, v_{10+j}, v_{10+j+3}, \dots, v_{10+j+3k}, v_{20+j}, v_{20+j+3}, \dots, v_{20+j+3k})$ $j = \overline{1, 3}$, $g_4 = g_0 + g_1$, $g_5 = g_3 - g_2$, $g_6 = g_0 / k$. Обчислити $G = \sum_{j=1}^6 \max g_j - \min g_{j-1}$.
18. Задано ціле число $k > 3$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел $v = (v_1, v_1, \dots, v_N)$ де $N = 9k$. Знайти вектори $g_i = (v_{i+k}, v_{i+2k}, v_{i+3k}, v_{i+4k}, v_{i+5k}, v_{i+6k})$, $i = \overline{1, 3k}$,

перерахувати компоненти векторів $g_j, j = \overline{1, k}$, наступним чином

$$g_j(l) = g_j(l) + g_{j+k}(l) - g_{j+2k}(l-1), \quad j = \overline{1, k}, \quad l = \overline{2, 4, 6}; i$$

$$g_j(l) = g_j(l) + g_{j+2k}(l) - g_{j+k}(l+1), \quad j = \overline{1, k}, \quad l = \overline{1, 3, 5}; \text{ та обчислити}$$

$$G = \prod_{j=1}^k \sum_{i=1}^k \max g_i - \min g_j.$$

19. Задано ціле число $k > 4$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел

$$v = (v_0, v_1, \dots, v_{N-1}) \text{ де } N = 40k + 9k^2. \text{ Обчислити } G = \sum_{i=0}^4 g_j, \text{ де } g \text{ вектор } (g_0, g_1, g_2, g_3, g_4),$$

$$\text{компоненти якого визначаються наступним чином: } g_0 = \sum_{i=1}^k v_{4i},$$

$$g_i = \sum_{i=0}^{k-j-1} \sum_{s=0}^{k+j-1} \sum_{l=0}^{k-j-1} \cos(v_{(2k+j)i+(k+1+j)s+l+1}) \quad i = \overline{1, 4}; \quad g_1 = g_1 - g_3; \quad g_2 = g_2 + g_4.$$

20. Задано ціле число $k > 4$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел

$$v = (v_0, v_1, \dots, v_{N-1}) \text{ де } N = 40k + 9k^2. \text{ Обчислити } G = \max g - \min g, \text{ де } g \text{ вектор}$$

$$(g_0, g_1, g_2, g_3, g_4), \text{ компоненти якого визначаються наступним чином: } g_0 = \sum_{i=1}^k v_{4i},$$

$$g_j = \sum_{i=0}^{k-j-1} \sum_{s=0}^{k+j-1} \cos(v_{(5k+j)i+s+1}) + \sin(v_{(4k-j)i+s+2}) \quad j = \overline{1, 2};$$

$$g_j = \sum_{i=0}^{k-j-1} \sum_{s=0}^{k+j-1} \cos(v_{(4k-j)i+s+1}) + \sin(v_{(5k+j)i+s+2}) \quad j = \overline{3, 4}.$$

21. Задано ціле число $k > 4$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел

$$v = (v_0, v_1, \dots, v_{N-1}) \text{ де } N = 40k + 9k^2. \text{ Обчислити } G = \sum_{i=0}^4 g_j / \max g, \text{ де } g \text{ вектор}$$

$$(g_0, g_1, g_2, g_3, g_4), \text{ компоненти якого визначаються наступним чином:}$$

$$g_j = \sum_{i=1}^k v_{j+2(i+j)}, \quad j = \overline{0, 1}, \quad g_j = \sum_{i=0}^{k-j-1} \sum_{s=0}^{k+j-1} \sum_{l=0}^{k-j-1} \cos(v_{(2k+j)i+(k+1+j)s+l+1}) \quad j = \overline{2, 4}.$$

22. Задано ціле число $k > 3$ та масив дійсних/ **цілих, натуральних, комплексних** / чисел

$$v = (v_0, v_1, \dots, v_{N-1}) \text{ де } N = 30k + 7k^2. \text{ Обчислити } G = \sum_{i=0}^6 g_i / (\max g - \min g + 1), \text{ де } g \text{ вектор}$$

$$(g_0, g_1, g_2, g_3, g_4, g_5, g_6), \text{ компоненти якого визначаються наступним чином:}$$

$$g_0 = \sum_{i=1}^k v_{3i+1} + 1, \quad g_j = \sum_{i=0}^{k-j-1} \sum_{s=0}^{k+j-1} v_{(5k+j)i+s+1}^2 - v_{(5k+j)s+i+2} \quad j = \overline{1, 6}.$$