Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до виконаної лабораторної роботи № 7

з дисципліни

“Дослідження операцій”

на тему:

**«Задачі динамічного програмування.»**

Виконав

студент групи *ОІ-11 сп*

*Вальчевський П. В.*

Викладач:

*Сенета М. Я.*

Львів – 2024

## Лабораторна робота № 7

*Тема роботи:*  **Задачі динамічного програмування.**

*Мета роботи:***навчитися моделювати багатокрокові процеси та знаходити розв’язок задач оптимального управління для організаційних економічних систем.**

***Завдання***

* + Номер таблиці відповідає номеру студента в журналі групи старости: таблиця 1 -1,7,13,19,25; таблиця 2 -2,8,14,20,26; **таблиця 3 -3,9,15,21,27**; таблиця 4 -4,10,16,22,28; таблиця 5 -5,11,17,23,29; таблиця6 -6,12,18,24,30.
  + Значення ’х’ визначається крайнім правим розрядом номера залікової книжки студента:

0: х=0; **1- х=5**; 2: х=10; 3: х=15; 4: х=20; 5: х=25; 6: х=30; 7: х=35; 8: х=40; 9: х=45.

* + Розподілити наявні кошти між підприємствами, щоб отримати найбільший сумарний прибуток.

***Порядок виконання роботи***

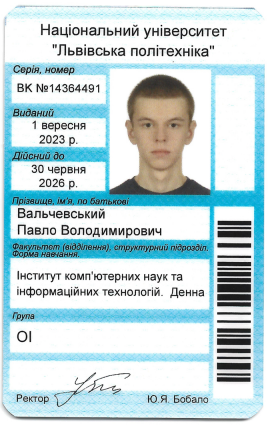
1. **Номер варіанту – 91.**
2. ****

Рис. 1 Фото мого студентського квитка для варіанту ЛР.

1. **Умова завдання:**

Я є 3 у списку у журналі, тому моя таблиця № 3 з методички і маю крайній правий розряд у студентському квитку – 1, тому моє х = 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | **Прибутки підприємств**, тис.грн. | | | |
| **f1** | **f2** | **f3** | **f4** |
| 0 | 0 | 0 | 0 | 0 |
| 10 | 5+x | 6+x | 7+x | 5+x |
| 20 | 15+x | 14+x | 14+x | 10+x |
| 30 | 22+x | 20+x | 20+x | 18+x |
| 40 | 28+x | 27+x | 28+x | 27+x |

Таблиця № 1. Таблиця мого порядкового номера з методички (Таблиця № 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | **Прибутки підприємств**, тис.грн. | | | |
| **f1** | **f2** | **f3** | **f4** |
| 0 | 0 | 0 | 0 | 0 |
| 10 | 10 | 11 | 12 | 10 |
| 20 | 20 | 19 | 19 | 15 |
| 30 | 27 | 25 | 25 | 23 |
| 40 | 33 | 32 | 33 | 32 |

Таблиця № 2 Сформована таблиця згідно параметрів для мого завдання.

**Розв’язок вручну**

Виділення коштів для першого підприємства:

|  |  |
| --- | --- |
| f1(c) | x1\*(c) |
| 0 | 0 |
| 10 | 10 |
| 20 | 20 |
| 27 | 30 |
| 33 | 40 |

Розподіл коштів між двома першими підприємствами:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | X | | | |  | | |
| **0** | **10** | **20** | **30** | **40** | f2(c) | x2\*(c) |
| 0 | 0 + 0 | - | - | - | - | 0 | 0 |
| 10 | 0 + 10 | 11 + 0 | - | - | - | 11 | 10 |
| 20 | 0 + 20 | 11 + 10 | 19 + 0 | - | - | 21 | 10 |
| 30 | 0 + 27 | 11 + 20 | 19 + 10 | 25 + 0 | - | 31 | 10 |
| 40 | 0 + 33 | 11 + 27 | 19 + 20 | 25 + 10 | 32 + 0 | 39 | 20 |

Розподіл коштів між трьома першими підприємствами:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | X | | | |  | | |
| **0** | **10** | **20** | **30** | **40** | f3(с) | x3\*(c) |
| 0 | 0 + 0 | - | - | - | - | 0 | 0 |
| 10 | 0 + 11 | 12 + 0 | - | - | - | 12 | 10 |
| 20 | 0 + 21 | 12 + 11 | 19 + 0 | - | - | 23 | 10 |
| 30 | 0 + 31 | 12 + 21 | 19 + 11 | 25 + 0 | - | 33 | 10 |
| 40 | 0 + 39 | 12 + 31 | 19 + 21 | 25 + 11 | 33 + 0 | 43 | 10 |

Розподіл коштів між чотирма першими підприємствами:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | X | | | |  | | |
| **0** | **10** | **20** | **30** | **40** | f4(c) | x4\*(c) |
| 0 | 0 + 0 | - | - | - | - | 0 | 0 |
| 10 | 0 + 12 | 10 + 0 | - | - | - | 12 | 0 |
| 20 | 0 + 23 | 10 + 12 | 15 + 0 | - | - | 23 | 0 |
| 30 | 0 + 33 | 10 + 23 | 15 + 12 | 23 + 0 | - | 33 | 0 |
| 40 | 0 + 43 | 10 + 33 | 15 + 23 | 23 + 12 | 32 + 0 | 43 | 0 |

Оптимальний розв’язок:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вкладення коштів**, тис.грн. | f1(c) | x1\*(c) | f2(c) | x2\*(c) | f3(с) | x3\*(c) | f4(c) | x4\*(c) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 10 | 10 | 11 | 10 | 12 | 10 | 12 | 0 |
| 20 | 20 | 20 | 21 | 10 | 23 | 10 | 23 | 0 |
| 30 | 27 | 30 | 31 | 10 | 33 | 10 | 33 | 0 |
| 40 | 33 | 40 | 39 | 20 | 43 | 10 | 43 | 0 |

У перше підприємство вкладаємо 20 тис. грн.

У друге підприємство вкладаємо 10 тис. грн.

У третє підприємство вкладаємо 10 тис. грн.

У четверте підприємство вкладаємо 0 тис. грн.

Максимальний прибуток буде дорівнювати 43 тис. грн.

**Програмна реалізація**

#include <iostream> // Для виводу

#include <limits> // Для обробки очистки та даних вводу, якщо користувач вводить не число

#define m 11

using namespace std;

int mas[m][5], ms[m][8], n, Q\_0, k;

void inputDataDeposits();

void calculateDepositProfits(int depositIndex);

void findMaxProfitsAndDeposits();

void updateDepositInBusiness(int step);

int main()

{

cout << "Pavlo Valchevskyi, OI-11sp, LR # 7, variant 91, Doslidzhennya operatsiy" << endl;

inputDataDeposits();

printf("\n");

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

calculateDepositProfits(i);

}

printf("\n");

findMaxProfitsAndDeposits();

printf("\n");

system("pause");

}

/\*

Функції:

inputDataDeposits() - функція для введення вхідних даних, таких як кількість депозитів, початкова сума для депозиту та крок збільшення депозиту.

updateDepositInBusiness(int step) - функція для оновлення інформації про депозити у бізнесі залежно від кроку.

calculateDepositProfits(int depositIndex) - функція для розрахунку прибутку для кожного депозиту. Розрахунок виконується за допомогою деяких внутрішніх механізмів, і дані зберігаються у двовимірному масиві ms.

findMaxProfitsAndDeposits() - функція для знаходження максимального прибутку та депозитів, що його забезпечують.

Змінні, які використовуються:

mas[m][5]: Це двовимірний масив розміром m на 5, де m визначається як 11 за допомогою макросу #define m 11. Цей масив використовується для зберігання даних про депозити. Кожен рядок у цьому масиві представляє один депозит, а кожний стовпець - певний параметр депозиту.

ms[m][8]: Це ще один двовимірний масив, також розміром m на 8. Він використовується для зберігання додаткових даних, пов'язаних з розрахунками прибутку для кожного депозиту.

n: Це змінна, яка визначає кількість депозитів, введених користувачем.

Q\_0: Це початкова сума для депозитів, яка введена користувачем.

k: Це крок збільшення депозитів, також введений користувачем.

i, j, p, dod, max, s: Змінні для ітерації та тимчасового зберігання значень (використовуються у функціях).

\*/

void inputDataDeposits()

{

int j = 0;

cout << "Input number of deposits ( <10 )" << endl << "\tn - ";

while (!(cin >> n) || (n <= 1 || n >= 10)) {

cout << "\tInvalid input. Please enter number of deposits ( <10 )\tn - ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cout << "Enter start sum for deposit (default - 0)" << endl << "\tQ\_0 - ";

while (!(cin >> Q\_0) || Q\_0 < 0) {

cout << "\tInvalid input. Please enter number start sum for deposit (default - 0)\tQ\_0- ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

mas[0][0] = Q\_0;

cout << "Input step add to deposit" << endl << "\tk - ";

while (!(cin >> k) || k <= 0) {

cout << "\tInvalid input. Please enter number step add to deposit\tk- ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

updateDepositInBusiness(k);

cout << "\nEnter business profits" << endl;

cout << "f\_0 f\_1 f\_2 f\_3" << endl;

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

cin >> mas[i][1] >> mas[i][2] >> mas[i][3] >> mas[i][4];

}

}

void updateDepositInBusiness(int step)

{

for (int i = 1; i < n; i++)

{

mas[i][0] = mas[i - 1][0] + k;

}

}

void calculateDepositProfits(int depositIndex)

{

int max, i, dod, j, p, s;

max = -1;

if (depositIndex == 0)

{

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

{

ms[i][0] = mas[i][1];

ms[i][1] = Q\_0 + i \* k;

}

}

if (depositIndex == 1)

{

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

if (i == 0)

{

ms[i][2] = mas[0][1];

ms[i][3] = mas[0][1];

}

else

{

p = i;

for (j = 0; p >= 0; j++, p--)

{

dod = mas[j][2] + ms[p][0];

if (dod > max)

{

max = dod;

s = j;

}

}

ms[i][2] = max;

ms[i][3] = k \* s;

max = -1;

}

}

max = -1;

if (depositIndex == 2)

{

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

if (i == 0)

{

ms[i][4] = mas[0][2];

ms[i][5] = mas[0][2];

}

else

{

p = i;

for (j = 0; p >= 0; j++, p--)

{

dod = mas[j][3] + ms[p][2];

if (dod > max)

{

max = dod;

s = j;

}

}

ms[i][4] = max;

ms[i][5] = k \* s;

max = -1;

}

}

max = -1;

if (depositIndex == 3)

{

for (j = 0, p = n; j <= n; j++, p--)

{

dod = mas[j - 1][4] + ms[p][4];

if (dod > max)

{

max = dod;

s = j;

}

}

ms[n - 1][6] = max;

ms[n - 1][7] = k \* (s - 1);

}

if (depositIndex == 4) {

cout << "Final table:\n";

}

else cout << "Step " << depositIndex + 1 << ":\n";

printf("L1\tx1\tL2\tx2\tL3\tx3\tL4\tx4\n");

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

for (int j = 0; j < 8; j++) {

cout << ms[i][j] << "\t";

}

printf("\n");

}

cout << "\n";

}

void findMaxProfitsAndDeposits()

{

int s, i, j, p, sum;

sum = 0;

s = ms[n - 1][7];

p = mas[n - 1][0];

if (s != 0)

{

sum += mas[n - 1][4];

printf("In the fourth business, the enterprise deposits %d \* 10 ^ 3 uah.\n", s);

}

p -= s;

for (i = 0; i < 100; i++)

if (i \* k == p)

break;

s = ms[i][5];

if (s != 0)

{

sum += mas[i][3];

printf("In the third business, the enterprise deposits %d \* 10 ^ 3 uah.\n", s);

}

p -= s;

for (i = 0; i < 100; i++)

if (i \* k == p)

break;

s = ms[i][3];

if (s != 0)

{

sum += mas[i][2];

printf("In the second business, the enterprise deposits %d \* 10 ^ 3 uah.\n", s);

}

p -= s;

for (i = 0; i < 100; i++)

if (i \* k == p)

break;

s = ms[i][1];

if (s != 0)

{

sum += mas[i][1];

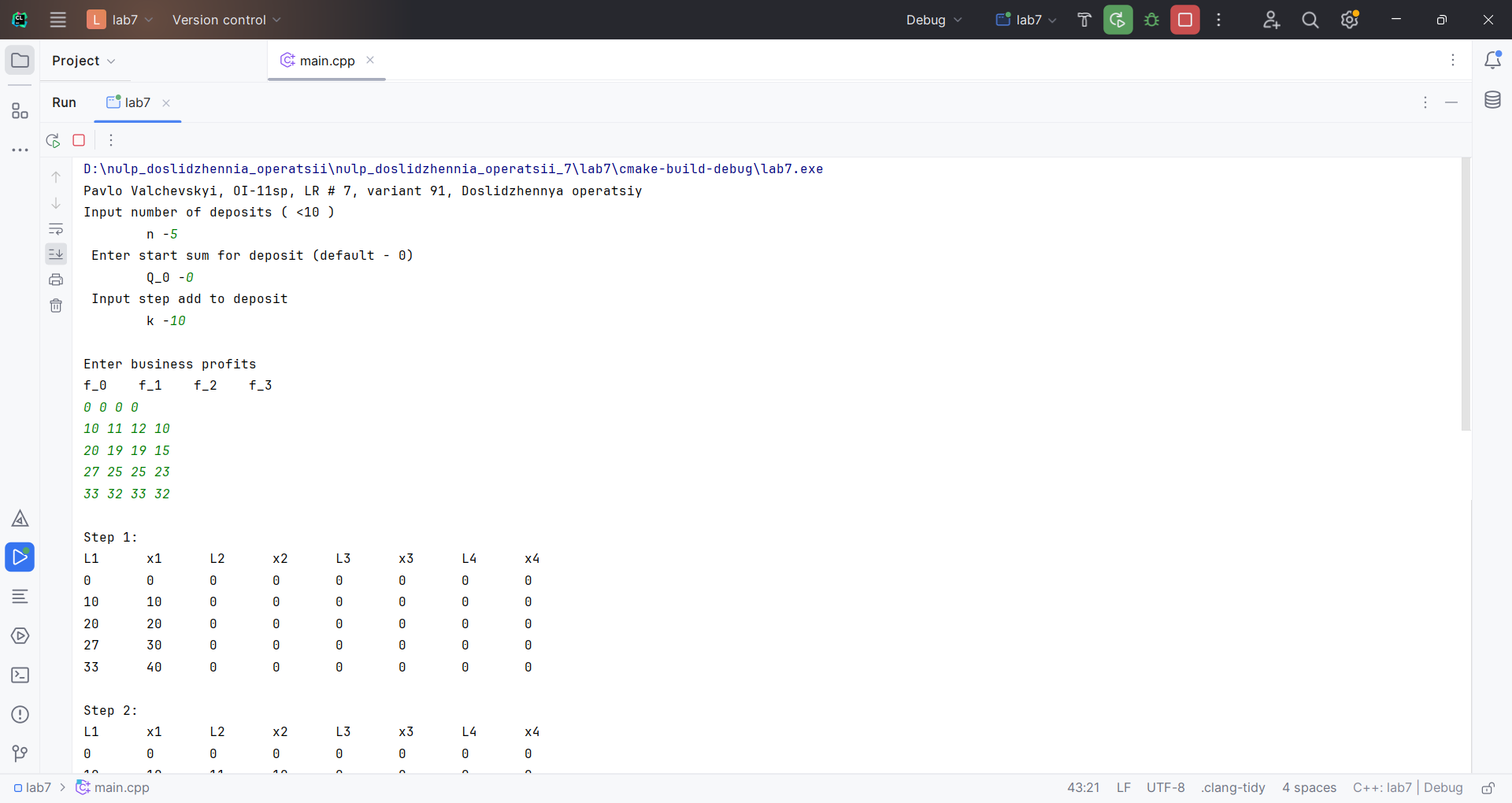
printf("In the first business, the enterprise deposits %d \* 10 ^ 3 uah.\n", s);

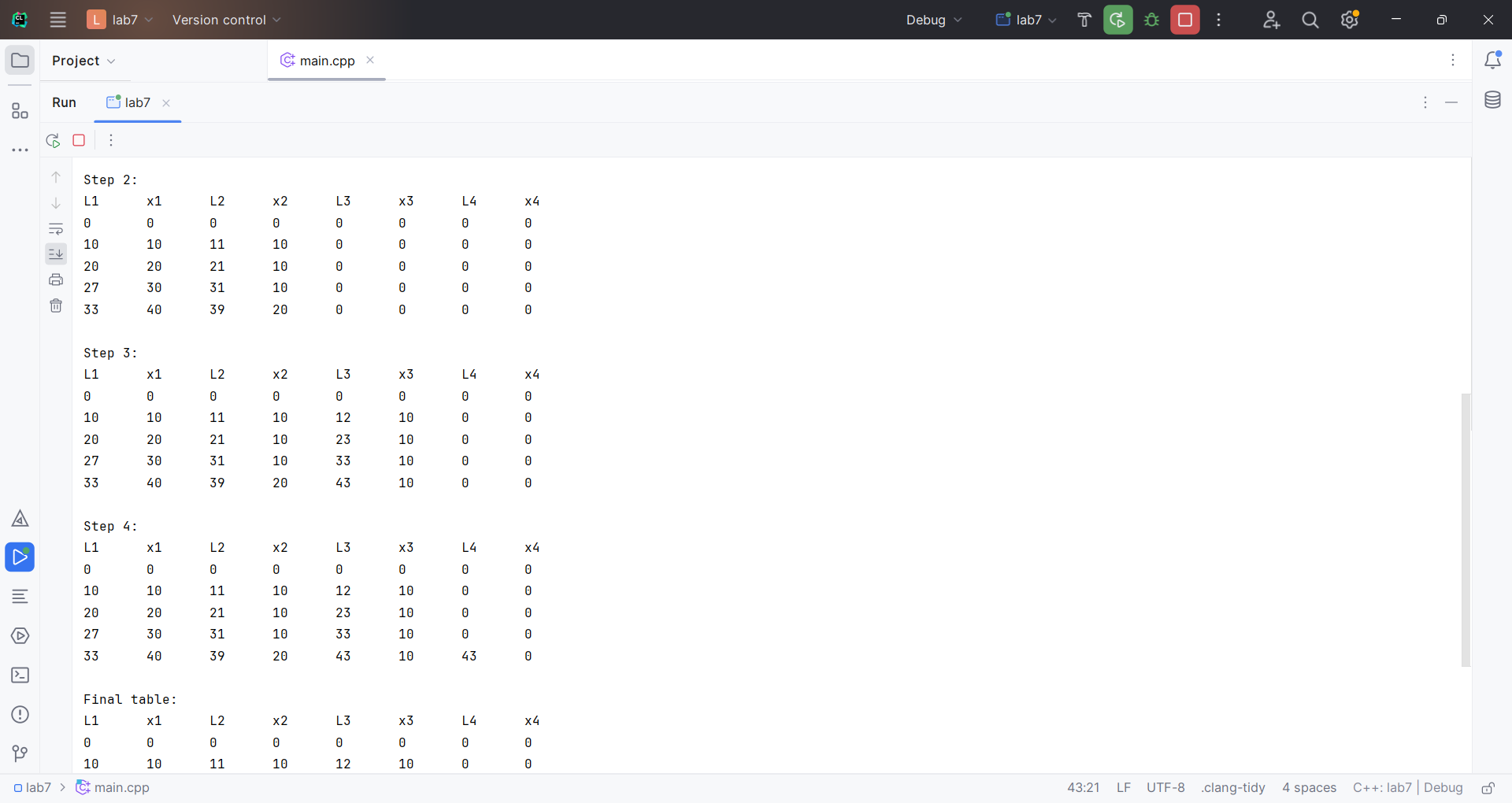
}

printf("The maximum profit will be - %d \* 10 ^ 3 uah.\n", ms[n - 1][6]);

}

**Результат виконання програми**





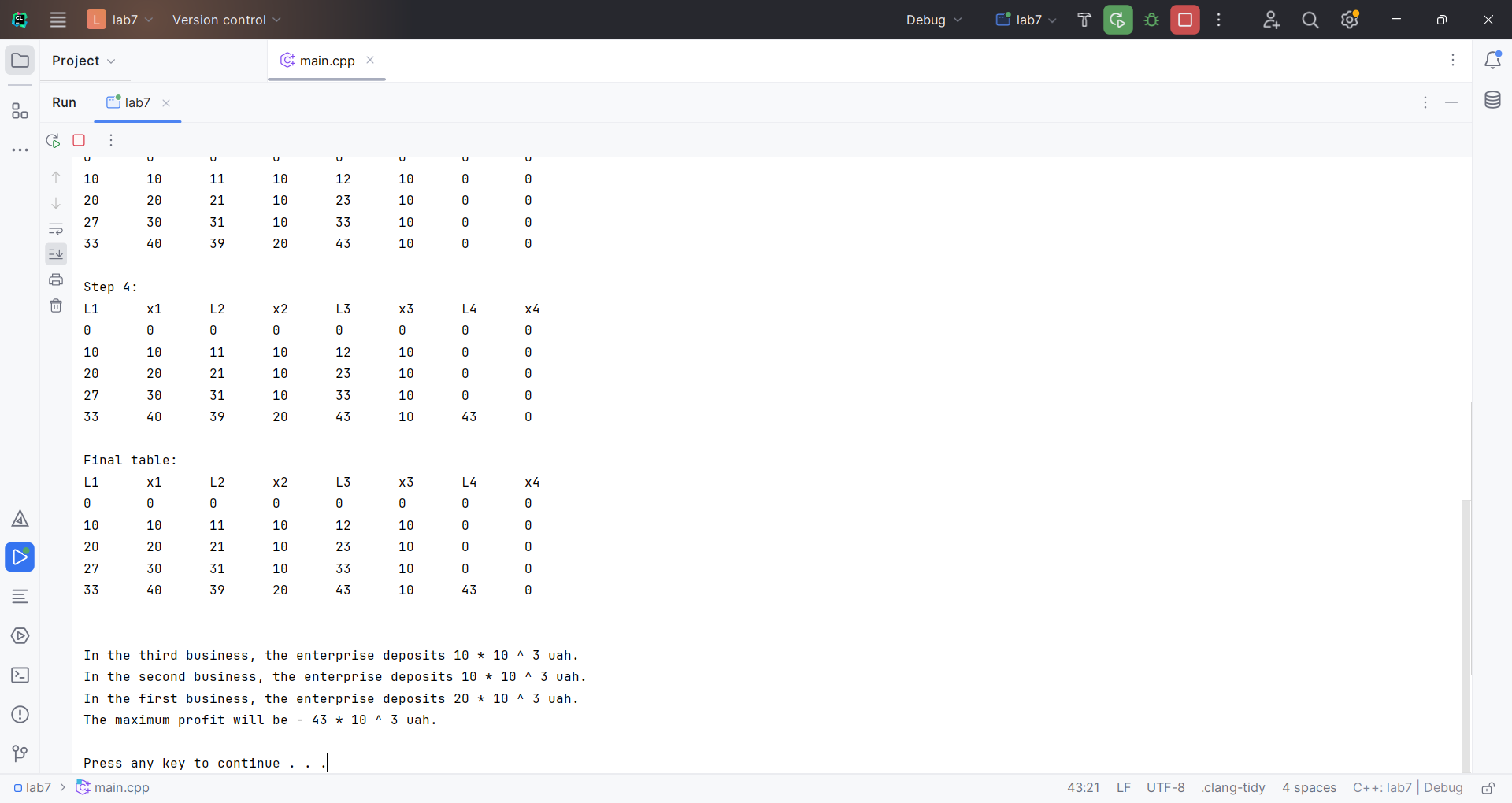


Рис. Результат виконання програми.

**Висновки**

Згідно результатів та порівняння отриманих під час виконання програми – алгоритм виконується правильно та без помилок.

Спершу було сформовано таблицю для виконання завдання.

Далі було знайдено аналітичний розв’язок, який в подальшому використовувався для перевірки результатів.

Далі було розроблено програму для реалізації цього завдання та протестовано доки не було досягнуто відповідних результатів. Програма було скомпільована та написана на мові програмування С++, також, було додано коментарі, які коротко описують усі функції та змінні (глобальні чи локальні), що використовуються.

Програма було виконано у середовищі CLion 2023.2.2 з використанням компілятора MinGW (g++) 6.3.0.