

Центральноукраїнський національний технічний університет
Кафедра програмування комп'ютерних систем і мереж
Дисципліна: Базові методології та технології програмування

Звіт

з лабораторних робіт №20

Теми: Освоєння технології та принципів програмування з використанням
функцій C/C++

Мета: Ознайомитись з базою створення та застосування функцій у мові C++ та
отримати практичні навички оголошення, визначення та виклика функцій.

Виконав: ст. гр. КМ-20

Калиновський В.М.

Перевірила: доцент кафедри ПКСМ

Рибакова Л.В.

Варіант 3

Дата виконання: 9 травня 2021

Дата оформлення: 10 травня 2021

Кропивницький 2020 р.

Завдання 1. Розробити схему алгоритма та написати програму, що включає оголошення визначення та демонстрацію роботи через виклик двох функцій. Варіант 3: Аргумент — змінна покажчик на масив символічних значень і кількість елементів масива; значення, що повертається — кількість латинських літер у масиві.

Завдання 2. Розробити схему алгоритма та написати програму, що включає оголошення визначення та демонстрацію роботи через виклик двох функцій. Варіант 3: Аргумент — координати кінців відрізка АВ; значення, що повертається — координати точки середини відрізка.

Завдання 3. Розробити програму з:

- Написати функцію, що виводить ціле число між 1 та 32767 й друкує це число як послідовність цифр, між якими стоїть два пробіли.

Робота. Всі три завдання зібрані в одному проекті. Кожне із завдань являє собою окрему функцію. Так як немає ніяких уточнень, буде вважатися, що для всіх завдань можна самостійно завчасно прописати всі вхідні дані для функцій. Для функції із завдання 1 розроблено наступну загальну структуру функції:

Крок 1: Створюється масив з елементами латинського алфавіту;

Крок 2: Кожен з елементів переданого масива перевіряється з кожним елементом латинського алфавіта;

Крок 3: Якщо є співпадіння — зарахувати його (так як у масиві елементів латинського алфавіта нема повторів, то співпадінь бути не може й кожна літера якщо і буде зарахована у разі співпадіння, то лише один раз);

Крок 4: Повернути кількість співпадінь.

Що очікується від функції? Вона приймає масив з символами й повертає кількість символів що є літерами латинського алфавіта.

Код функції:

```
const int latinaAlpha = 26;
int elLat(char mass[gLEN])
{
    char alf[latinaAlpha] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',
                              'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
    int elLat = 0;
    for (int i = 0; i < gLEN; i++)
        for (int j = 0; j < latinaAlpha; j++)
            if (mass[i] == alf[j])
                elLat++;
    return elLat;
}
```

Для функції із завдання 2 написана загальна структура функції так:

Крок 1: Прийняти заповнений масив координат кінців відрізка та пустий масив координат середини відрізка;

Крок 2: Порахувати першу координату середини відрізка;

Крок 3: Порахувати другу координату.

Що очікується від функції? Вона приймає масиви кінців відрізка та середини

відрізка й розраховує та заповнює координати середини відрізка.

Код функції:

```
const int coord = 2;
void coordC(int AB[coord][coord], int C[coord])
{
    C[0] = (AB[0][0] + AB[1][0]) / 2;
    C[1] = (AB[0][1] + AB[1][1]) / 2;
}
```

Для функції із завдання 3 написано таку загальну структуру функції:

Крок 1: Визначається, якого порядку число (1 цифра, 2, 3, 4 чи 5 цифр);

Крок 2: Створюється масив символів цифр числа;

Крок 3: Заповнити масив:

Крок 3.1: Знайти залишок від ділення на 10 й занести його у останній елемент масива;

Крок 3.2: Так як цифри при заносенні в даний масив символів позначають номер символа за таблицею ASCII, треба додавати до даної цифри 48 [див. *ілюстр1*] (цифра “нуль” в таблиці ASCII починається на сорок восьмому елементі, тому для збереження, наприклад, того ж нуля, необхідно занести 48 символ тощо);

Крок 3.3: Поділити дане число на десять;

Крок 3.4: Повторити дану операцію (3.1 — 3.3) стільки разів, скільки розрядів числа;

Крок 4: Вивести масив, вставивши між кожним елементом два пробіли.

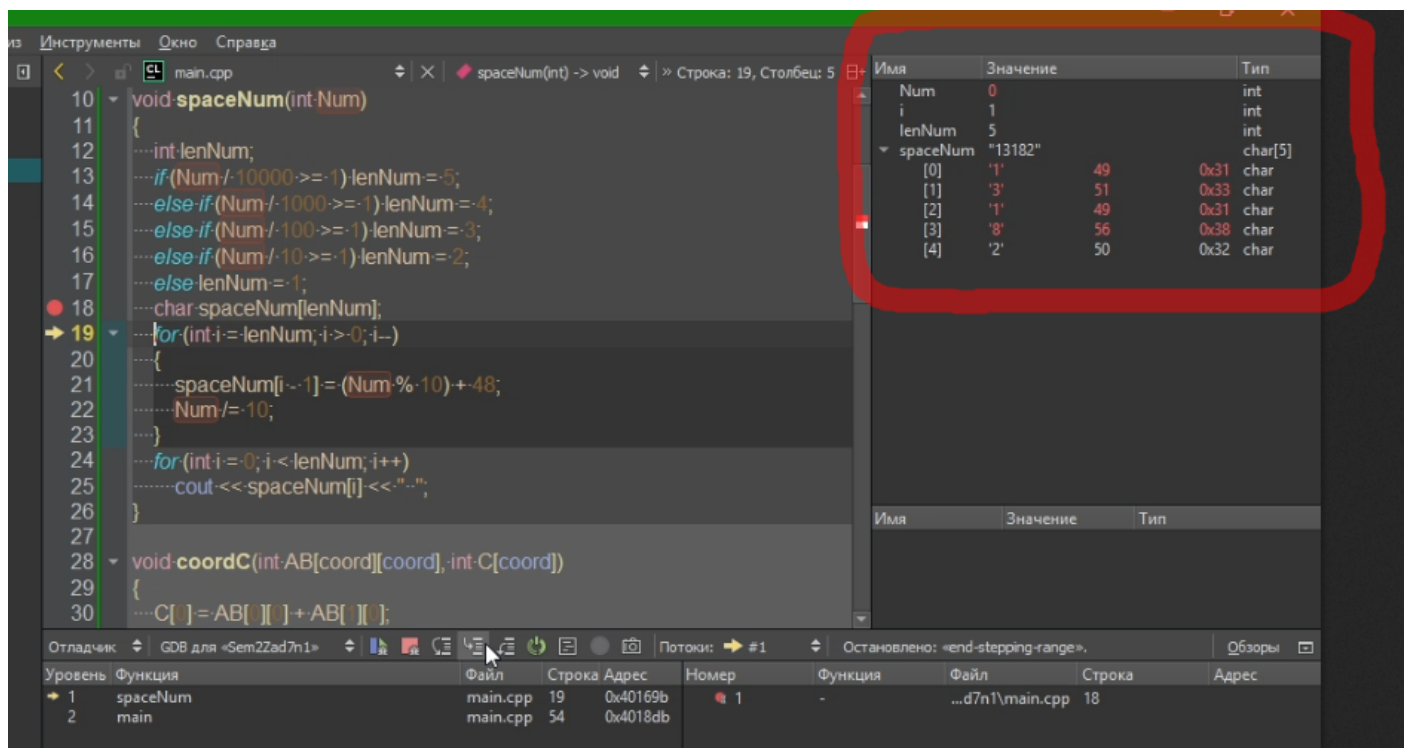
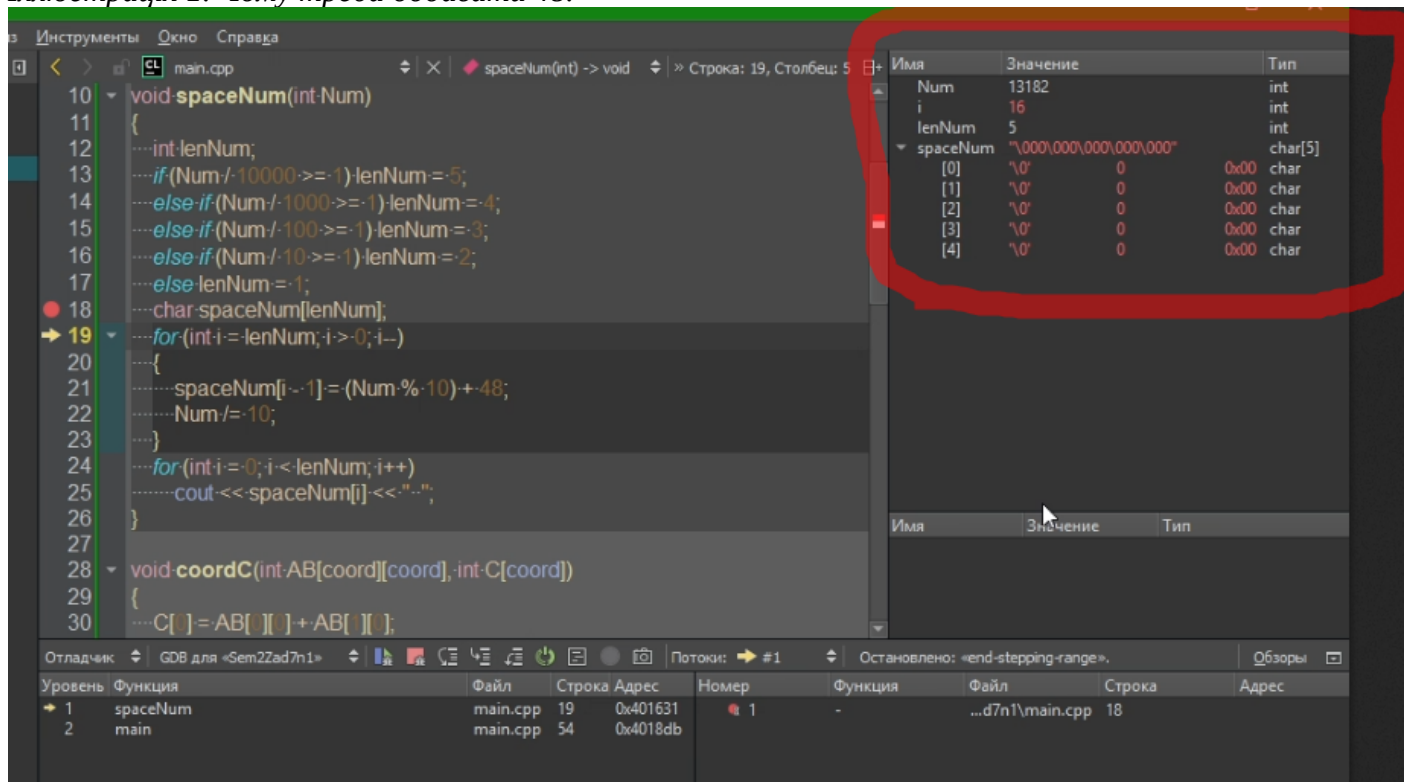
Що очікується від функції? Вона приймає число й виводить строку цифр, при цьому між кожною цифрою має бути по два пробіли.

Код функції:

```
void spaceNum(int Num)
{
    int lenNum;
    if (Num / 10000 >= 1) lenNum = 5;
    else if (Num / 1000 >= 1) lenNum = 4;
    else if (Num / 100 >= 1) lenNum = 3;
    else if (Num / 10 >= 1) lenNum = 2;
    else lenNum = 1;
    char spaceNum[lenNum];
    for (int i = lenNum; i > 0; i--)
    {
        spaceNum[i - 1] = (Num % 10) + 48;
        Num /= 10;
    }
    for (int i = 0; i < lenNum; i++)
        cout << spaceNum[i] << "  ";
}
```

Зауваження: я знаю, що можна було краще зробити функцію виявлення розрядності числа, але так як всі завдання не передбачає ускладнень кода, я вирішив написати це топорним методом не заморочуючись.

Ілюстрація 1. Чому треба додавати 48.



Загалом, *програма* має наступну структуру:

Крок 1: Створити масив символів, частина з яких має бути буквами латинського алфавіта (так як символи кирилиці у кодуванні windows-1251 займають 2 байти, я не можу заповнити масив символами кирилиці, тому частину масива заповню просто цифрами);

Крок 2: Створити масив координат кінців відрізка АВ (і заповнити) й масив координат середини відрізка С (не заповнювати);

Крок 3: Вивести результат першої функції (викликати її у cout);

Крок 4: Викликати функцію визначення координат середини відрізка;

Крок 5: Вивести координати середини відрізка;

Крок 6: Згенерувати випадкове число від 1 до 32767;

Крок 7: Викликати функцію, яка вставитиме по два пробіли між всіма цифрами згенерованого числа;

Крок 8: Завершення програми.

Код програми:

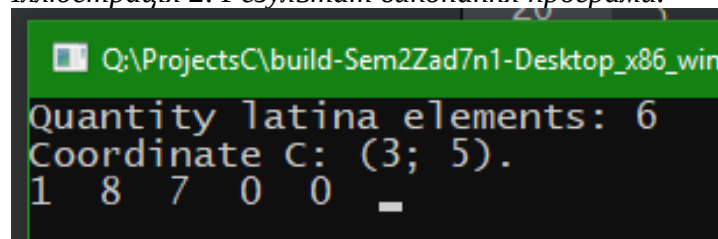
```
#include <random>
#include <time.h>
#include <iostream>
using namespace std;

const int gLEN = 10;
const int latinaAlpha = 26;
const int coord = 2;

int main()
{
    char mass[gLEN] = {'b', 'c', 'd', 'e', '4', '3', '7', 'y', '9', 'g'};
    int AB[coord][coord] = { {2, 1}, {4, 9} }, C[coord];
    cout << "Quantity latina elements: " << elLat(mass);
    coordC(AB, C);
    cout << "\nCoordinate C: (" << C[0] << "; " << C[1] << ").\n";
    srand(time(NULL));
    int randNum = 1 + rand() % 32766;
    spaceNum(randNum);
    cin.get();
    return 0;
}
```

Що очікується від програми? Програма має порахувати кількість латинських літер у заданому масиві, координати середини відрізка, знаючи координати кінців цього самого відрізка та вивести випадково згенероване число, вставивши у нього при виводі між кожною цифрою по два пробіли.

Ілюстрація 2. Результат виконання програми.



Результат: як ми бачимо з кода, ми завчасно задали масив символів і координати відрізків. Ми власноруч можемо перевірити, чи правильно працює наша програма. Ми рахуємо, що масиві символів латинських символів є 6, а координати середини (3; 5). Перевіряємо результати — співпадають. Третє завдання ми самостійно не перевіримо, але ми бачимо, що у виведеному числі між кожною цифрою стоїть по два пробіли. Отже можемо вважати, що програма написана правильно й виконує свої функції (гра слів), і *вважаємо завдання виконаним!* ✓

Далі наведено повний код і блок-схеми:

```
#include <random>
#include <time.h>
#include <iostream>
using namespace std;

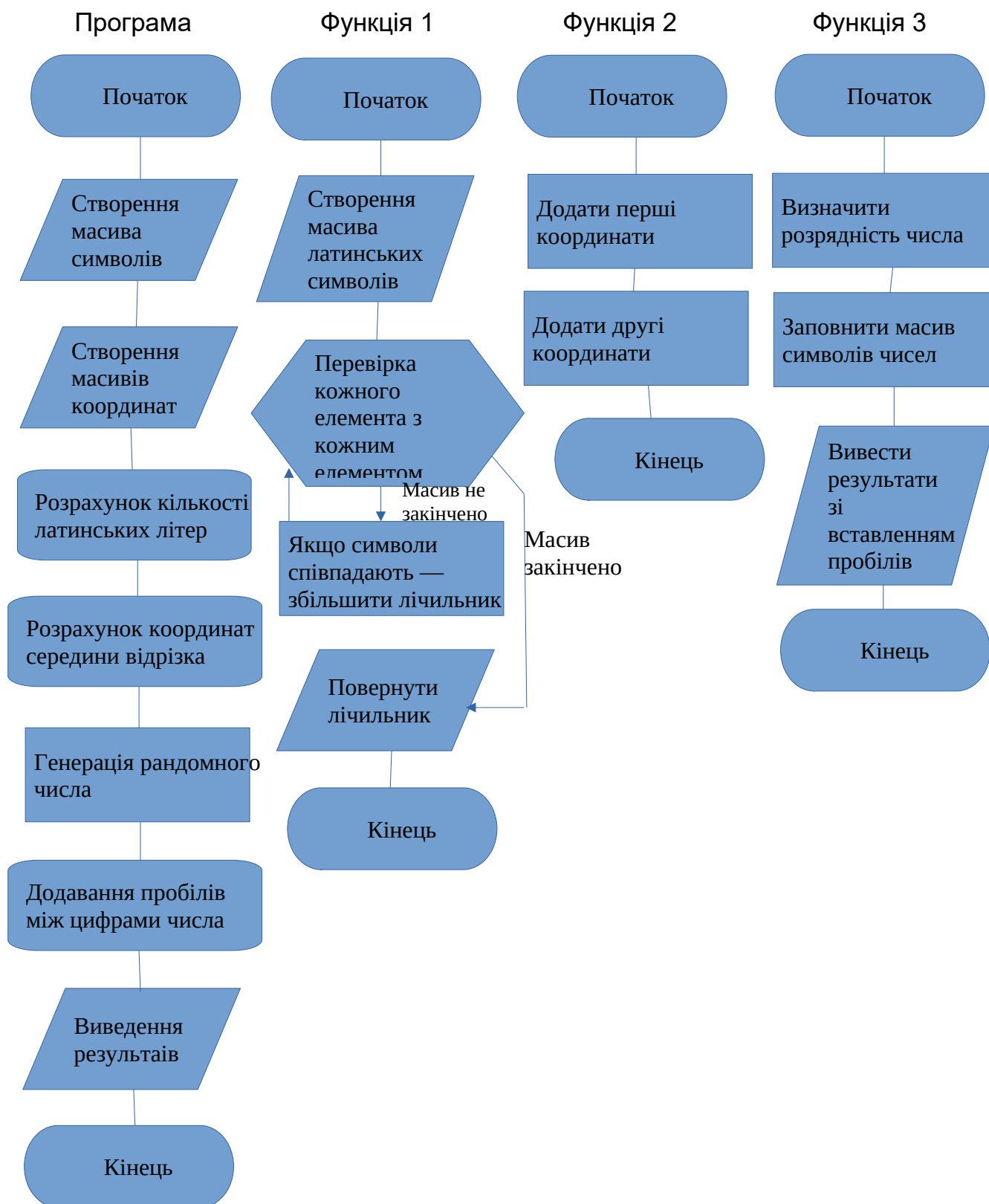
const int gLEN = 10;
const int latinaAlpha = 26;
const int coord = 2;

void spaceNum(int Num)
{
    int lenNum;
    if (Num / 10000 >= 1) lenNum = 5;
    else if (Num / 1000 >= 1) lenNum = 4;
    else if (Num / 100 >= 1) lenNum = 3;
    else if (Num / 10 >= 1) lenNum = 2;
    else lenNum = 1;
    char spaceNum[lenNum];
    for (int i = lenNum; i > 0; i--)
    {
        spaceNum[i - 1] = (Num % 10) + 48;
        Num /= 10;
    }
    for (int i = 0; i < lenNum; i++)
        cout << spaceNum[i] << " ";
}

void coordC(int AB[coord][coord], int C[coord])
{
    C[0] = (AB[0][0] + AB[1][0]) / 2;
    C[1] = (AB[0][1] + AB[1][1]) / 2;
}

int elLat(char mass[gLEN])
{
    char alf[latinaAlpha] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',
    'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
    int elLat = 0;
    for (int i = 0; i < gLEN; i++)
        for (int j = 0; j < latinaAlpha; j++)
            if (mass[i] == alf[j])
                elLat++;
    return elLat;
}

int main()
{
    char mass[gLEN] = {'b', 'c', 'd', 'e', '4', '3', '7', 'y', '9', 'g'};
    int AB[coord][coord] = { {2, 1}, {4, 9} }, C[coord];
    cout << "Quantity latina elements: " << elLat(mass);
    coordC(AB, C);
    cout << "\nCoordinate C: (" << C[0] << "; " << C[1] << ").\n";
    srand(time(NULL));
    int randNum = 1 + rand() % 32766;
    spaceNum(randNum);
    cin.get();
    return 0;
}
```



Висновок: всі завдання було виконано й усі функції виконують свій функціонал — що від них потребується, те вони й виконують. Ми ознайомились з базою створення та застосування функцій у мові C++ та отримали практичні навички оголошення, визначення та виклика функцій.