Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Катедра «Комп'ютерна інженерія та програмування»

ЗВІТ

про виконання лабораторної роботи №3

з навчальної дисципліни «Алгоритми та структури даних»

Варіант 9

Виконав студент:

Ульянов Кирило Юрійович
Група: КН-1023b

Перевірив:

старший викладач

Бульба С.С.

Харків-2024

1 Мета роботи

Отримати та закріпити знання про внутрішнє (комп'ютерне) подання числових типів даних у мовах програмування.

2 Хід роботи

1) Написати програму, яка виводить на екран внутрішнє (комп'ютерне) подання даних чотирьох типів. Типи даних обрати за табл. 3.1 згідно із своїм номером у журналі групи. Тип елементів масиву обрати за своїм розсудом.

№ З/П	integer	short int	long int	float	double	long double	char	[] *n*-вимірний
1	2	3	4	5	6	7	8	9
1	*			*			*	1
2	*			*			*	2
3	*			*			*	3
4		*			*		*	1
5		*			*		*	2
6		*			*		*	1

7			*			*	*	2
8			*			*	*	3
9			*			*	*	1

Реалізація програми:

```c
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#include "general_utils.h"

void printBinary(unsigned char byte) {
    for (int i = 7; i >= 0; i--) {
        int bit = (byte >> i) & 1;
        printf("%d", bit);
    }
    printf(" ");
}

void printInternalLongInt(long int val){
    printf("\033[35mMachine code of (long int = %ld)\033[0m: ", val);
    for (size_t i = 0; i < sizeof(long int); i++)
    {
        printBinary((unsigned char)(val >> (i * 8)));
    }
    printf("\n");
}

void printInternalLongDouble(long double val){
    printf("\033[35mMachine code of (long double = %Lf)\033[0m: ", val);
    unsigned char *longDoubleBytes = (unsigned char*)&val;
    for (int i = 0; i < sizeof(long double); i++) {
        printBinary(longDoubleBytes[i]);
    }
    printf("\n");
}

void printInternalChar(char val) {
    printf("\033[35mMachine code of (char = '%c')\033[0m: ", val);
    printBinary((unsigned char)val);
    printf("\n");
}

void printInternalInt(int val) {
    printf("\033[35mMachine code of (int = %d)\033[0m: ", val);
    for (size_t i = 0; i < sizeof(int); i++) {
        printBinary((unsigned char)(val >> (i * 8)));
    }
    printf("\n");
}

void printInternalIntArray(int* arr, size_t size) {
    printf("\033[34mMachine code of array with int values\033[0m:\n\n");
    for (size_t i = 0; i < size; i++) {
        printf("    %zu: ", i);
        printInternalInt(arr[i]);
    }
}

void task1() {
    long int longIntValue = 78L;
    long double longDoubleValue = 3.14159L;
```

```
    char charValue = 'A';

    int arr[5] = {1, 2, 3, 4, 5};

    highlightText("Machine code of values:", "blue");
    printf("\n");

    printInternalLongInt(longIntValue);
    printInternalLongDouble(longDoubleValue);
    printInternalChar(charValue);
    printf("\n");
    printInternalIntArray(arr, 5);
}
```

2) Для відображення всіх 8 бітів у байті було створено функцію **printBinary**, вона робить це шляхом бінарного здвигу вправо на кількість позицій на поточній ітерації та застосовує маску **”И”з одиницею** щоб отримати тільки останній біт та вивести його. Все це вона робить з типом **unsigned char**, бо він зручний для відображення одного байту.

```
void printBinary(unsigned char byte) {
    for (int i = 7; i >= 0; i--) {
        int bit = (byte >> i) & 1;
        printf("%d", bit);
    }
    printf(" ");
}
```

3) Вивід **long int** здійснюється за допомогою функції **printInternalLongInt**, вона прийме в себе значення та виведе його машинне представлення у такій кількості байт який займає тип данних **long int**:

```
void printInternalLongInt(long int val){
    printf("\033[35mMachine code of (long int = %ld)\033[0m: ", val);
    for (size_t i = 0; i < sizeof(long int); i++)
    {
        printBinary((unsigned char)(val >> (i * 8)));
    }
    printf("\n");
}
```

4) Вивід **long double** здійснюється за допомогою функції **printInternalLongDouble**, вона прийме в себе значення та виведе його машинне представлення у такій кількості байт який займає тип данних **long double**:

```
void printInternalLongDouble(long double val){
    printf("\033[35mMachine code of (long double = %Lf)\033[0m: ", val);
    unsigned char *longDoubleBytes = (unsigned char*)&val;
    for (int i = 0; i < sizeof(long double); i++) {
        printBinary(longDoubleBytes[i]);
    }
    printf("\n");
}
```

5) Вивід **char** здійснюється за допомогою функції **printInternalChar**, вона прийме в себе значення та виведе його машинне представлення у такій кількості байт який займає тип данних **char**:

```
void printInternalChar(char val) {
    printf("\033[35mMachine code of (char = '%c')\033[0m: ", val);
    printBinary((unsigned char)val);
    printf("\n");
}
```

6) Виведення масиву (в моєму випадку масиву цілих чисел) відбувається за допомогою почергового представленя у машиному вигляді кожного елементу масиву, для цього була додана функція **printInternalInt** для виводу машиного представлення цілих чисел:

```
void printInternalInt(int val) {
    printf("\033[35mMachine code of (int = %d)\033[0m: ", val);
    for (size_t i = 0; i < sizeof(int); i++) {
        printBinary((unsigned char)(val >> (i * 8)));
    }
    printf("\n");
}

void printInternalIntArray(int* arr, size_t size) {
    printf("\033[34mMachine code of array with int values\033[0m:\n\n");
    for (size_t i = 0; i < size; i++) {
        printf("    %zu: ", i);
        printInternalInt(arr[i]);
    }
}
```

7) Результати роботи програми:

```
whitegolyb@Kyrylo: /mnt/c/Users/kiril/Documents/Algos_Labs/lab3/build
  1 -- Завдання 1


Введіть номер опціїї або номер завдання: 1

=====================================================
Завдання 1
=====================================================
Machine code of values:

Machine code of (long int = 78): 01001110 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Machine code of (long double = 3.141590): 00011101 01110010 00110011 11011100 10000000 11001111 00001111 11001001 000000
00 01000000 10110101 01110001 11111111 01111111 00000000 00000000
Machine code of (char = 'A'): 01000001

Machine code of array with int values:

Елемент 0: Machine code of (int = 1): 00000001 00000000 00000000 00000000
Елемент 1: Machine code of (int = 2): 00000010 00000000 00000000 00000000
Елемент 2: Machine code of (int = 3): 00000011 00000000 00000000 00000000
Елемент 3: Machine code of (int = 4): 00000100 00000000 00000000 00000000
Елемент 4: Machine code of (int = 5): 00000101 00000000 00000000 00000000
=====================================================
```

Можна зазначити що від‘ємні числа показаються у додатковому коді, наприклад змінимо елементи масиву, це можна легко перевірити якщо власноруч порахувати:

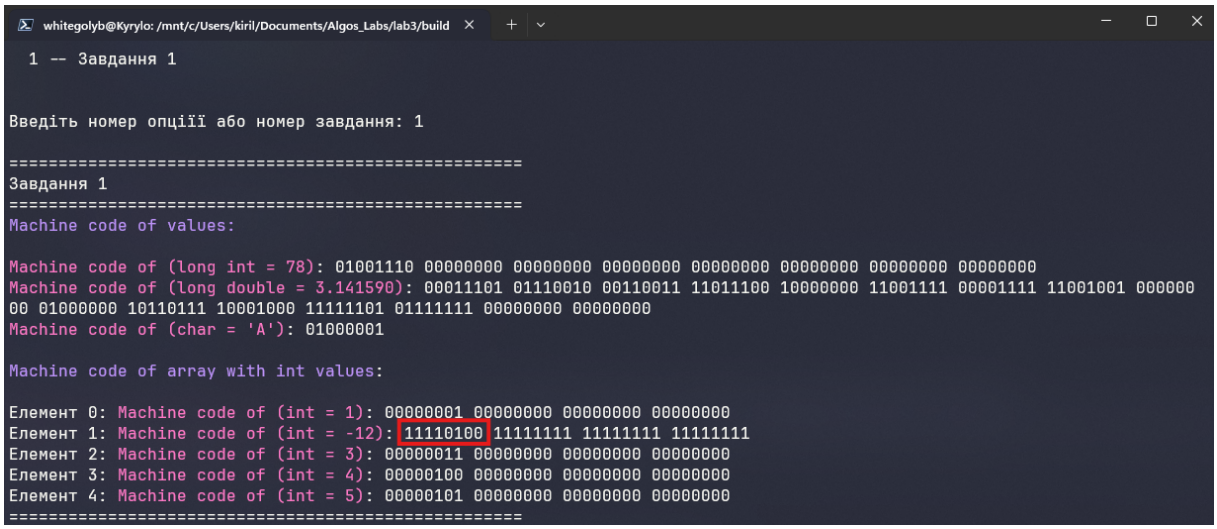

Рис. 1. Представленя у додатковому коді числа -12 у програмі

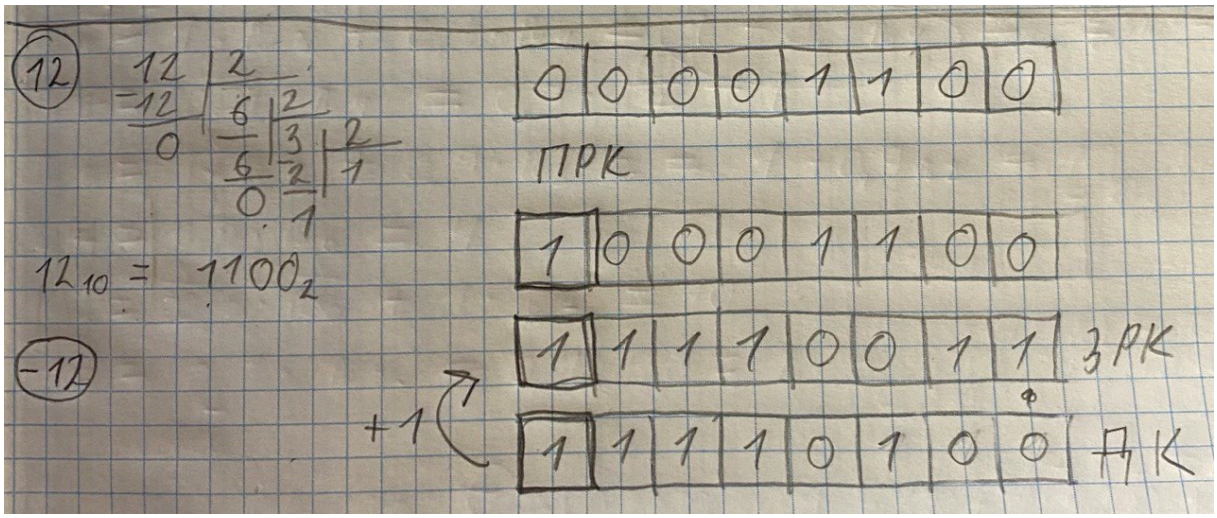

Рис. 2. Представленя у додатковому коді числа -12 власноруч

3 Висновки

В ході виконання лабоpіторної роботи я навчився прдеставляти різні типи даних у машинному коді, перводити їх у машинне представлення як програмно, так і власноруч. Можна побачити що всі числа відображаються у тій кількості байт, яка відведена під певний тип даних. Також можна побачити що знакові числа виводяться у додатковому коді, бо це є зручним для арифметичних операцій, оскільки операція віднімання зводится до операції додавання.