ANS w Elblągu

Instytut Informatyki Stosowanej im. Krzysztofa Brzeskiego

ASK SD – laboratorium

Studium stacjonarne, sem. 3, 2025/2026

Nazwisko i imię: **Jakub Chmielewski**

Nr albumu: **21432**

Sprawozdanie
Reprezentacja wartości liczbowych w pamięci komputera oraz konwersje typów danych w języku C

1. Repozytorium projektu

Kompletny kod źródłowy aplikacji wraz z plikami projektu został umieszczony w publicznym repozytorium na platformie GitHub. Repozytorium zawiera wszystkie pliki niezbędne do uruchomienia oraz analizy działania programu.

https://github.com/Chmielu6969/Projekt1-ASK/tree/main

2. Cel ćwiczenia

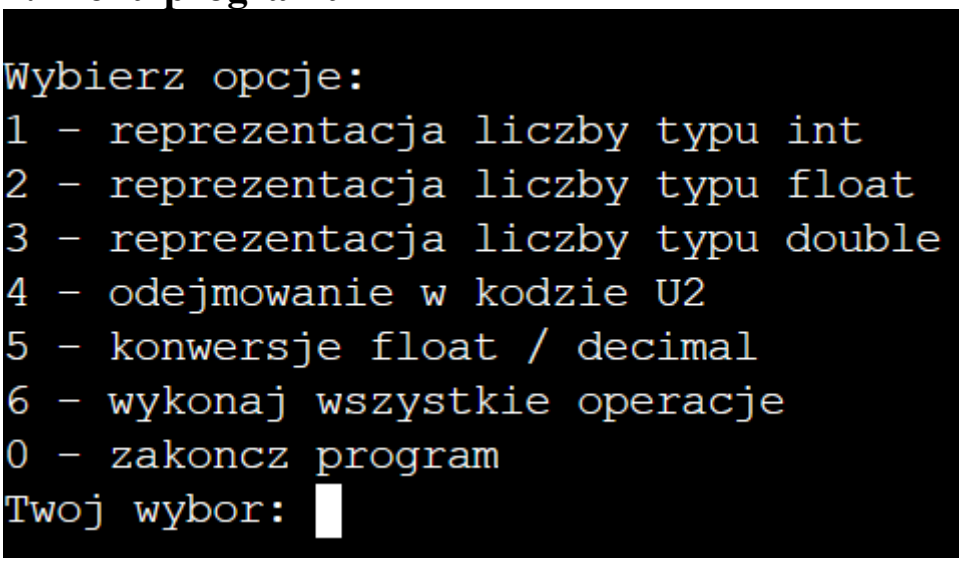
Celem realizowanego ćwiczenia było zapoznanie się ze sposobem zapisu danych liczbowych w pamięci komputera oraz ich interpretacją na poziomie sprzętowym. Program demonstracyjny miał na celu przedstawienie:

- zapisu bajtowego liczb typu int, float oraz double,
- binarnej reprezentacji liczb całkowitych i zmiennoprzecinkowych,
- mechanizmu odejmowania liczb całkowitych z wykorzystaniem kodu uzupełnień do dwóch (U2),
- konwersji pomiędzy formatami float i double zgodnie ze standardem IEEE 754.

3. Opis programu

Aplikacja została zaimplementowana w języku C i działa w trybie tekstowym. Obsługa programu odbywa się poprzez menu wyświetlane w konsoli, które umożliwia użytkownikowi wybór konkretnej operacji. Dane wejściowe wprowadzane są bezpośrednio z klawiatury. Program pozwala na uruchamianie pojedynczych funkcji lub wykonanie wszystkich dostępnych operacji jednocześnie dla jednej, wspólnej wartości wejściowej.

4. Menu programu



Po uruchomieniu aplikacji użytkownik otrzymuje menu zawierające następujące opcje:

1. prezentacja reprezentacji liczby typu int,
2. prezentacja reprezentacji liczby typu float,
3. prezentacja reprezentacji liczby typu double,
4. odejmowanie liczb z użyciem kodu U2,
5. konwersje pomiędzy typami float i decimal
6. wykonanie wszystkich dostępnych operacji,
0. zakończenie działania programu.

Po wybraniu odpowiedniej opcji użytkownik proszony jest o podanie wymaganych danych.

5. Reprezentacja danych liczbowych

5.1. Typ int

```
Twoj wybor: 1
Podaj liczbe typu int: 41

Podano liczbe typu int: 41
Postac bajtowa:
29 00 00 00
Postac binarna:
00000000 00000000 00000000 00101001
```

Dla wartości całkowitych program prezentuje:

- zapis bajtowy liczby w pamięci komputera z uwzględnieniem kolejności little endian,
- binarną postać liczby w kodzie uzupełnień do dwóch.

Pozwala to na obserwację sposobu przechowywania liczb całkowitych w pamięci operacyjnej.

5.2. Typ float

```
Twoj wybor: 2
Podaj liczbe typu float: 12.5

Podano liczbe typu float: 12.500000
Postac bajtowa:
00 00 48 41
Postac binarna:
01000001 01001000 00000000 00000000
```

W przypadku liczb typu float wyświetlana jest:

- reprezentacja bajtowa,
- binarna postać liczby zgodna ze standardem IEEE 754 dla 32 bitów.

Dzięki temu możliwa jest analiza bitu znaku, wykładnika oraz mantysy.

5.3. Typ double

```
Twoj wybor: 3
Podaj liczbe typu double: 12.6

Podano liczbe typu double: 12.600000
Postac bajtowa:
33 33 33 33 33 33 29 40
Postac binarna:
01000000 00101001 00110011 00110011 00110011 00110011 00110011 00110011
```

Dla typu double program przedstawia:

- zapis bajtowy,
- binarną reprezentację zgodną ze standardem IEEE 754 w formacie 64-bitowym.

Umożliwia to porównanie dokładności zapisu pomiędzy typami float i double.

7. Odejmowanie w kodzie U2

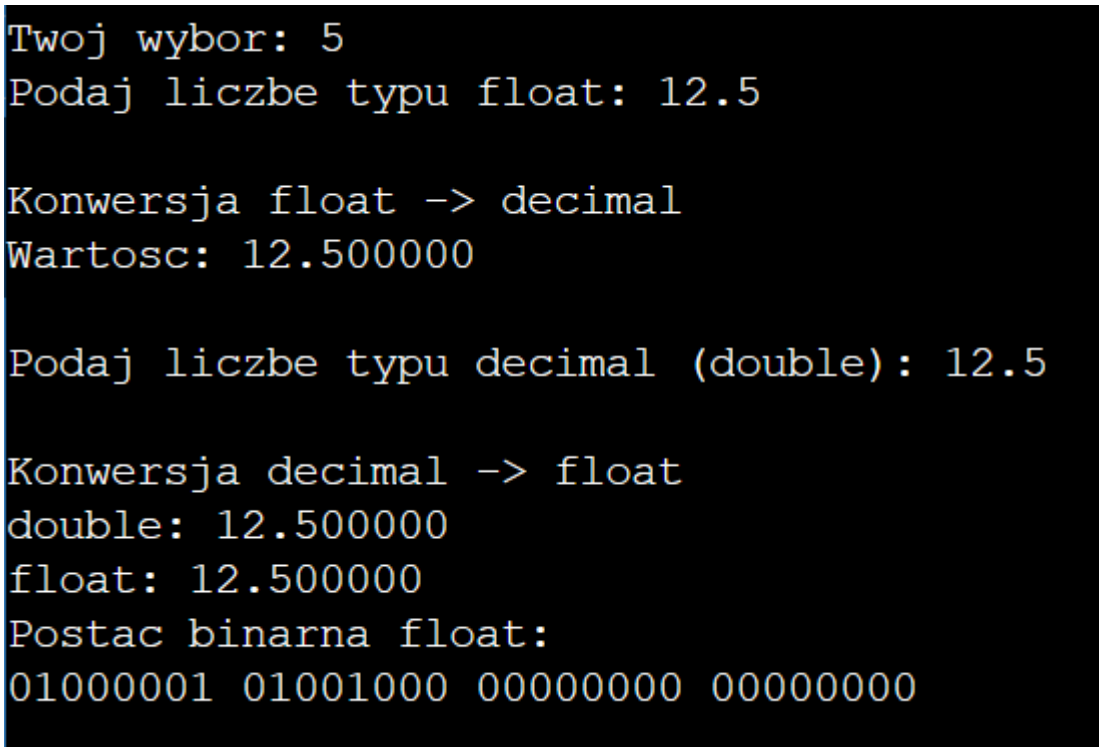
```
Twoj wybor: 4
Podaj dwie liczby typu int (a b): 50 10

Odejmowanie w kodzie U2: 50 - 10
Liczba a:
00000000 00000000 00000000 00110010
Liczba b:
00000000 00000000 00000000 00001010
Liczba -b (U2):
11111111 11111111 11111111 11110110
Wynik a + (-b):
00000000 00000000 00000000 00101000
Wynik dziesietny: 40
```

Program ilustruje sposób realizacji odejmowania liczb całkowitych przy użyciu kodu uzupełnień do dwóch. Proces ten polega na:

1. zanegowaniu drugiego argumentu,
2. dodaniu jedynki w celu uzyskania kodu U2,
3. dodaniu otrzymanej wartości do pierwszej liczby.

Każdy etap obliczeń prezentowany jest w postaci binarnej, co umożliwia szczegółowe prześledzenie działania algorytmu.

8. Konwersje float ↔ decimal



Aplikacja realizuje dwa niezależne mechanizmy konwersji:

- konwersję typu float na wartość dziesiętną w celu prezentacji jej dokładnej postaci,
- rzutowanie liczby typu double na float wraz z wyświetleniem jej binarnej reprezentacji.

Dla wybranych wartości, takich jak 10.1, możliwe jest zaobserwowanie ograniczonej precyzji zapisu liczb zmiennoprzecinkowych.

8. Podsumowanie

Zaimplementowany program umożliwia praktyczne poznanie sposobu reprezentacji danych liczbowych w pamięci komputera oraz mechanizmów ich przetwarzania. Realizacja ćwiczenia pozwoliła na lepsze zrozumienie działania kodu uzupełnień do dwóch oraz standardu IEEE 754, a także uwidoczniła ograniczenia związane z precyzją liczb zmiennoprzecinkowych.

Kod programu:

```
#include <stdio.h>

#include <stdint.h>

#include <string.h>



// Funkcja wypisuje bajtową reprezentację zmiennej


void print_bytes(void *ptr, size_t size) {

    unsigned char *b = (unsigned char *)ptr;

    for (size_t i = 0; i < size; i++) {

        printf("%02X ", b[i]);

    }

    printf("\n");

}


// Funkcja wypisuje 32-bitową postać binarną liczby


void print_binary32(uint32_t v) {

    for (int i = 31; i >= 0; i--) {

        printf("%d", (v >> i) & 1);

        if (i % 8 == 0) printf(" ");

    }

    printf("\n");
```

```
}

// Funkcja wypisuje 64-bitową postać binarną liczby

void print_binary64(uint64_t v) {
    for (int i = 63; i >= 0; i--) {
        printf("%d", (v >> i) & 1);
        if (i % 8 == 0) printf(" ");
    }
    printf("\n");
}

// Prezentacja liczby typu int

void show_int(int x) {
    printf("Podano liczbe typu int: %d\n", x);
    printf("Postac bajtowa:\n");
    print_bytes(&x, sizeof(int));
    printf("Postac binarna:\n");
    print_binary32((uint32_t)x);
}

// Prezentacja liczby typu float

void show_float(float x) {
```

```
    uint32_t bits;
    memcpy(&bits, &x, sizeof(float));

    printf("Podano liczbe typu float: %f\n", x);
    printf("Postac bajtowa:\n");
    print_bytes(&x, sizeof(float));
    printf("Postac binarna:\n");
    print_binary32(bits);
}

// Prezentacja liczby typu double

void show_double(double x) {
    uint64_t bits;
    memcpy(&bits, &x, sizeof(double));

    printf("Podano liczbe typu double: %lf\n", x);
    printf("Postac bajtowa:\n");
    print_bytes(&x, sizeof(double));
    printf("Postac binarna:\n");
    print_binary64(bits);
}

// Prezentacja odejmowania w kodzie U2
```

```
void show_u2(int a, int b) {
    int neg_b = ~b + 1;
    int result = a + neg_b;

    printf("Odejmowanie w kodzie U2: %d - %d\n", a, b);

    printf("Liczba a:\n");
    print_binary32((uint32_t)a);

    printf("Liczba b:\n");
    print_binary32((uint32_t)b);

    printf("Liczba -b (U2):\n");
    print_binary32((uint32_t)neg_b);

    printf("Wynik a + (-b):\n");
    print_binary32((uint32_t)result);

    printf("Wynik dziesietny: %d\n", result);
}

// Konwersja float -> decimal

void float_to_decimal(float x) {
    printf("Konwersja float -> decimal\n");
```

```
    printf("Wartosc: %f\n", x);
}


// Konwersja decimal -> float

void decimal_to_float(double x) {
    float f = (float)x;
    uint32_t bits;
    memcpy(&bits, &f, sizeof(float));

    printf("Konwersja decimal -> float\n");
    printf("double: %lf\n", x);
    printf("float: %f\n", f);
    printf("Postac binarna float:\n");
    print_binary32(bits);
}


// Menu

void show_menu(void) {
    printf("\n");
    printf("Wybierz opcje:\n");
    printf("1 - reprezentacja liczby typu int\n");
```

```
    printf("2 - reprezentacja liczby typu float\n");
    printf("3 - reprezentacja liczby typu double\n");
    printf("4 - odejmowanie w kodzie U2\n");
    printf("5 - konwersje float / decimal\n");
    printf("6 - wykonaj wszystkie operacje\n");
    printf("0 - zakoncz program\n");
}

// Main

int main() {
    int choice;

    do {
        show_menu();
        printf("Twoj wybor: ");
        scanf("%d", &choice);

        if (choice == 1) {
            int x;
            printf("Podaj liczbe typu int: ");
            scanf("%d", &x);
            printf("\n");
            show_int(x);
        }
```

```
else if (choice == 2) {
    float x;
    printf("Podaj liczbe typu float: ");
    scanf("%f", &x);
    printf("\n");
    show_float(x);
}
else if (choice == 3) {
    double x;
    printf("Podaj liczbe typu double: ");
    scanf("%lf", &x);
    printf("\n");
    show_double(x);
}
else if (choice == 4) {
    int a, b;
    printf("Podaj dwie liczby typu int (a b): ");
    scanf("%d %d", &a, &b);
    printf("\n");
    show_u2(a, b);
}
else if (choice == 5) {
    float f;
    double d;
    printf("Podaj liczbe typu float: ");
```

```
    scanf("%f", &f);
    printf("\n");
    float_to_decimal(f);
    printf("\n");
    printf("Podaj liczbe typu decimal (double): ");
    scanf("%lf", &d);
    printf("\n");
    decimal_to_float(d);
}
else if (choice == 6) {
    double x;
    printf("Podaj jedna liczbe (zostanie uzyta we wszystkich operacjach): ");
    scanf("%lf", &x);

    printf("\n");
    show_int((int)x);

    printf("\n");
    show_float((float)x);

    printf("\n");
    show_double(x);

    printf("\n");
    show_u2((int)x, 1);
```

```
            printf("\n");
            float_to_decimal((float)x);

            printf("\n");
            decimal_to_float(x);
        }

    } while (choice != 0);

    return 0;
}
```