

ANS w Elblągu

**Instytut Informatyki Stosowanej im. Krzysztofa
Brzeskiego**

**Architektura systemów komputerowych SD
– laboratorium**

Studium stacjonarne, sem. 3, 2025/2026

Nazwisko i imię: Jakub Chmielewski

Nr albumu: 21432

OPIS DZIAŁANIA PROGRAMU:

Projekt przedstawia sposób przechowywania danych w pamięci komputera oraz operacje wykonywane na liczbach na poziomie binarnym. Program został napisany w języku C i pokazuje rzeczywistą reprezentację liczb całkowitych oraz zmiennoprzecinkowych w pamięci RAM.

Drukuje postać bajtową danych INT, FLOAT, DOUBLE

Drukuje postać binarną dla liczb jak w p 1

Prezentuje zasadę wykonywania odejmowania w U2

Prezentuje konwersję FLOAT -> DEC i DEC -> FLOAT

LINK DO GITHUB'A:

<https://github.com/Chmielu6969/Projekt1-ASK>

KOD:

```
#include <stdio.h>

#include <stdint.h>

/* ===== Funkcja drukująca bajty ===== */

void print_bytes(const void *ptr, size_t size)
{
    const unsigned char *bytes = (const unsigned char *)ptr;

    for (size_t i = 0; i < size; i++)
    {
        printf("%02X ", bytes[i]);
    }
}
```

```

    }

    printf("\n");
}

/* ===== Funkcja drukująca bity ===== */
void print_bits(const void *ptr, size_t size)
{
    const unsigned char *bytes = (const unsigned char *)ptr;

    for (int i = size - 1; i >= 0; i--)
    {
        for (int bit = 7; bit >= 0; bit--)
        {
            printf("%d", (bytes[i] >> bit) & 1);
        }

        printf(" ");
    }

    printf("\n");
}

```

```

/* ===== Odejmowanie w U2 ===== */
void subtraction_U2(int8_t a, int8_t b)
{

```

```
int8_t minus_b = ~b + 1;
```

```
int8_t result = a + minus_b;
```

```
printf("\nODEJMOWANIE W U2\n");
```

```
printf("A = %d\n", a);
```

```
printf("B = %d\n", b);
```

```
printf("A bin:  ");
```

```
print_bits(&a, sizeof(a));
```

```
printf("B bin:  ");
```

```
print_bits(&b, sizeof(b));
```

```
printf("-B (U2):  ");
```

```
print_bits(&minus_b, sizeof(minus_b));
```

```
printf("A + (-B):  ");
```

```
print_bits(&result, sizeof(result));
```

```
printf("Wynik DEC: %d\n", result);
```

```
}
```

```
/* ===== FLOAT ↔ DEC ===== */
```

```

void float_conversion(float f)
{
    uint32_t raw;

    raw = *(uint32_t *)&f;

    printf("\nFLOAT -> DEC\n");

    printf("FLOAT: %f\n", f);

    printf("BITY IEEE754: ");

    print_bits(&raw, sizeof(raw));

    printf("\nDEC -> FLOAT\n");

    uint32_t dec = raw;

    float back = *(float *)&dec;

    printf("DEC (surowe bity): %u\n", dec);

    printf("FLOAT: %f\n", back);
}

int main(void)
{

    int i = -42;

    float f = 3.14159f;

```

```
double d = 3.14159;
```

```
printf("POSTAĆ BAJTOWA:\n");
```

```
printf("INT:\n");
```

```
print_bytes(&i, sizeof(i));
```

```
printf("FLOAT:\n");
```

```
print_bytes(&f, sizeof(f));
```

```
printf("DOUBLE:\n");
```

```
print_bytes(&d, sizeof(d));
```

```
printf("\nPOSTAĆ BINARNA:\n");
```

```
printf("INT:\n");
```

```
print_bits(&i, sizeof(i));
```

```
printf("FLOAT:\n");
```

```
print_bits(&f, sizeof(f));
```

```
printf("DOUBLE:\n");
```

```
print_bits(&d, sizeof(d));
```

```
subtraction_U2(25, 13);
```

```
float_conversion(f);
```

```
return 0;
```

```
}
```