

Wstęp Do Informatyki 03. Podstawowe typy danych

Rafał Grot

October 21, 2022

Contents

1	Operacje na liczbach całkowitych	1
1.1	Arytmetyczne	1
1.1.1	+	2
1.1.2	-	2
1.2	Logiczne	2
1.2.1	AND	2
1.2.2	OR	3
1.2.3	NOT	3
1.3	Bitowe	3
1.3.1	AND	3
1.3.2	\ll	4
1.3.3	\gg	4
1.3.4	Odczyt bitów (badanie bitów)	5
1.3.5	Ustawianie bitów	5
1.3.6	Zerowanie bitów	5
1.3.7	Negacja bitów	5

- Liczby całkowite
- liczby rzeczywiste

1 Operacje na liczbach całkowitych

1.1 Arytmetyczne

Wydaje mi się że to nie powinno być POW, bo POW to skrót od power, czyli potęgi po angielsku.

+	-	/	*	%
ADD	SUB	DIV	POW	MOD

1.1.1 +

$$\begin{array}{r}
 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\
 \\
 \begin{array}{r}
 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1_{\text{NKB}(8)} = 217_{\text{DEC}} \\
 + \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1_{\text{NKB}(8)} = 47_{\text{DEC}} \\
 \hline
 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0_{\text{NKB}(8)} = 264_{\text{DEC}} - 256_{\text{DEC}} = 8_{\text{DEC}}
 \end{array}
 \end{array}$$

1.1.2 -

$$\begin{array}{r}
 \begin{array}{r}
 1 \\
 0 \quad 2 \quad 0 \quad 2 \quad 2
 \end{array} \\
 \\
 \begin{array}{r}
 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1_{\text{NKB}(8)} = 217_{\text{DEC}} \\
 - \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1_{\text{NKB}(8)} = 47_{\text{DEC}} \\
 \hline
 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0_{\text{NKB}(8)} = 170_{\text{DEC}}
 \end{array}
 \end{array}$$

1.2 Logiczne

&&		!
AND	OR	NOT
Koniunkcja	Alternatywa	

1.2.1 AND

1. A=1, B=2

$A \&\& B = \text{TRUE}$

Bo

$A = 1 \rightarrow A = \text{TRUE}$

$B = 2 \rightarrow B = \text{TRUE}$

$\text{TRUE} \&\& \text{TRUE} = \text{TRUE}$

```
int a = 1;
int b = 2;
if (a && b) {}
```

2. $A = 0, B = 0$

$A \&\& B = \text{FALSE}$

$\text{FALSE} \&\& \text{FALSE} = \text{FALSE}$

1.2.2 OR

$A = 1, B = 2$

$A \parallel B = \text{TRUE}$

1.2.3 NOT

$A = 0$

$\neg A = \text{TRUE}$

$A = \text{FALSE}$

```
int a = false;
```

```
int b = true;
```

$a=0$

$b \neq 0$

1.3 Bitowe

$\&$	$ $	\sim	\ll	\gg	\wedge
AND	OR	NOT	SHL	SHR	XOR

1.3.1 AND

$A = 1$

$B = 2$

$A \& B = 0$

$$\begin{array}{rcccc} A = & 0 & 0 & 0 & 1 \\ B = & 0 & 0 & 1 & 0 \\ \hline A \& B = & 0 & 0 & 0 & 0 \end{array}$$

Częsty błąd:

```
int a=1;
```

```
int b=2;
```

```
if (a && b) {} // dobrze
```

```
if (a & b) {} // źle
```

1.3.2 \ll

$$A = 1$$

$$B = 2$$

$$A \ll B = 4$$

$$A = a_{N-1}a_{N-2} \dots a_1a_0$$

$$A \ll B = \underbrace{a_{N-1}a_{N-2} \dots a_1a_0}_{N-B \text{ bitów}} \underbrace{0 \dots 0}_{B \text{ bitów}}$$

1.3.3 \gg

$$A \gg B = ?$$

$$A = a_{N-1}a_{N-2} \dots a_1a_0 \underset{\text{U}_2}{\text{NKB}}$$

1. NKB

$$A_{\text{NKB}} \gg B = \underbrace{0 \dots 0}_{B \text{ bitów}} \underbrace{a_{N-1}a_{N-2} \dots a_{N+1}a_N}_{N-B \text{ bitów}} \text{NKB}$$

1. U₂

$$A_{\text{U}_2} \gg B = \underbrace{0 \dots 0}_{B \text{ bitów}} \underbrace{a_{N-1}a_{N-2} \dots a_{N+1}a_N}_{N-B \text{ bitów}} \text{U}_2$$

```
char a = -2;
char b = 2;
char c = a >> b;
```

$$\begin{array}{cccccccc} a = & 1 & 1 & 1 & 1 & 1 & 1 & 0_{\text{U}_2} & 1 \\ c = & 1 & 1 & 1 & 1 & 1 & 1 & 1_{\text{U}_2} & 1 & = -1_{\text{DEC}} \end{array}$$

```
unsigned char a = -2;      // a=254(DEC)
unsigned char b = 2;
unsigned char c = a >> b; // 63(DEC)
char d = c;                // 63(DEC)
```

$$\begin{array}{cccccccc} a = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0_{\text{NKB}} \\ c = & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & = 63 \end{array}$$

1.3.4 Odczyt bitów (badanie bitów)

char A = 10;

$A = 00001010_{U_2}$

$k = 3$ bit

$B = 00000100_{U_2}$

$A \& B = 00000000_{U_2} = 0_{DEC} \rightarrow \text{FAŁSZ}$

$A \& 4 = 00000000_{U_2} = 0_{DEC} \rightarrow \text{FAŁSZ}$

1.3.5 Ustawianie bitów

$k = 3$

$$\begin{array}{rcl} A = & 00001010_{U_2} & = 10_{DEC} \\ B = & 00000100_{U_2} & = -3_{DEC} \\ \hline A|B = & 00001110_{U_2} & = 14_{DEC} \end{array}$$

1.3.6 Zerowanie bitów

$k = 3$

$$\begin{array}{rcl} A = & 00001010_{U_2} & = 10_{DEC} \\ B = & 11111101_{U_2} & = -3_{DEC} \\ \hline A\&B = & 00001000_{U_2} & = 8_{DEC} \end{array}$$

1.3.7 Negacja bitów

$k = 3$

$$\begin{array}{rcl} A = & 00001010_{U_2} & = 10_{DEC} \\ B = & 00000100_{U_2} & = 4_{DEC} \\ \hline A\^B = & 00001110_{U_2} & \end{array}$$

int A = 256;

		B	G	R	
bity →					
	32	24	16	8	0

```
int R = A & 255;          // A & 0xFF
int G = (A >> 8) & 255;
int B = (A >> 16) & 255;
```

```
int R = 5;
int G = 10;
int B = 20;
int A = (A>>16) | (G<<8) | R;
```

```
int a = 2;
int b = a * 2;
int c = a << 1 ; // 2^k
```

```
if(a % 2 != 0) {} // Z automatu nizdany egzamin w tym semestrze
if(a & 1) {}
```

```
if(a > 0);
```