

Programowanie w języku C (EAR1S02005)

Politechnika Białostocka - Wydział Elektryczny
Automatyka i Robotyka, semestr II, studia stacjonarne I stopnia
Rok akademicki 2020/2021

Zajęcia nr 5 (09.03.2021)

dr inż. Jarosław Forenc

Język C - tablica elementów

- **Tablica** - ciągły obszar pamięci, w którym umieszczone są elementy tego samego typu

wektor

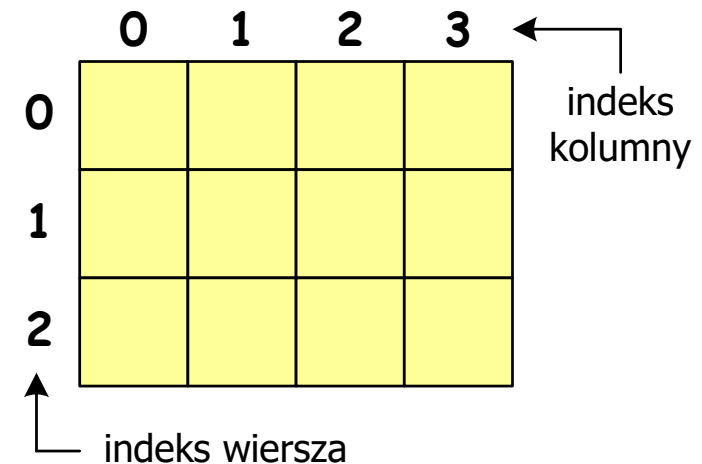
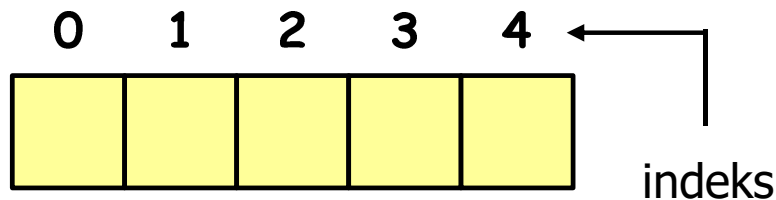
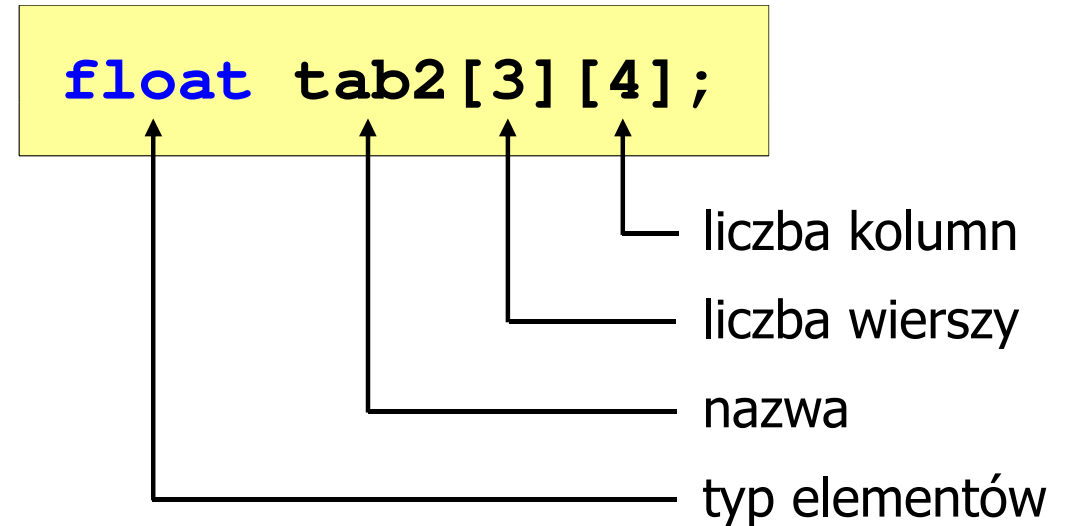
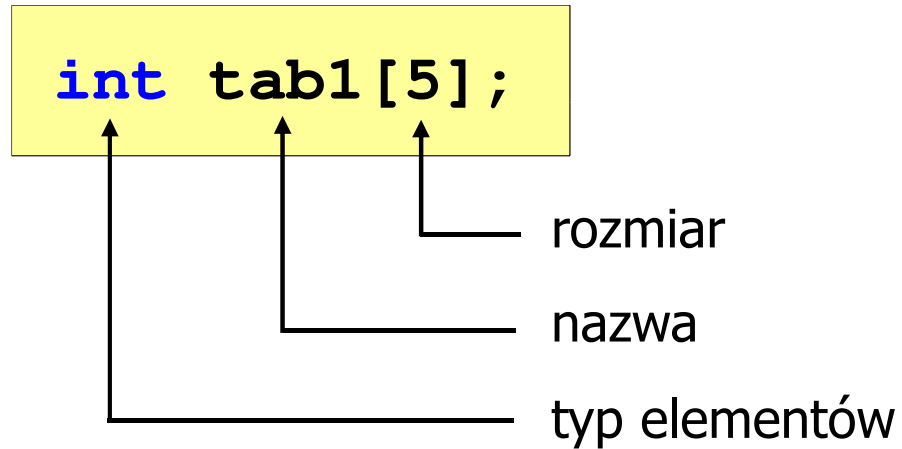
5	3	-2	1	-4
---	---	----	---	----

macierz

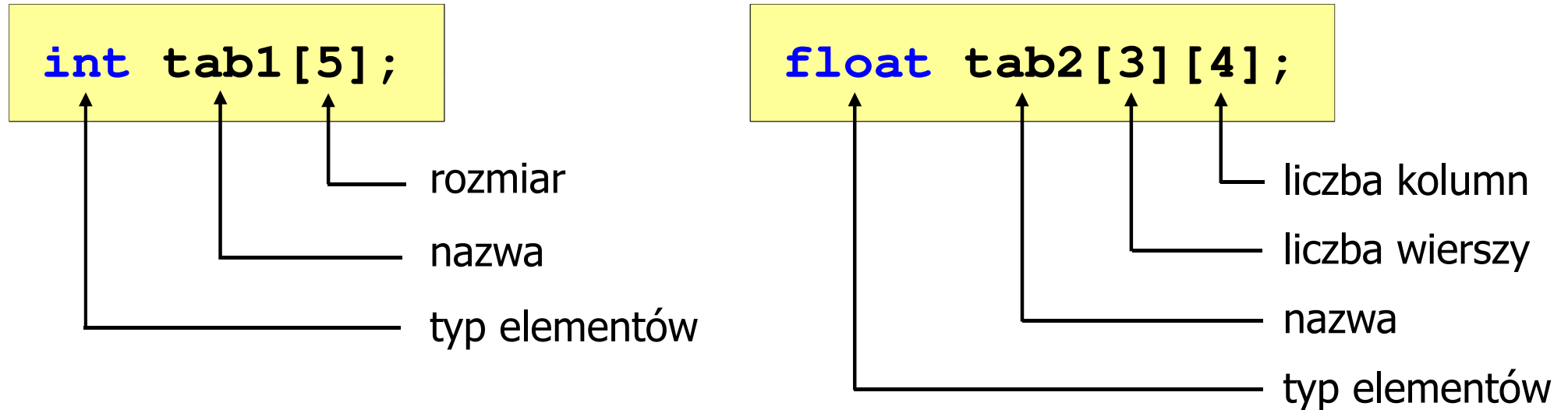
a	c	d	m
p	d	q	l
a	t	x	v

1.2	2.5	2.0	10.0
-0.1	4.3	6.2	-5.1
0.0	12.2	4.1	-2.2

Język C - deklaracja tablicy



Język C - deklaracja tablicy



- **Rozmiar** tablicy to wartość:
 - całkowita, dodatnia
 - znana na etapie kompilacji programu
(stała liczbowa: `5` `#define N 5` `const int n = 5;`)

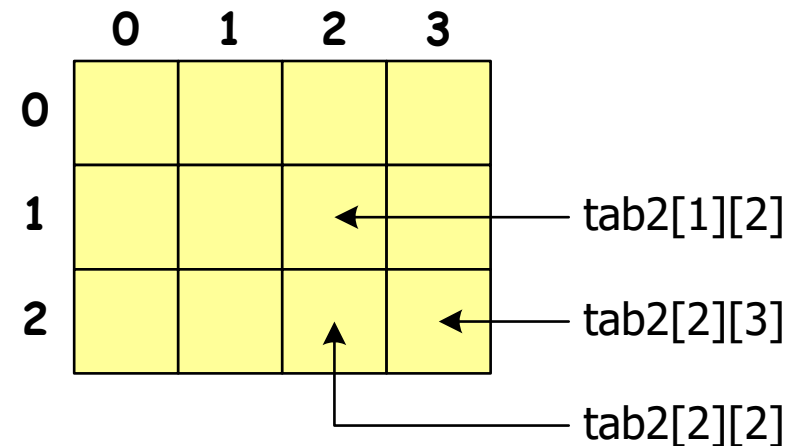
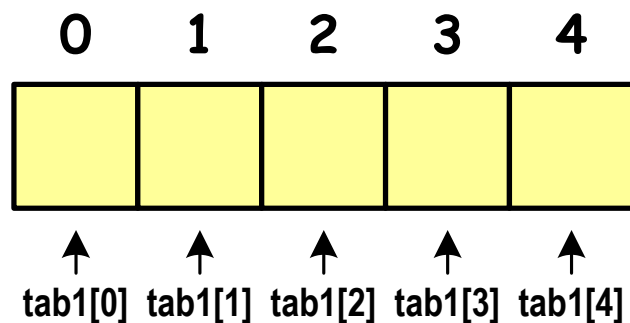
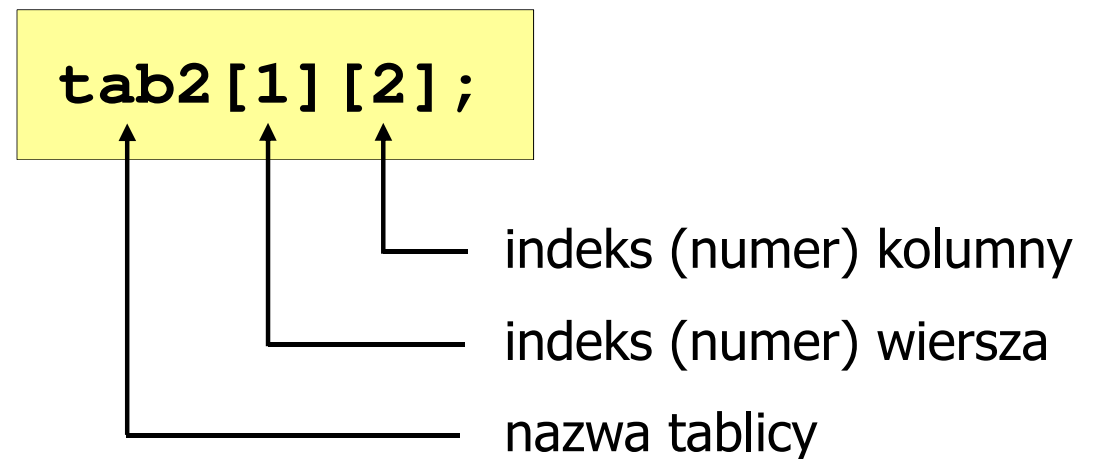
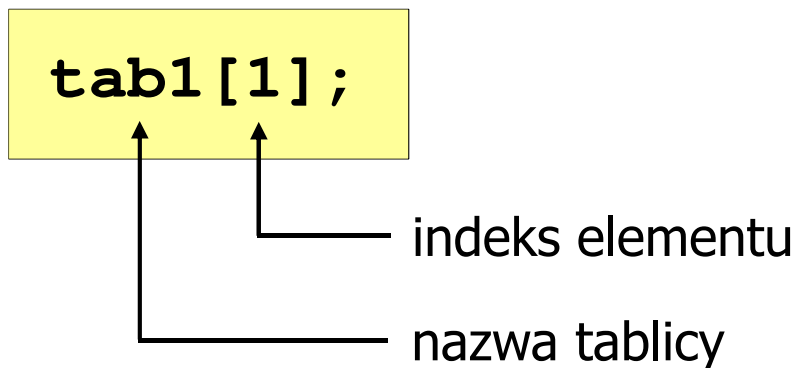
```
int tab1[5];
```

```
int tab1[N];
```

```
int tab1[n];
```

Język C - odwołania do elementów tablicy

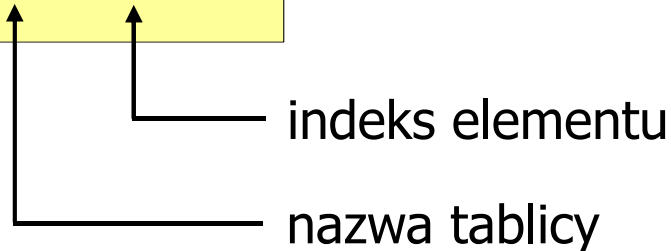
[] - dwuargumentowy operator indeksowania



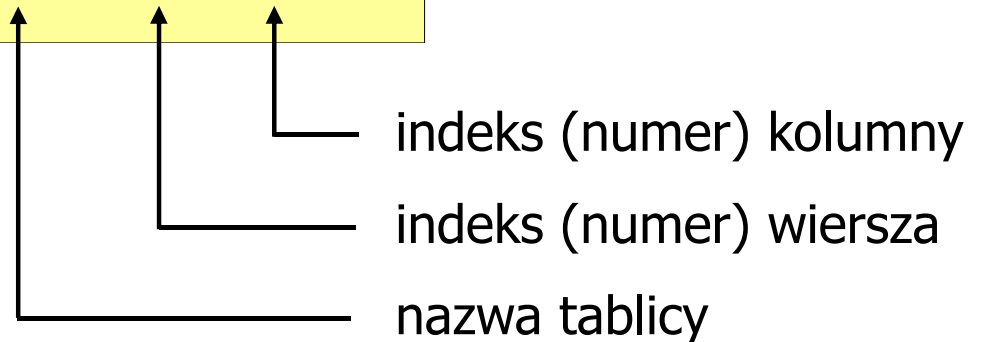
Język C - odwołania do elementów tablicy

[] - dwuargumentowy operator indeksowania

```
tab1[1];
```



```
tab2[1][2];
```



- Indeks:
 - stała liczbowa, np. **0**, **1**, **10**
 - nazwa zmiennej, np. **i**, **idx**
 - wyrażenie, np. **i*j+5**
- Brak sprawdzania poprawności indeksów!

Język C - odwołania do elementów tablicy

```
int tab1[4];
```



0	1	2	3
?	?	?	?

```
tab1[0] = 3;  
tab1[1] = -5;  
tab1[2] = 1;  
tab1[3] = 2;
```



0	1	2	3
3	-5	1	2

- Każdy element tablicy traktowany jest jak zmienna określonego typu

```
printf("%d", tab1[0]);
```

```
printf("%f", tab2[2][2]);
```

```
scanf("%d", &tab1[1]);
```

```
scanf("%f", &tab2[1][2]);
```

Język C - inicjalizacja elementów tablicy

```
int tab1[5] = {1, 2, 3, 4, 5};
```

0	1	2	3	4
1	2	3	4	5

```
int tab1[5] = {1, 2, 3};
```

0	1	2	3	4
1	2	3	0	0

```
int tab1[5] = {1, 2, 3, 4, 5, 6};
```

- błąd kompilacji

```
int tab1[5] = {0};
```

0	1	2	3	4
0	0	0	0	0

Język C - inicjalizacja elementów tablicy

```
int tab2[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

	0	1	2
0	1	2	3
1	4	5	6

```
int tab2[2][3] = {1, 2, 3, 4, 5, 6};
```

	0	1	2
0	1	2	3
1	4	0	0

```
int tab2[2][3] = {1, 2, 3, 4};
```

	0	1	2
0	1	0	0
1	4	5	0

```
int tab2[2][3] = {{1}, {4, 5}};
```

Język C - generowanie pseudolosowe liczb

- `rand()` - zwraca liczbę pseudolosową - zakres: `0 ... RAND_MAX`
(`0 ... 32767`)
- `srand()` - inicjalizuje generator liczb pseudolosowych
- plik nagłówkowy: `stdlib.h` (`time.h`)

```
int x, y, z;
srand((unsigned int) time(NULL));
x = rand();           // zakres <0, 32767>
y = rand() % 100;     // zakres <0, 99>
z = rand() % (b-a+1) - a; // zakres <a, b>
```

Język C - operacje na wektorze

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

```
#define N 10
```

```
int main(void)
{
```

```
    int tab1[N], i;
```

```
    /* generowanie elementów tablicy */
```

```
    srand((unsigned int) time(NULL));
```

```
    for (i=0; i<N; i++)
        tab1[i] = rand() % 20;
```

0	1	2	3	4	5	6	7	8	9
7	12	1	16	1	11	14	5	19	8

Język C - operacje na wektorze

```
/* wyświetlenie elementów tablicy */  
  
printf("Elementy tablicy:\n");  
for (i=0; i<N; i++)  
    printf("%d  ", tab1[i]);  
printf("\n");
```

Elementy tablicy:

7 12 1 16 1 11 14 5 19 8

0	1	2	3	4	5	6	7	8	9
7	12	1	16	1	11	14	5	19	8

N = 10

Język C - operacje na wektorze

```
/* wyszukanie elementu o najmniejszej wartości */  
  
int min;  
  
min = tabl[0];  
for (i=1; i<N; i++)  
    if (tabl[i]<min)  
        min = tabl[i];  
printf("Wartosc elementu najmniejszego: %d\n",min);
```

Wartosc elementu najmniejszego: 1

0	1	2	3	4	5	6	7	8	9
7	12	1	16	1	11	14	5	19	8

N = 10

Język C - operacje na wektorze

```
/* suma i średnia arytmetyczna elementów tablicy */  
  
int suma = 0;  
float srednia;  
  
for (i=0; i<N; i++)  
    suma = suma + tabl[i];  
srednia = (float) suma/N;  
printf("Suma: %d, srednia: %g\n", suma, srednia);
```

Suma: 94, srednia: 9.4

0	1	2	3	4	5	6	7	8	9
7	12	1	16	1	11	14	5	19	8

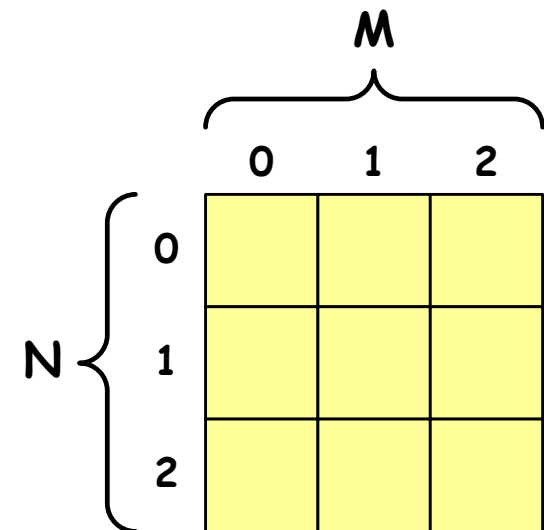
N = 10

Język C - operacje na macierzy

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

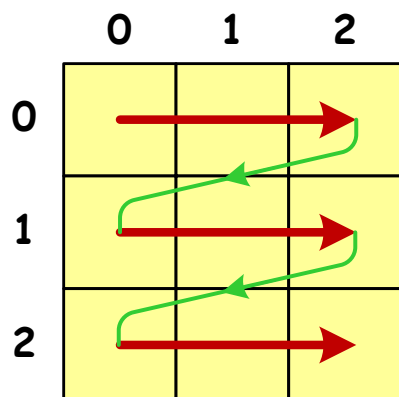
#define N 3      /* liczba wierszy */
#define M 3      /* liczba kolumn */

int main(void)
{
    int tab2[N][M];
    int i, j;
```



Język C - operacje na macierzy

```
/* generowanie pseudolosowe elementów macierzy */  
  
srand((unsigned int) time(NULL));  
  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        tab2[i][j] = rand() % 10;
```



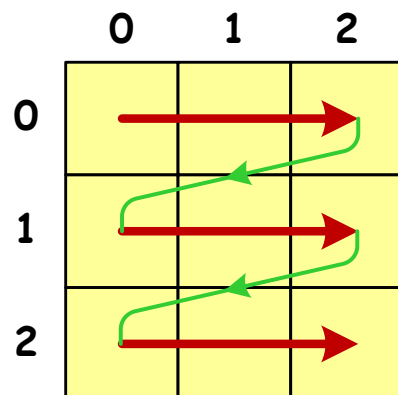
kolejność zapisywania
wartości elementów
macierzy

		M		
		0	1	2
N	0	9	3	1
	1	6	4	8
	2	9	4	6

Język C - operacje na macierzy

```
/* wyświetlenie elementów macierzy */  
  
for (i=0; i<N; i++)  
{  
    for (j=0; j<M; j++)  
        printf("%3d", tab2[i][j]);  
    printf("\n");  
}
```

9	3	1
6	4	8
9	4	6

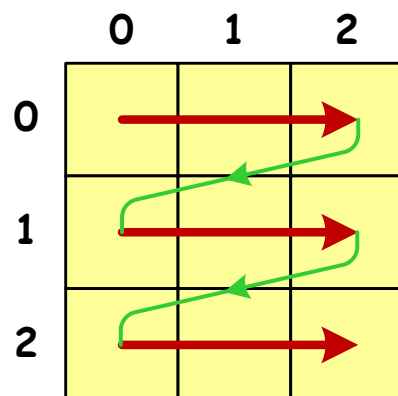


	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Język C - operacje na macierzy

```
/* poszukiwanie elementu o wartości minimalnej */  
  
int min = tab2[0][0];  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        if (tab2[i][j] < min)  
            min = tab2[i][j];  
printf("Wartosc min: %d\n",min);
```

Wartosc min: 1

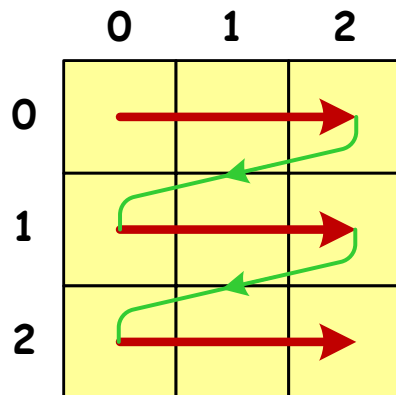


	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Język C - operacje na macierzy

```
/* sumy elementów w poszczególnych wierszach */  
  
for (i=0; i<N; i++)  
{  
    suma = 0;  
    for (j=0; j<M; j++)  
        suma = suma + tab2[i][j];  
    printf("Suma wiersza %d = %d\n", i, suma);  
}
```

	0	1	2
0			
1			
2			



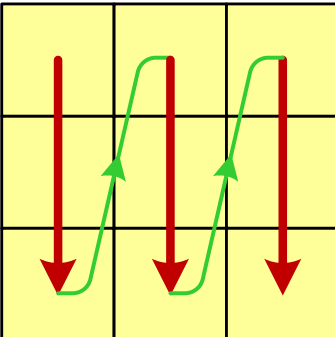
	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Suma wiersza 0 = 13
Suma wiersza 1 = 18
Suma wiersza 2 = 19

Język C - operacje na macierzy

```
/* sumy elementów w poszczególnych kolumnach */  
for (j=0; j<M; j++)  
{  
    suma = 0;  
    for (i=0; i<N; i++)  
        suma = suma + tab2[i][j];  
    printf("Suma kolumny %d = %d\n", j, suma);  
}
```

	0	1	2
0			
1			
2			



	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Suma kolumny 0 = 24
Suma kolumny 1 = 11
Suma kolumny 2 = 15

Język C - mnożenie macierzy

- Operacja: $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$
 - $\mathbf{A}[N][M]$ - macierz $N \times M$ - elementowa
 - $\mathbf{B}[M][K]$ - macierz $M \times K$ - elementowa
 - $\mathbf{C}[N][K]$ - macierz $N \times K$ - elementowa

$$\begin{matrix} N \\ \left\{ \begin{array}{ccc} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{array} \right. \\ \underbrace{\hspace{10em}} \\ K \end{matrix} = \begin{matrix} N \\ \left\{ \begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right. \\ \underbrace{\hspace{10em}} \\ M \end{matrix} \cdot \begin{matrix} \left\{ \begin{array}{ccc} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{array} \right. \\ \underbrace{\hspace{10em}} \\ K \end{matrix} \quad M$$

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}$$

$$c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}$$

$$c_{31} = a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31}$$

$$c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}$$

$$c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32}$$

$$c_{32} = a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32}$$

$$c_{13} = a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33}$$

$$c_{23} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33}$$

$$c_{33} = a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33}$$

Język C - mnożenie macierzy

- Operacja: $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ ($\mathbf{A}[N][M]$, $\mathbf{B}[M][K]$, $\mathbf{C}[N][K]$)

$$\begin{array}{c} N \\ (i) \end{array} \left\{ \begin{array}{ccc} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{array} \right\} = \begin{array}{c} N \\ (i) \end{array} \left\{ \begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right\} \cdot \begin{array}{ccc} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{array} \left\{ \begin{array}{c} M \\ (j) \end{array} \right.$$

$\underbrace{\hspace{10em}}_{K \ (k)} \quad \underbrace{\hspace{10em}}_{M \ (j)} \quad \underbrace{\hspace{10em}}_{K \ (k)}$

Program w języku C:

```
for (i=0; i<N; i++)
  for (k=0; k<K; k++)
    for (j=0; j<M; j++)
      C[i][k] = C[i][k] + A[i][j] * B[j][k];
```

$$c_{ik} = \sum_{j=1}^M a_{ij} \cdot b_{jk}, \quad \begin{array}{l} i = 1, 2, \dots, N \\ k = 1, 2, \dots, K \end{array}$$