



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Informatyka 1 (EMS1A1004)

Politechnika Białostocka - Wydział Elektryczny
Elektromobilność, semestr I, studia stacjonarne I stopnia

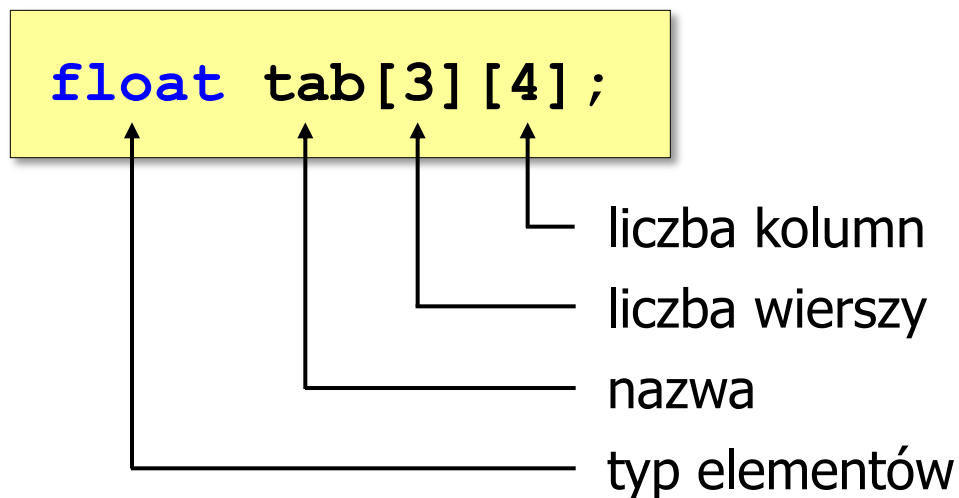
Wykład nr 8

dr inż. Jarosław Forenc

Plan wykładu nr 8

- Tablice dwuwymiarowe (macierze)
- Tablice wielowymiarowe
- Tablice o zmiennym rozmiarze (VLA)

Język C - deklaracja tablica dwuwymiarowej



	0	1	2	3	
0					← indeks kolumny
1					
2					
					↑ indeks wiersza

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

Język C - odwołania do elementów macierzy

```
tab[1][2];
```

[] - dwuargumentowy operator indeksowania

↑
↑
↑
nazwa tablicy
indeks (numer) wiersza
indeks (numer) kolumny

	0	1	2	3
0				
1				
2				

tab[1][2]
tab[2][3]
tab[2][2]

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

Język C - inicjalizacja elementów macierzy

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

	0	1	2
0	1	2	3
1	4	5	6

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

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

	0	1	2
0	1	2	3
1	4	0	0

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

	0	1	2
0	1	0	0
1	4	5	0

Język C - inicjalizacja elementów macierzy

```
int T[2][3] = {0};
```

```
int T[2][3] = {};
```

wyzerowanie elementów macierzy

	0	1	2
0	0	0	0
1	0	0	0

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

pominięcie liczby wierszy

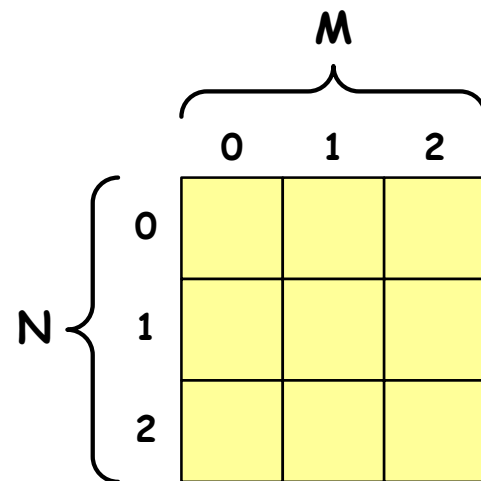
	0	1	2
0	1	2	3
1	4	5	6

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  tab[N][M];
    int  i, j;
```



Język C - operacje na macierzy

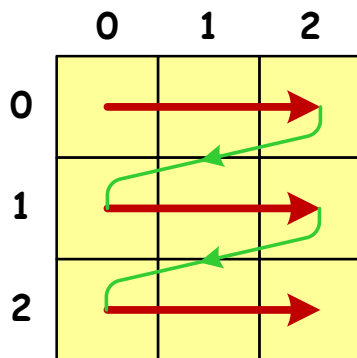
```
/* generowanie pseudolosowe elementów macierzy */
```

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

```
for (i=0; i<N; i++)
```

```
    for (j=0; j<M; j++)
```

```
        tab[i][j] = rand() % 10;
```

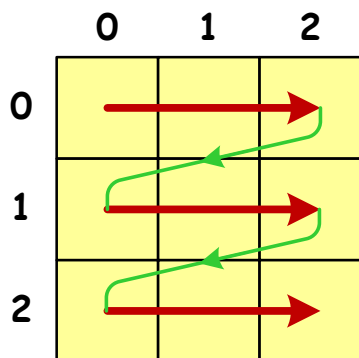


kolejność zapisywania
wartości elementów
macierzy

		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", tab[i][j]);  
    printf("\n");  
}
```



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

9	3	1
6	4	8
9	4	6

Język C - operacje na macierzy

```
/* poszukiwanie elementu o wartości minimalnej */
```

```
int min = tab[0][0];  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        if (tab[i][j] < min)  
            min = tab[i][j];  
printf("Wartosc min: %d\n",min);
```

Wartosc min: 1

	0	1	2
0			
1			
2			

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

Język C - operacje na macierzy

```
/* suma i średnia arytmetyczna elementów */  
  
int suma = 0;  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        suma = suma + tab[i][j];  
float srednia = (float) suma / (N*M);  
printf("Suma:      %d\n", suma);  
printf("Srednia:  %f\n\n", srednia);
```

	0	1	2
0			
1			
2			

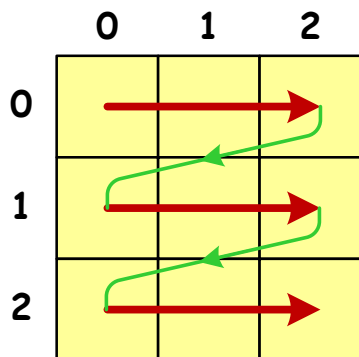
	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Suma: 50
Srednia: 5.555555

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 + tab[i][j];  
    printf("Suma wiersza %d = %d\n",i,suma);  
}
```

	0	1	2
0			
1			
2			



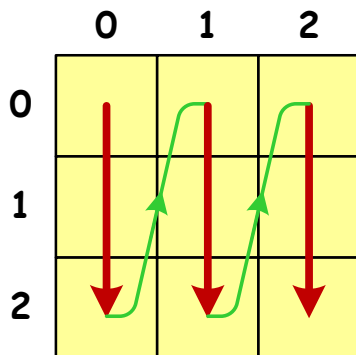
	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

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 + tab[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 - operacje na macierzy

```
/* sumy elementów nad, na i poniżej przekątnej */  
  
suma = suma1 = suma2 = 0;  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
    {  
        if (i < j) suma1+=tab[i][j]; /* nad */  
        if (i > j) suma2+=tab[i][j]; /* pod */  
        if (i == j) suma+=tab[i][j]; /* na */  
    }  
  
printf("Suma nad: %d\n", suma1);  
printf("Suma na: %d\n", suma);  
printf("Suma pod: %d\n", suma2);
```

```
Suma nad: 12  
Suma na: 19  
Suma pod: 19
```

Język C - operacje na macierzy

		j		
		0	1	2
i	0	0,0	0,1	0,2
	1	1,0	1,1	1,2
	2	2,0	2,1	2,2

$i < j$

		j		
		0	1	2
i	0	0,0	0,1	0,2
	1	1,0	1,1	1,2
	2	2,0	2,1	2,2

$i = j$

		j		
		0	1	2
i	0	0,0	0,1	0,2
	1	1,0	1,1	1,2
	2	2,0	2,1	2,2

$i > j$

	0	1	2
0			
1			
2			

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

Suma nad: 12
Suma na: 19
Suma pod: 19

Język C - tablice wielowymiarowe

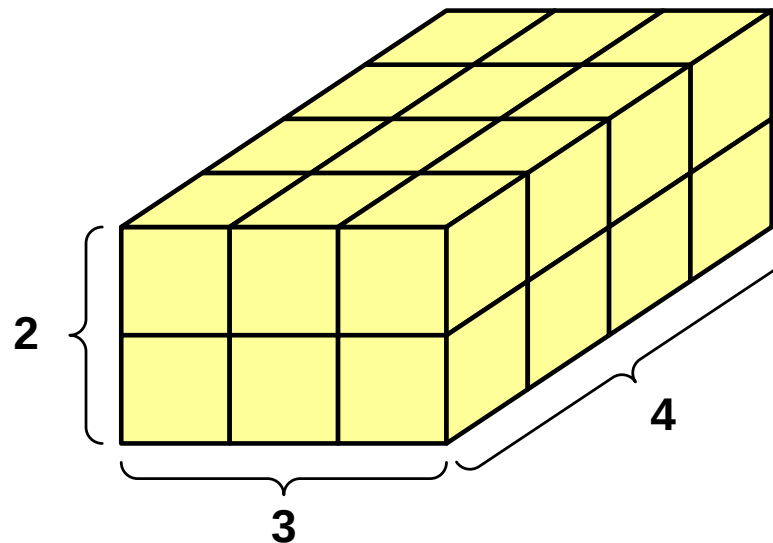
- Deklaracja tablicy wielowymiarowej

typ nazwa[wymiar_1][wymiar_2]...[wymiar_N]

- Deklaracja tablicy trójwymiarowej

```
int tab[4][2][3];
```

- Inicjalizacja i odwoływanie się do elementów są analogiczne jak w przypadku macierzy



Język C - tablice wielowymiarowe

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

```
#define X 3
```

```
#define Y 2
```

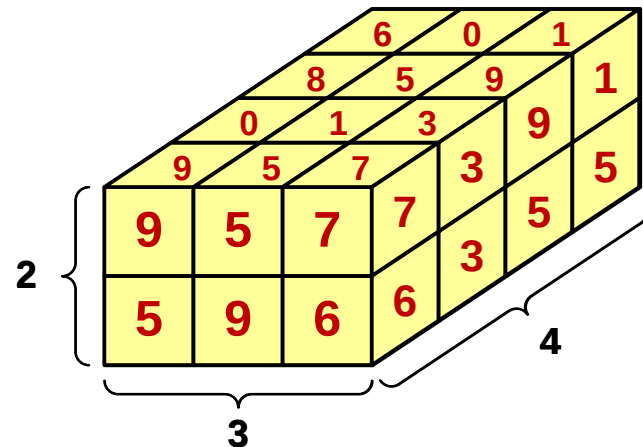
```
#define Z 4
```

```
int main(void)
```

```
{
```

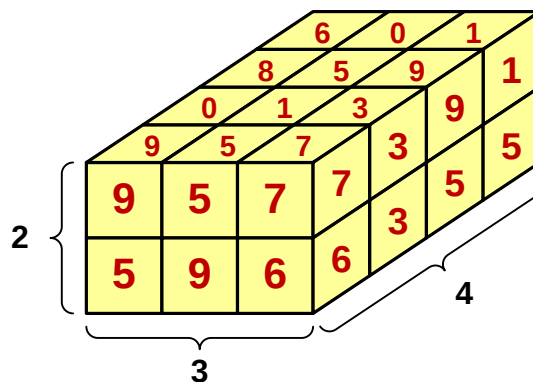
```
    int x, y, z;
```

```
    int tab[Z][Y][X] = {{{9,5,7},{5,9,6}},  
                          {{0,1,3},{7,4,3}},  
                          {{8,5,9},{1,3,5}},  
                          {{6,0,1},{8,2,5}}};
```



Język C - tablice wielowymiarowe

```
for (z=0; z<Z; z++)  
{  
    for (y=0; y<Y; y++)  
    {  
        for (x=0; x<X; x++)  
            printf("%3d", tab[z][y][x]);  
        printf("\n");  
    }  
    printf("\n");  
}  
  
return 0;  
}
```



9	5	7
5	9	6
0	1	3
7	4	3
8	5	9
1	3	5
6	0	1
8	2	5

Język C - tablice o zmiennym rozmiarze (VLA)

- **VLA** (ang. variable length array) - tablice, których rozmiar określany jest na etapie wykonywania programu (np. jako rozmiar może wystąpić nazwa zmiennej)

```
int n;  
n = 10;  
int T[n];
```

```
int n;  
scanf("%d", &n);  
int T[n];
```

- Rozmiar tablicy, a standardy języka C:
 - do standardu C99 rozmiar tablicy musiał być stałym wyrażeniem całkowitym (stała liczbowa: 5, #define N 5, const int n = 5;)
 - w standardzie C99 wprowadzono tablice o zmiennym rozmiarze
 - w standardzie C11 tablice o zmiennym rozmiarze określone są jako opcjonalne dla implementacji

Język C - tablice VLA (Visual Studio 2008 / 2019)

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    int n, i;

    printf("Rozmiar wektora: ");
    scanf("%d",&n);

    float T[n];

    for (i=0; i<n; i++)
        T[i] = sqrt((float)i);

    for (i=0; i<n; i++)
        printf("T[%d] = %f\n",i,T[i]);

    return 0;
}
```

Język C - tablice VLA (Visual Studio 2008 / 2019)

```
#include <stdio.h>
#include <math.h>
```

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

```
    int n, i;
```

```
    printf("Rozmiar wektora: ");
```

```
    scanf("%d", &n);
```

```
    float T[n];
```

```
    for (i=0; i<n; i++)
        T[i] = sqrt((float)i);
```

```
    for (i=0; i<n; i++)
        printf("T[%d] = %f\n", i, T[i]);
```

```
    return 0;
```

```
}
```

error C2057: expected constant expression
error C2466: cannot allocate an array of constant size 0
error C2133: 'T' : unknown size

error C2057: oczekiwano stałego wyrażenia
error C2466: nie można przydzielić tablicy stałego rozmiaru 0
error C2133: "T": nieznany rozmiar

Język C - tablice VLA (Dev-C++, Code::Blocks)

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    int n, i;

    printf("Rozmiar wektora: ");
    scanf("%d", &n);

    float T[n];

    for (i=0; i<n; i++)
        T[i] = sqrt((float)i);

    for (i=0; i<n; i++)
        printf("T[%d] = %f\n", i, T[i]);

    return 0;
}
```

```
Rozmiar wektora: 8
T[0] = 0.000000
T[1] = 1.000000
T[2] = 1.414214
T[3] = 1.732051
T[4] = 2.000000
T[5] = 2.236068
T[6] = 2.449490
T[7] = 2.645751
```

Język C - tablice VLA

- Tablica VLA może być także tablicą dwu- lub wielowymiarową

```
int n = 5, m = 6;  
int T1[n][m], T2[n][m][n];
```

- Nie można modyfikować rozmiaru tablic VLA po deklaracji
- Tablice VLA nie mogą być inicjalizowane podczas deklaracji
 - błędy i ostrzeżenia w **Code::Blocks**

```
error: variable-sized object may not be initialized  
warning: excess elements in array initializer  
warning: (near initialization for 'T')
```

- w **Dev-C++** inicjalizacja jest dopuszczalna!

Koniec wykładu nr 8

Dziękuję za uwagę!



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Materiały przygotowane w ramach projektu „PB 5.0 - dostosowanie oferty dydaktycznej Politechniki Białostockiej do potrzeb nowoczesnej gospodarki oraz zielonej i cyfrowej transformacji” (nr umowy FERS.01.05-IP.08-0327/23-00) w ramach programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Społecznego Plus.

Publikacja jest udostępniona na licencji Creative Commons - Uznanie autorstwa 4.0 (CC BY 4.0). Pełna treść licencji dostępna na stronie creativecommons.org.