

# Informatyka 2 (EZ1E3012)

Politechnika Białostocka - Wydział Elektryczny  
Elektrotechnika, semestr III, studia niestacjonarne I stopnia  
Rok akademicki 2022/2023

**Pracownia nr 1 (08/09.10.2022)**

dr inż. Jarosław Forenc

## Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,  
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki  
ul. Wiejska 45D, 15-351 Białystok  
WE-204
- e-mail: [j.forenc@pb.edu.pl](mailto:j.forenc@pb.edu.pl)
- tel. (0-85) 746-93-97
- <http://jforenc.prv.pl>
  - Dydaktyka - dodatkowe materiały do pracowni
- Konsultacje
  - poniedziałek, 08:30-10:00, 17:40-18:25, WE-204
  - środa, 16:00-16:45, WE-204
  - sobota, 13:00-13:30, WE-204 (studia zaoczne)
  - niedziela, 16:00-17:00, WE-204 (studia zaoczne)

## Program przedmiotu

1. Zajęcia organizacyjne. Tablice dwuwymiarowe, operacje na tablicach dwuwymiarowych.
2. Łańcuchy znaków. Plik nagłówkowy string.h.
3. Struktury, odwołania do pól struktury. Inicjalizacja zmiennej strukturalnej.
4. Wskaźniki. Dynamiczny przydział pamięci w języku C.
5. Funkcje, ogólna struktura funkcji. Umieszczanie definicji funkcji w programie. Przekazywanie argumentów do funkcji przez wartość i wskaźnik. Rekurencyjne wywołanie funkcji.
6. **Kolokwium nr 1.** Programy wielomodułowe.

## Program przedmiotu

7. Zaawansowane operacje wejścia-wyjścia w języku C. Pliki tekstowe w języku C.
8. Pliki binarne w języku C.
9. Operacje na plikach tekstowych i binarnych. **Kolokwium nr 2.**
10. Operatory bitowe.

## Literatura

1. S. Prata: „Język C. Szkoła programowania. Wydanie VI”. Helion, 2016.
2. B.W. Kernighan, D.M. Ritchie: „Język ANSI C. Programowanie. Wydanie II”. Helion, Gliwice, 2010.
3. P.J. Deitel, H. Deitel: „Język C. Solidna wiedza w praktyce. Wydanie VIII”. Helion, Gliwice, 2020.
4. K.N. King: „Język C. Nowoczesne programowanie. Wydanie II”. Helion, Gliwice, 2011.
5. S.G. Kochan: „Język C. Kompendium wiedzy. Wydanie IV”. Helion, Gliwice, 2015.
6. R. Reese: „Wskaźniki w języku C. Przewodnik”. Helion, Gliwice, 2014.
7. Instrukcje do pracowni specjalistycznej.
8. Materiały na stronie internetowej:  
<http://jforenc.prv.pl/dydaktyka.html>

## Warunki zaliczenia przedmiotu

- Zaliczenie dwóch kolokwίων - praktycznych sprawdzianów pisania programów komputerowych (c.d.):
  - poprawy kolokwίων odbędą się poza zajęciami
  - oba kolokwia muszą być zaliczone na ocenę pozytywną (min. 51 pkt.)
  - na kolokwiach można korzystać z materiałów (instrukcje do przedmiotu, książki, itp.)
  - za każde kolokwium można otrzymać od 0 do 100 pkt.
  - na podstawie otrzymanych punktów wystawiana jest ocena:

| Punkty   | Ocena | Punkty  | Ocena |
|----------|-------|---------|-------|
| 91 - 100 | 5,0   | 61 - 70 | 3,5   |
| 81 - 90  | 4,5   | 51 - 60 | 3,0   |
| 71 - 80  | 4,0   | 0 - 50  | 2,0   |

## Warunki zaliczenia przedmiotu

- Obecność na zajęciach:
  - więcej niż dwie nieusprawiedliwione nieobecności skutkują niezaliczeniem pracowni
- Realizacja w trakcie zajęć zadań przedstawionych przez prowadzącego
- Zaliczenie dwóch kolokwίων - praktycznych sprawdzianów pisania programów komputerowych
  - kolokwium nr 1: zajęcia nr 6 (ocena efektów EU1 i EU2)
  - kolokwium nr 2: zajęcia nr 9 (ocena efektów EU3 i EU4)

## Warunki zaliczenia przedmiotu

- Zaliczenie prac domowych:
  - **prace domowe** polegają na napisaniu programów komputerowych wskazanych przez prowadzącego zajęcia
  - pracę domową należy wysłać na adres e-mailowy: [j.forenc@pb.edu.pl](mailto:j.forenc@pb.edu.pl) do końca dnia (godz. 23:59), w którym odbywają się **kolejne** zajęcia
  - błędy w programach mogą być poprawione do końca dnia, w którym będą odbywały się **następne** zajęcia
  - za każdy poprawnie działający program student otrzymuje 1 pkt.
  - w przypadku stwierdzenia niesamodzielności pracy domowej: nie jest ona zaliczana, nie można jej ponownie oddać, student otrzymuje -1 pkt.
  - należy wysłać tylko pliki z kodem źródłowym (.cpp)
  - przed wysłaniem programu należy sprawdzić czy program:
    - kompiluje się oraz kompilator nie wyświetla ostrzeżeń
    - jest poprawnym rozwiązaniem zadania

## Warunki zaliczenia przedmiotu

- przed wysłaniem programu należy sprawdzić czy program:
  - nosi nazwę zgodną z poniższym wzorcem:  
**Nazwisko\_Imie\_NrGrupy\_NrInstrukcji\_NrZadania.cpp**  
np. **Kowalski\_Jan\_PS1\_INF21Z\_3.cpp**
  - zawiera na początku komentarz nagłówkowy:

```
/*  
Nazwa: Kowalski_Jan_PS1_INF21Z_3.cpp  
Autor: Jan Kowalski, gr. PS1  
Album: 123456  
Data: 08-10-2022  
Kod: EZ1E3012  
Forma: Pracownia specjalistyczna  
IDE: Visual Studio 2008  
Opis: Program wyświetlający tekst "Witaj świecie"  
*/
```

## Warunki zaliczenia przedmiotu

- Zaliczenie prac domowych (c.d.):
  - zaliczenie prac domowych wymaga otrzymania min. **70%** maksymalnej liczby punktów
- Ocena końcowa wyznaczana jest na podstawie sumy otrzymanych punktów za kolokwia i prace domowe:

| Punkty    | Ocena | Punkty    | Ocena |
|-----------|-------|-----------|-------|
| 182 - 200 | 5,0   | 122 - 141 | 3,5   |
| 162 - 181 | 4,5   | 102 - 121 | 3,0   |
| 142 - 161 | 4,0   | 0 - 101   | 2,0   |

## Efekty uczenia się i system ich oceniania

Podstawę do zaliczenia przedmiotu (uzyskanie punktów ECTS) stanowi stwierdzenie, że każdy z założonych **efektów uczenia się** został osiągnięty.

|            |   |
|------------|---|
| <b>EU1</b> | wykonuje podstawowe operacje na tablicach dwuwymiarowych w programach w języku C            |
| <b>EU2</b> | definiuje i wykorzystuje własne funkcje w programach w języku C                             |
| <b>EU3</b> | tworzy programy wielomodułowe w języku C  |
| <b>EU4</b> | stosuje operacje zapisu i odczytu plików w samodzielnie napisanych programach komputerowych |

## Język C - tablica elementów

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

wektor 

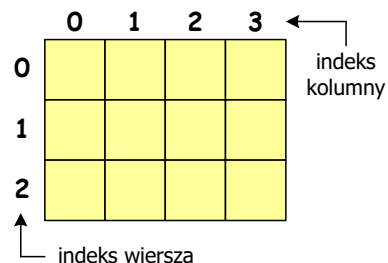
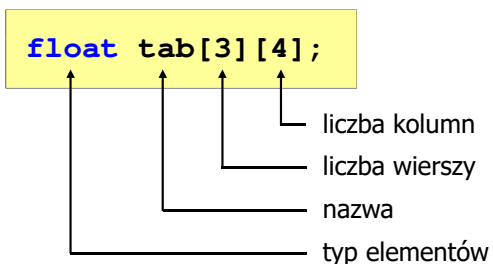
|   |   |    |   |    |
|---|---|----|---|----|
| 5 | 3 | -2 | 1 | -4 |
|---|---|----|---|----|

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 tablica dwuwymiarowej



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

## 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};
```

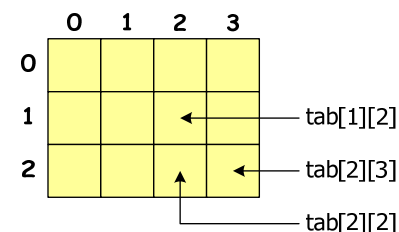
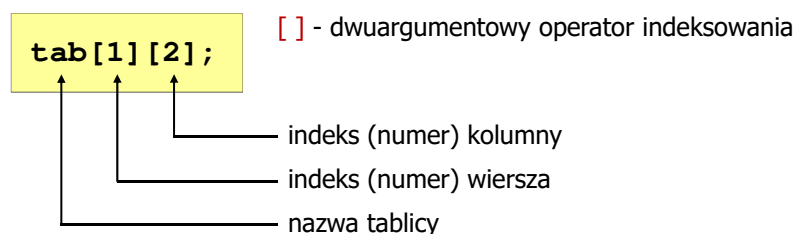
```
int T[2][3] = {1, 2, 3, 4};
```

|   | 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 - odwołania do elementów macierzy



- Indeks:
  - stała liczbowo, 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] = {0};
```

|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |

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

wyzerowanie elementów macierzy

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

|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 |

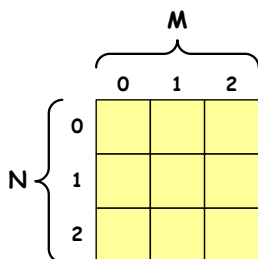
pominięcie liczby wierszy

## Język C - operacje na macierzy

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

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

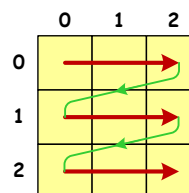
int main(void)
{
    int tab[N][M];
    int i, j;
```



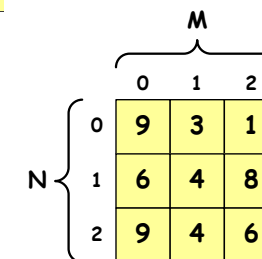
## Język C - operacje na macierzy

```
/* generowanie pseudolosowe elementow 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

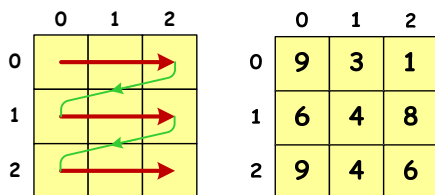


## Język C - operacje na macierzy

```
/* wyswietlenie elementow macierzy */

for (i=0; i<N; i++)
{
    for (j=0; j<M; j++)
        printf("%3d", tab[i][j]);
    printf("\n");
}
```

```
9 3 1
6 4 8
9 4 6
```

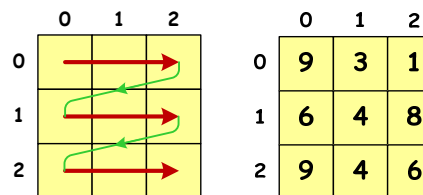


## Język C - operacje na macierzy

```
/* poszukiwanie elementu o wartosci 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
```



## Język C - operacje na macierzy

```
/* suma i srednia arytmetyczna elementow */  
  
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 | 9 | 3 | 1 |
| 1 | 6 | 4 | 8 |
| 2 | 9 | 4 | 6 |

|   | 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 elementow w poszczegolnych 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 | 9 | 3 | 1 |
| 1 | 6 | 4 | 8 |
| 2 | 9 | 4 | 6 |

|   | 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 elementow w poszczegolnych 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 | 9 | 3 | 1 |
| 1 | 6 | 4 | 8 |
| 2 | 9 | 4 | 6 |

|   | 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