Fundusze Europejskie Wiedza Edukacja Rozwój | Rzeczpospolita Polska | Unia Europejska Europejski Fundusz Społeczny

Wydział Elektryczny
Katedra Elektrotechniki Teoretycznej i Metrologii

Materiały do wykładu z przedmiotu:
Informatyka
Kod: EDS1B1007

WYKŁAD NR 6

Opracował: dr inż. Jarosław Forenc

Białystok 2019

Materiały zostały opracowane w ramach projektu „PB2020 - Zintegrowany Program Rozwoju Politechniki Białostockiej" realizowanego w ramach Działania 3.5 Programu Operacyjnego Wiedza, Edukacja, Rozwój 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Społecznego.

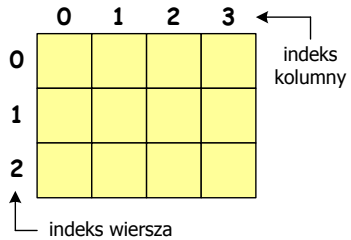
Plan wykładu nr 6

- Tablice dwuwymiarowe (macierze)
- Łańcuchy znaków w języku C

Język C - deklaracja tablica dwuwymiarowej

```
float tab[3][4];
```

liczba kolumn
liczba wierszy
nazwa
typ elementów



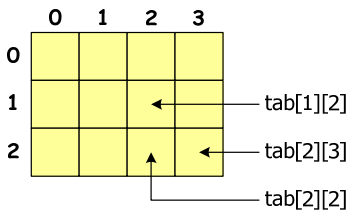
- Rozmiar tablicy (liczb wierszy i kolumn) to wartość:
  - całkowita, dodatnia
  - znana na etapie kompilacji programu
    (stała liczbowa: 5, #define N 5, const int n = 5;)

Język C - odwołania do elementów macierzy

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

[ ] - dwuargumentowy operator indeksowania

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



- 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};
```

	0	1	2
0	1	2	3
1	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};
```

wyzerowanie elementów macierzy

	0	1	2
0	0	0	0
1	0	0	0

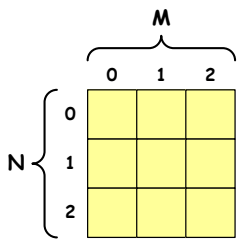
```
int T[][3] = {{1,2,3},{4,5,6}};
```

pominięcie liczby wierszy

	0	1	2
0	1	2	3
1	4	5	6

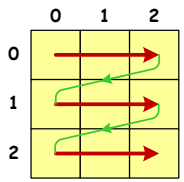
Język C - operacje na macierzy

```
#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;
```

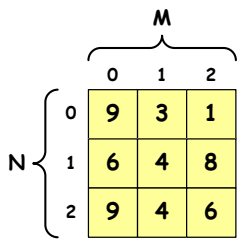


Język C - operacje na macierzy

```
/* 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;
```



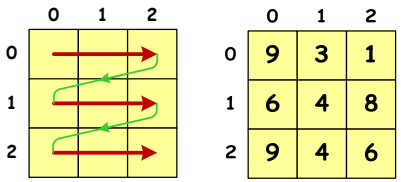
kolejność zapisywania wartości elementów macierzy



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");
}
```

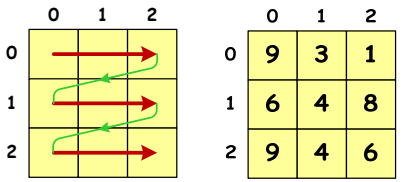
```
  9   3   1
  6   4   8
  9   4   6
```



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);
```

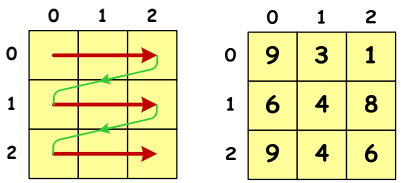
```
Wartosc min: 1
```



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);
```

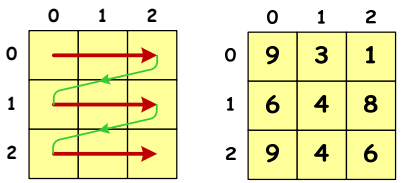
```
Suma:    50
Srednia: 5.555555
```



Język C - operacje na macierzy

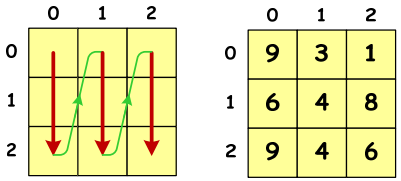
```
/* sumy elementów w poszczególnych wierszach */

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);
}
```

```
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);
}
```



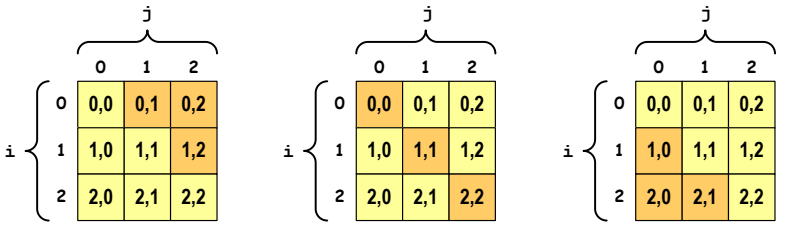

```
Suma kolumny 0 = 24
Suma kolumny 1 = 11
Suma kolumny 2 = 15
```

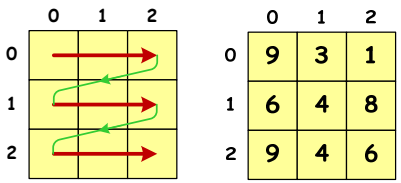

Język C - operacje na macierzy

```
/* 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





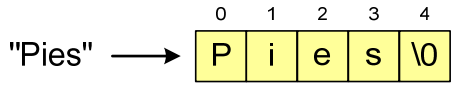

```
Suma nad: 12
Suma na:  19
Suma pod: 19
```


Język C - łańcuchy znaków

- Łańcuch znaków (ciąg znaków, napis, literał łańcuchowy, stała łańcuchowa, C-string) - ciąg złożony z zera lub większej liczby znaków zawartych między znakami cudzysłowu

```
"Pies"
```

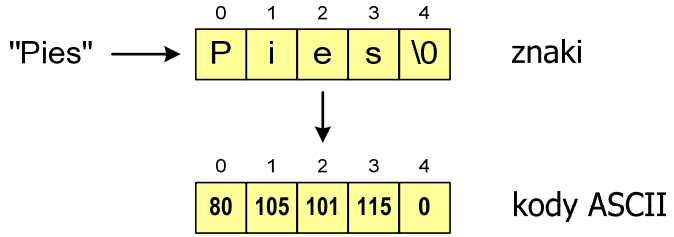
- Implementacja - tablica, której elementami są pojedyncze znaki (typ char)



- Ostatni znak (\0, liczba zero, znak zerowy) oznacza koniec napisu

Język C - łańcuchy znaków

- W rzeczywistości w tablicy zamiast znaków przechowywane są odpowiadające im kody ASCII (czyli liczby)



Język C - deklaracja łańcucha znaków

- Deklaracja zmiennej przechowującej łańcuch znaków

```
char nazwa_zmiennej[rozmiar];
```

Przykład:

```
chat txt[10];
```

- Tablica txt może przechowywać napisy o maksymalnej długości do 9 znaków

Język C - inicjalizacja łańcucha znaków

- Inicjalizacja łańcucha znaków

```
char txt1[10] = "Pies";
char txt2[10] = {'P','i','e','s'};
char txt3[10] = {80,105,101,115};
```

- Pozostałe elementy tablicy otrzymują wartość zero

P	i	e	s	\0	\0	\0	\0	\0	\0

```
char txt4[]   = "Pies";
char *txt5    = "Pies";
```


Język C - inicjalizacja łańcucha znaków

- Inicjalizacja możliwa jest tylko przy deklaracji

```
char txt[10];
txt = "Pies";    /* BŁĄD!!! */
```

- Przypisanie zmiennej txt wartości "Pies" wymaga zastosowania funkcji strcpy() z pliku nagłówkowego string.h

```
char txt[10];
strcpy(txt,"Pies");
```

Język C - stała znakowa

- Stałą znakową tworzy jeden znak ujęty w apostrofy

```
char zn = 'x';
```

- W rzeczywistości stała znakowa jest to liczba całkowita, której wartość odpowiada wartości kodu ASCII reprezentowanego znaku
- Zamiast powyższego kodu można napisać:

```
char zn = 120;
```

- Uwaga:
  - `'x'` - stała znakowa (jeden znak)
  - `"x"` - łańcuch znaków (dwa znaki: x oraz \0)

Język C - stała znakowa

- Niektóre znaki mogą być reprezentowane w stałych znakowych przez sekwencje specjalne, które wyglądają jak dwa znaki, ale reprezentują tylko jeden znak

`'\n'`	- nowy wiersz	`'\\'`	- \ (ang. backslash)
`'\t'`	- tabulator poziomy	`'\''`	- apostrof
`'\v'`	- tabulator pionowy	`'\"'`	- cudzysłów
`'\a'`	- alarm	`'\?'`	- znak zapytania

Język C - wyświetlenie tekstu

- Wyświetlenie tekstu funkcją printf() wymaga specyfikatora %s

```
char napis[15] = "Jan Kowalski";
printf("Osoba: [%s]\n", napis);
```

```
Osoba: [Jan Kowalski]
```

- W specyfikatorze %s: szerokość określa szerokość pola, zaś precyzja - liczbę pierwszych znaków z łańcucha

```
char napis[15] = "Jan Kowalski";
printf("[%10.6s]\n", napis);
```

```
[    Jan Ko]
```


Język C - wyświetlenie tekstu

- Do wyświetlenia tekstu można zastosować funkcję puts()

`puts()`	`int puts(const char *s);`

- Funkcja puts() wypisuje na stdout (ekran) zawartość łańcucha znakowego (ciąg znaków zakończony znakiem '\0'), zastępując znak '\0' znakiem '\n'

```
char napis[15] = "Jan Kowalski";
puts(napis);
```

```
Jan Kowalski
```

Język C - wyświetlenie tekstu

- Wyświetlenie znaku funkcją printf() wymaga specyfikatora %c

```
char zn = 'x';
printf("Znak to: [%c]\n", zn);
```

```
Znak to: [x]
```


Język C - wyświetlenie tekstu

- Łańcuch znaków jest zwykłą tablicą - można więc odwoływać się do jej pojedynczych elementów

```
char txt[15] = "Ola ma laptopa";

printf("Znaki: ");
for (int i=0; i<15; i++) printf("%c ",txt[i]);
printf("\n");

printf("Kody:  ");
for (int i=0; i<15; i++) printf("%d ",txt[i]);
printf("\n");
```

```
Znaki: O l a   m a   l a p t o p a
Kody:  79 108 97 32 109 97 32 108 97 112 116 111 112 97 0
```


Język C - wczytanie tekstu

- Do wczytania tekstu funkcją scanf() stosowany jest specyfikator %s

```
char napis[15];
scanf("%s", napis);
```

brak znaku &

- W specyfikatorze formatu %s można podać szerokość

```
char napis[15];
scanf("%10s", napis);
```

- W powyższym przykładzie scanf() zakończy wczytywanie tekstu po pierwszym białym znaku (spacja, tabulacja, enter) lub w momencie pobrania 10 znaków

Język C - wczytanie tekstu

- W przypadku wprowadzenia tekstu "To jest napis", funkcja scanf() zapamięta tylko wyraz "To"
- Zapamiętanie całego wiersza tekstu (do naciśnięcia klawisza Enter) wymaga użycia funkcji gets()

gets()	char *gets(char *s);

- Funkcja gets() wprowadza wiersz (ciąg znaków zakończony '\n') ze strumienia stdin (klawiatura) i umieszcza w obszarze pamięci wskazywanym przez wskaźnik s zastępując '\n' znakiem '\0'

```
char napis[15];
gets(napis);
```

Język C - plik nagłówkowy string.h

strcpy()	char *strcpy(char *s1, const char *s2);

- Kopiuje łańcuch s2 do łańcucha s1

strlen()	size_t strlen(const char *s);

- Zwraca długość łańcucha znaków, nie uwzględnia znaku '\0'

strcat()	char *strcat(char *s1, const char *s2);

- Dołącza do łańcucha s1 łańcuch s2

Język C - plik nagłówkowy string.h

strcmp()	int strcmp(const char *s1, const char *s2);

- Porównuje łańcuchy s1 i s2 z rozróżnianiem wielkości liter

strcmpi()	int strcmpi(const char *s1, const char *s2);

- Porównuje łańcuchy s1 i s2 bez rozróżniania wielkości liter

strchr()	char *strchr(const char *s, int c);

- Szuka w łańcuchu s znaku c

Język C - plik nagłówkowy string.h

strlwr()	char *strlwr(char *s);

- Zamienia w łańcuchu s wielkie litery na małe

strupr()	char *strupr(char *s);

- Zamienia w łańcuchu s małe litery na wielkie

strrev()	char *strrev(char *s);

- Odwraca kolejność znaków w łańcuchu s

Język C - plik nagłówkowy string.h (przykład)

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

int main(void)
{
    char napis1[] = "Tekst w buforze", napis2[20];

    printf("napis1: %s \n",napis1);
    int dlugosc = strlen(napis1);
    printf("liczba znakow w napis1: %d \n",dlugosc);
    strcpy(napis2,napis1);
    printf("napis2: %s \n",napis2);
    strrev(napis2);
    printf("napis2 (odwr): %s \n",napis2);

    return 0;
}
```

Język C - plik nagłówkowy string.h (przykład)

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

int main(void)
{
    char napis1[] = "

    printf("napis1: %s \n",napis1);
    int dlugosc = strlen(napis1);
    printf("liczba znakow w napis1: %d \n",dlugosc);
    strcpy(napis2,napis1);
    printf("napis2: %s \n",napis2);
    strrev(napis2);
    printf("napis2 (odwr): %s \n",napis2);

    return 0;
}
```

```
napis1: Tekst w buforze
liczba znakow w napis1: 15
napis2: Tekst w buforze
napis2 (odwr): ezrofub w tskeT
```


Język C - macierz elementów typu char

- Szczególny przypadek tablicy dwuwymiarowej

```
char txt[3][15] = {"Programowanie",
                   "nie jest",
                   "trudne"};
```

- Tablica w pamięci komputera

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	P	r	o	g	r	a	m	o	w	a	n	i	e	\0	\0
1	n	i	e		j	e	s	t	\0	\0	\0	\0	\0	\0	\0
2	t	r	u	d	n	e	\0	\0	\0	\0	\0	\0	\0	\0	\0

Język C - macierz elementów typu char

- Używając dwóch indeksów (nr wiersza i nr kolumny) można odwoływać się do jej pojedynczych elementów (znaków)
- Użycie jednego indeksu (numeru wiersza) powoduje potraktowanie całego wiersza jako łańcuch znaków (napisu)

```
char txt[3][15] = {"Programowanie",
                   "nie jest",
                   "trudne"};
printf("%s ",txt[1]);
printf("%s ",txt[2]);
printf("%s ",txt[0]);
```

```
nie jest trudne Programowanie
```


Koniec wykładu nr 6

Dziękuję za uwagę!