Informatyka 1

Politechnika Białostocka - Wydział Elektryczny

Elektrotechnika, semestr II, studia stacjonarne I stopnia

Rok akademicki 2017/2018

Pracownia nr 9
(08/11.05.2018)

dr inż. Jarosław Forenc

Tablica elementów

- Tablica - ciągły obszar pamięci, w którym umieszczone są elementy tego samego typu
- Wektor - tablica jednowymiarowa

5	3	-2	0	-4

- liczby całkowite

3.1	0.2	2.3	-1.3	1.5	1.1	-4.0

- liczby rzeczywiste

a	Z	x	&	M	+

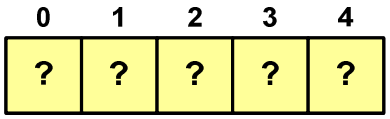
- znaki

Deklaracja tablicy

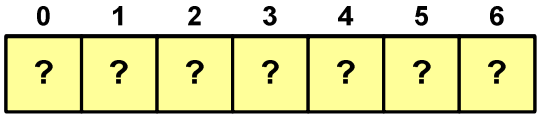
```
typ nazwa[rozmiar];
```

indeks

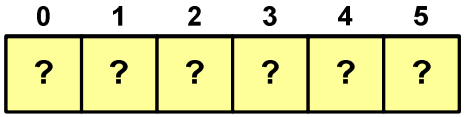
```
int tab[5];
```




```
double tab1[7];
```



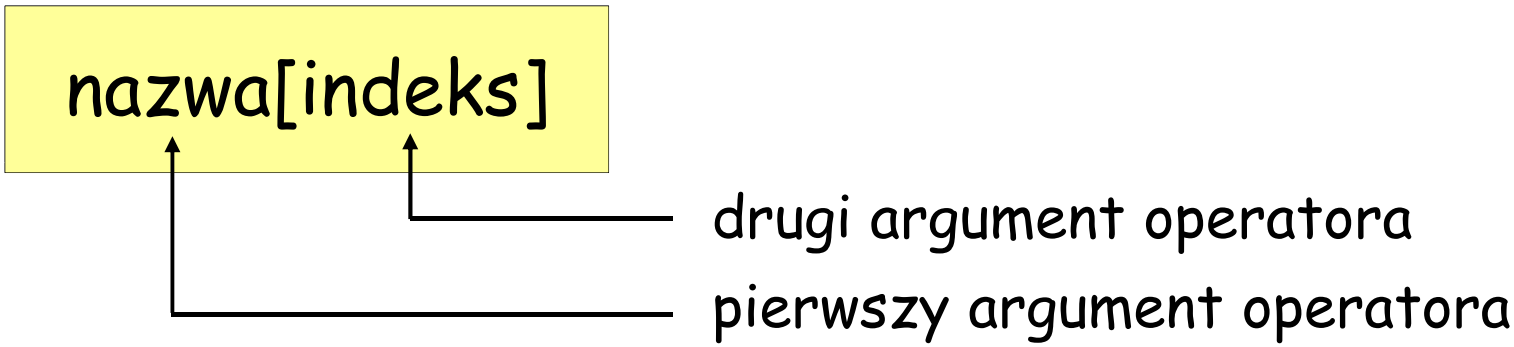

```
char tab2[6];
```

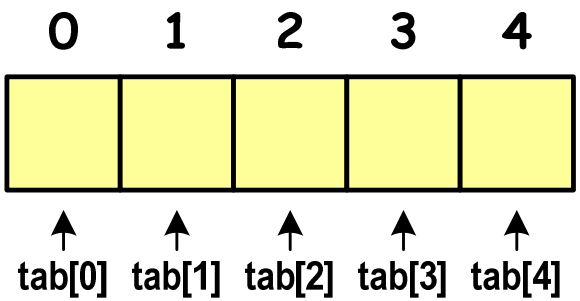


- rozmiar tablicy to wartość:
  - całkowita, dodatnia, znana na etapie kompilacji programu
    (stała liczbowa: 5, #define N 7, const int n = 6;)

Odwołania do elementów tablicy

- [ ] - dwuargumentowy operator indeksowania

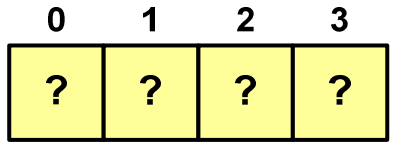




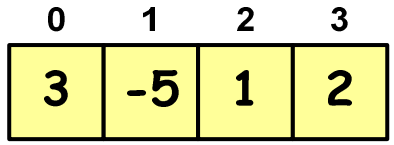
- indeks:
  - stała liczbowa, np. 0, 1, 10
  - nazwa zmiennej, np. i, idx
  - wyrażenie, np. i*j+5

Odwołania do elementów tablicy

```
int tab[4];
```




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



- Każdy element tablicy traktowany jest tak samo jak zmienna typu int

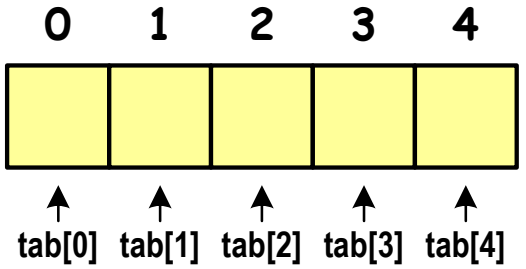
```
printf("%d",tab[0]);
```

```
scanf("%d",&tab[0]);
```

Odwołania do elementów tablicy

- przy odwołaniach do elementów tablicy kompilator nie sprawdza poprawności indeksów

```
int tab[5];
tab[5] = 10;
```

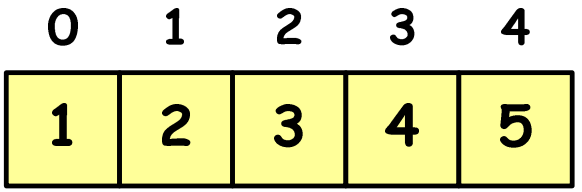


- błąd!!! - nie istnieje element tab[5]

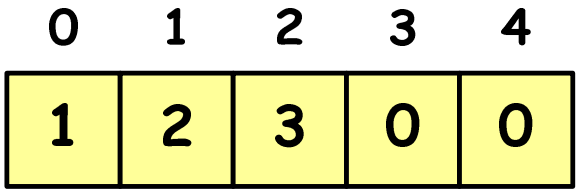
- kompilator nie zasygnalizuje błędu i wykona operację

Inicjalizacja wektora

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



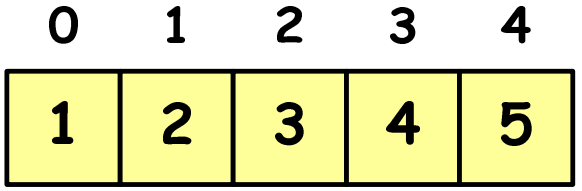

```
int tab[5] = { 1, 2, 3 };
```




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

- błąd kompilacji

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



Odwołania do elementów tablicy

- zapisujemy do elementów tablicy kolejne liczby: 1, 2, ..., 5

```
int tab[5];

tab[0] = 1;
tab[1] = 2;
tab[2] = 3;
tab[3] = 4;
tab[4] = 5;
```

```
int tab[5], i;

for (i=0; i<5; i++)
   tab[i] = i+1;
```

Przykład - operacje na wektorze

```
#include <stdio.h>
#define N 10

int main(void)
{
    int  tab[N];
    int  i;

    /* wczytanie elementów wektora */

    for (i=0; i<N; i++)
    {
        printf("Podaj liczbe nr %d: ",i+1);
        scanf("%d",&tab[i]);
    }
```

Przykład - operacje na wektorze

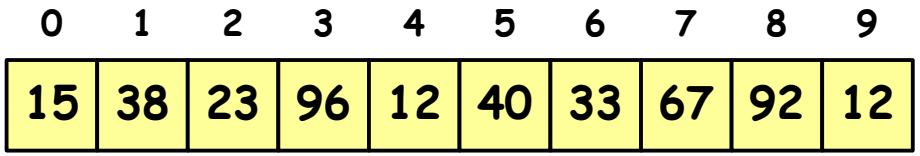
```
#include <stdio.h>
#define N 10

int main(void)
{
    int  tab[N];
    int  i;

    /* wczytanie elementów wektora */
    for (i=0; i<N; i++)
    {
        printf("Podaj liczbe nr %d: ",i+1
        scanf("%d",&tab[i]);
    }
```

```
Podaj liczbe nr 1: 15
Podaj liczbe nr 2: 38
Podaj liczbe nr 3: 23
Podaj liczbe nr 4: 96
Podaj liczbe nr 5: 12
Podaj liczbe nr 6: 40
Podaj liczbe nr 7: 33
Podaj liczbe nr 8: 67
Podaj liczbe nr 9: 92
Podaj liczbe nr 10: 12
```

Przykład - operacje na wektorze

```
/* wyświetlenie elementów wektora */
printf("Elementy wektora:\n");
for (i=0; i<N; i++)
    printf("%4d", tab[i]);
printf("\n");
```

```
Elementy wektora:
15  38  23  96  12  40  33  67  92  12
```



N = 10

Przykład - operacje na wektorze

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

Suma: 428, srednia: 42.799999

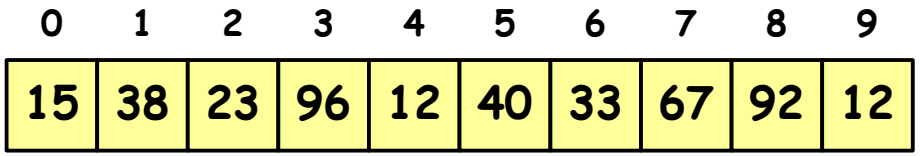
0	1	2	3	4	5	6	7	8	9
15	38	23	96	12	40	33	67	92	12

N = 10

Przykład - operacje na wektorze

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

```
Wartosc elementu najmniejszego: 12
```



N = 10

Przykład - operacje na wektorze

```
/* generowanie pseudolosowe elementów wektora */
srand((unsigned int) time(NULL));    /* stdlib.h, time.h */
for (i=0; i<N; i++)
    tab[i] = rand() % 100;                /* stdlib.h */
```

```
4  82  17  96  87  79  18  10  11  25
```

- srand((unsigned int) time(NULL)); - inicjalizacja generatora
- rand() - zwraca liczbę pseudolosową z zakresu 0 ... 32767
- rand() % 100 - zwraca liczbę pseudolosową z zakresu 0 ... 99