

Programowanie strukturalne - Wykład 6

Tablice jednowymiarowe

Definicja

Tablica - ciąg elementów jednego typu.

Deklaracja:

```
1   int tab[4];  
2   char tab2[3];  
3   float tabf[5];
```

Inicjalizacja

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

State tablice

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

Tworzenie przez stałą DEFINE

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define ROZMIAR 3
4
5 int main()
6 {
7     int tab[ROZMIAR] = {3,-4,3};
8     return 0;
9 }
```

Brak inicjalizacji?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 3
4
5  int main()
6  {
7      int tab[ROZMIAR];
8      for(int i=0;i<ROZMIAR;i++)
9          {
10             printf("%d\n",tab[i]);
11         }
12     return 0;
13 }
```

Za mało wartości?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 3
4
5  int main()
6  {
7      int tab[ROZMIAR] = {2, 3};
8      for(int i=0; i<ROZMIAR; i++)
9      {
10         printf("%d\n", tab[i]);
11     }
12     return 0;
13 }
```


Za dużo wartości?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 2
4
5  int main()
6  {
7      int tab[ROZMIAR] = {2, 3, -2, 0};
8      for(int i=0; i<ROZMIAR; i++)
9      {
10         printf("%d\n", tab[i]);
11     }
12     return 0;
13 }
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int tab[] = {2, 3, -2, 0};
7     for(int i=0; i<sizeof tab/sizeof(int); i++)
8     {
9         printf("%d\n", tab[i]);
10    }
11    return 0;
12 }
```

Oznaczona inicjalizacja - od C99

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 5
4
5  int main()
6  {
7      int tab[ROZMIAR] = {[2] = 34};
8      for(int i=0;i<ROZMIAR;i++)
9      {
10         printf("%d\n",tab[i]);
11     }
12     return 0;
13 }
```

Przypisanie wartości

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 5
4
5  int main()
6  {
7      int tab[ROZMIAR] = {[2] = 34};
8      tab[0]=2;
9      for(int i=0;i<ROZMIAR;i++)
10     {
11         printf("%d\n",tab[i]);
12     }
13     return 0;
14 }
```

Zakres tablic

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 5
4
5  int main()
6  {
7      int tab[ROZMIAR] = {1,-2,3};
8      printf("%d",tab[-1]);
9      return 0;
10 }
```

Jak określić rozmiar inaczej niż przez stałą DEFINE?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int n = 5;
7      int m = 8;
8      int tabb1[5]; // ok
9      int tabb2[3*2+1]; // ok
10     int tabb3[sizeof(int)+1]; // ok
11     //int tabb4[-2]; // nie, bo <0
12     int tabb5[0]; // ???
13     //int tabb6[6.5]; // nie, bo nie calc.
14     //int tabb7[(int)2.5]; // ???
15     int tabb8[n]; // brak dla C90
16     int tabb9[m]; // Brak dla C90
17     return 0;
18 }
```

Wskaźniki do tablic

Nazwa tablicy jest równocześnie adresem jej pierwszego elementu.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define ROZMIAR 3
4
5  int main()
6  {
7      int tab[ROZMIAR] = {4, 0, -2};
8      printf("%i", tab==&tab[0]);
9      return 0;
10 }
```

Równoważne:

```
1 tab + 3 == &tab[3] // ten sam adres  
2 *(tab + 3) == tab[3] // ta sama wartosc
```

Uwaga na nawiasy:

```
1 *(tab +2) // wartosc elementu tablicy o indeksie 2  
2 *tab +2 // 2 dodane do wartosci elementu o indeksie 0
```


Kod do analizy

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define ROZMIAR 3
4
5 int main()
6 {
7     int tab[ROZMIAR] = {4, 7, 11};
8     printf("%i\n", tab==&tab[0]);
9     printf("%p\n", tab);
10    printf("%p\n", &tab);
11    printf("%p\n", &tab[0]);
12    printf("%d\n", *tab);
13    printf("%d\n", tab[0]);
14    printf("%i\n", *tab==tab[0]);
15    printf("%p\n", tab+1);
16    printf("%p\n", &tab[1]);
17    printf("%d\n", *(tab+1));
18    printf("%d\n", tab[1]);
19    printf("%d\n", *tab+1);
```

Przekazywanie tablic do funkcji

Równoważnie

```
1 int suma(int *tab, int n)
2 {
3 // tutaj kod
4 }
```

lub

```
1 int suma(int tab[], int n)
2 {
3 // tutaj kod
4 }
```

Uwaga: nie możemy wprost odczytać rozmiaru(!).

Kod do analizy:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int suma(int tab[])
5  {
6      int temp=0;
7      for(int i=0;i<sizeof tab/sizeof(int);i++)
8      {
9          temp+=tab[i];
10     }
11     return temp;
12 }
13
14 int main()
15 {
16     int tablica[5]={1,-4,2,0,3};
17     printf("%d\n",suma(tablica));
18     return 0;
19 }
```

Działania na wskaźnikach

- Przypisanie - przypisywana wartość może być tablicą albo zmienną poprzedzoną operatorem adresu (&), ewentualnie innym wskaźnikiem.

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int tab[4]={1,-3,4,2};
5     int *wsk1, *wsk2;
6     wsk1=tab;
7     wsk2=&tab[2];
8 }
```

- Pobranie wartości (dereferencja, wyłuskiwanie) - *
- Pobranie adresu wskaźnika - &

- Dodawanie liczb całkowitych do wskaźnika — za pomocą operatora `+` można dodać liczbę całkowitą do wskaźnika albo wskaźnik do liczby całkowitej. Wtedy liczba całkowita jest mnożona przez liczbę bajtów zajmowaną przez wartość wskazywanego typu, a wynik jest dodawany do pierwotnej wartości adresu.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int tab[4]={1,-3,4,7};
5      int *wsk1, *wsk2;
6      wsk1=tab;
7      wsk2=wsk1+3;
8      printf("%p %p\n",wsk1,wsk2);
9      printf("%d %d\n",*wsk1,*wsk2);
10     return 0;
11 }
```

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("%Iu\n", sizeof(short));
6     short tab[4]={1,-3,4,7};
7     short *wsk1, *wsk2;
8     wsk1=tab;
9     wsk2=wsk1+3;
10    printf("%p %p\n", wsk1, wsk2);
11    printf("%d %d\n", *wsk1, *wsk2);
12    return 0;
13 }
```

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("%Iu\n", sizeof(double));
6      double tab[4]={1.2,-3.3,4.3,7.7};
7      double *wsk1, *wsk2;
8      wsk1=tab;
9      wsk2=wsk1+3;
10     printf("%p %p\n",wsk1,wsk2);
11     printf("%lf %lf\n",*wsk1,*wsk2);
12     return 0;
13 }
```


- Zwiększenie wskaźnika (inkrementacja). Można je uzyskać przez zwykłe dodawanie lub za pomocą operatora inkrementacji. Zwiększenie wskaźnika do elementu tablicy sprawia, iż wskazuje on kolejny element.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int tab[4]={1,-3,4,7};
5      int *wsk1, *wsk2;
6      wsk1=tab;
7      printf("%p %d\n",wsk1,*wsk1);
8      wsk1++;
9      printf("%p %d\n",wsk1,*wsk1);
10     return 0;
11 }
```

- Odejmowanie liczby całkowitej od wskaźnika — w tym celu można użyć operatora `-`; wskaźnik musi być pierwszym z operandów. Liczba całkowita jest mnożona przez liczbę bajtów zajmowaną przez wartość wskazywanego typu, a wynik jest odejmowany od pierwotnego adresu.
- Zmniejszenie (dekrementacja) wskaźników - analogicznie.

- Odejmowanie. Możliwe jest znalezienie różnicy między dwoma wskaźnikami. Działanie to jest wykonywane zwykle na wskaźnikach do elementów tej samej tablicy — w celu określenia, jak daleko od siebie się znajdują. Wynik jest wyrażony w jednostce o rozmiarze typu.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int tab[4]={1,-3,4,7};
5      int *wsk1, *wsk2;
6      wsk1=tab;
7      wsk2=&tab[3];
8      printf("%d\n",wsk2-wsk1);
9      return 0;
10 }
```

- Porównanie — możemy używać operatorów relacyjnych, aby porównać wartości dwóch wskaźników tego samego typu.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int tab[4]={1,-3,4,7};
5      int *wsk1, *wsk2;
6      wsk1=tab;
7      wsk2=&tab[3];
8      printf("%d\n",wsk2>wsk1);
9      return 0;
10 }
```

A czy jest dodawanie wskaźników?

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int tab[4]={1,-3,4,7};
5      int *wsk1, *wsk2;
6      wsk1=tab;
7      wsk2=&tab[3];
8      //printf("%d\n",wsk2+wsk1);
9      return 0;
10 }
```

Bibliografia

- Richard Reese, Wskaźniki w języku C, Wydawnictwo Helion 2014.
- <https://pl.wikibooks.org/wiki/C/Wska%C5%BAniki>, dostęp online 15.03.2020.
- http://wazniak.mimuw.edu.pl/index.php?title=Wst%C4%99p_do_programowania_w_j%C4%99zyku_C/Wska%C5%BAniki, dostęp online 15.03.2020.
- https://pl.wikibooks.org/wiki/C/Wska%C5%BAniki_-_wi%C4%99cej, dostęp online 15.03.2020.
- Stephen Prata, Język C. Szkoła programowania. Wydanie VI, Wyd. Helion, 2016.
- <https://pl.wikibooks.org/wiki/C/Tablice>, dostęp online 20.03.2020.
- https://pl.wikibooks.org/wiki/C/Tablice_-_wi%C4%99cej, dostęp online 20.03.2020.

