



## Wskaźniki \*

- Wskaźniki są to obiekty, które **wskazują** na inne obiekty danych.
- Wskaźnik to **adres** obiektu w pamięci.
- Wskaźników można używać w powiązaniu z obiektami dowolnego typu

*Przykład:*

**deklaracja wskaźnika**

char a;

char \*w; // w jest wskaźnikiem do obiektu typu char

w = &a; // teraz w wskazuje na a

// \*w jest obiektem pod adresem w i ma wartość a



## Wskaźniki i tablice

```
int x, a[5], int *p;
```

```
p = a; // lub: p = &a[0];
```

```
x = *p; // tzn. x = a[0];
```

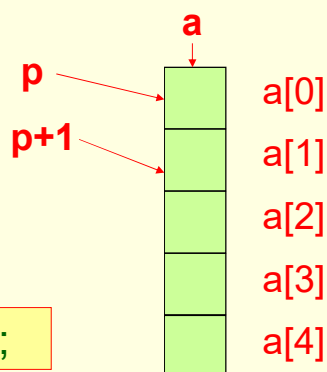
```
x = *(p+1); // tzn. x = a[1];
```

```
p++; x = *p; // tzn. x = a[1];
```

```
x = *(p+i); // tzn. x = a[i];
```

```
x = *(a+i); // tzn. x = a[i];
```

```
p += i; x = *(p); // tzn. x = a[i];
```





## Sposoby ustawiania wskaźników

Wskaźnik można ustawić poprzez:

- wstawienie do niego adresu obiektu  
`wsk = & obiekt;`
- przypisanie mu innego wskaźnika  
`wsk = inny_wskaznik;`
- przypisanie mu adresu tablicy  
`wsk = tablica;`
- ustawienie go na początku stringu  
`wsk = "napis";`

216

## Przykład 1

Napisz program, który wyznacza długość łańcucha wpisanego z klawiatury:

- a) przy użyciu tablicy i indeksów,
- b) przy użyciu tablicy i wskaźników.

217



```
int dluglanc (char lancuch [ ])  
{  
  int i;  
  for ( i = 0; lancuch [i] != '\0'; i++)  
    ; // instrukcja pusta  
  return i;  
}  
/*****/
```

### *Wyznaczenie długości łańcucha za pomocą wskaźników*

```
// deklaracja zmiennych  
char lancuch [50];  
char *cp, *lp;  
// zainicjowanie wskaźników  
cp = lp = lancuch;  
// określenie długości łańcucha znaków  
while (*lp)  
  lp ++;  
return lp - cp;
```



```
int dluglanc (char *cp)
{
  char *lp = cp;
  while (*lp)
    lp ++;
  return lp - cp;
}
/*****/
```

## Przykład 2

Napisz program, który sortuje niemalejąco dany ciąg liczb  $x_1, x_2, \dots, x_n$  ( $n < 201$ ) o wartościach całkowitych.

W programie zastosuj tablicę i wskaźniki oraz wybór metody sortowania bąbelkowego lub szybkiego.

## Program 4

```
// plik nagłówkowy tenprog.h  
/* program sortowania - ze wskaźnikami*/  
  
int sort_b (int, int *); //prototypy funkcji  
int sort_s (int, int, int *);  
int wybor (int, int *);  
void druk (int, int *);  
void wpisz (int, int *);
```

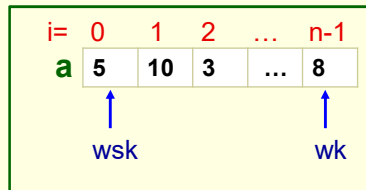
226

```
// plik źródłowy sortowanie.c  
# include <stdio.h>  
# include <stdlib.h>  
# include "tenprog.h,,  
  
/*****/  
int sort_b (int n, int *a) //sortowanie babelkowe  
{  
    int p,aa,l=0;  
    int *wsk, *wk;  
    wk=&a[n-1];
```

```

do {
  p=0;
  for (wsk=a; wsk<wk; wsk++) {
    if (*wsk>*(wsk+1))
    {
      aa = *wsk; //zamiana
      *wsk = *(wsk+1);
      *(wsk+1) = aa;
      p=1; l++;
    }
  } while (p!=0);
  return l;
}

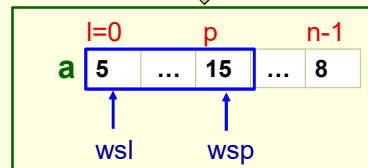
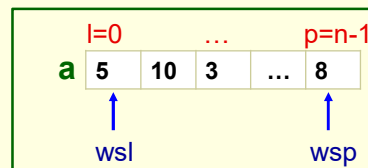
```



```

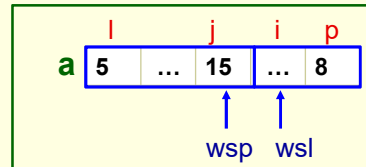
/*****/
int sort_s (int l, int p, int *a) //sortowanie szybkie
{
  static int lz=0;
  int i, j, aa;
  int *wsl,*wsp;
  int w=(a+(l+p)/2);
  wsl=a+l; wsp=a+p;

```





```
do {while (*wsl<w) wsl++;
    while (w<*wsp) wsp--;
    if (wsl<=wsp) {
        aa=*wsl; //zamiana a[i] z a[j]
        *wsl=*wsp;
        *wsp = aa;
        wsl++; wsp--; lz++;
    }
} while (wsl<=wsp);
j= wsp-a; i= wsl-a;
if (l<j) sort_s (l, j, a);
if (i<p) sort_s (i, p, a);
return lz;
}
```



```
int wybor (int n, int *a)
/* wybor metody sortowania tablic */
{
    int p, l;
    char c;

    do {
        fpurge(stdin); //stdio.h - czyszczenie bufora
        p=0;
        printf("Wybierz metode sortowania:
                b - babelkowe, s - szybkie\n");
        c = getchar();
    }
```

```
switch (c)
{
    case 'b': l = sort_b (n,a);
                break;
    case 's': l = sort_s (0,n-1,a);
                break;
    default: printf ("Bład! "); p=1;
                break;
}
} while (p);
return l;
}
```

```
// plik źródłowy program.c
#include <stdio.h>
#include <stdlib.h>
#include "tenprog.h"
/*****/
int main (void)
{
    ...
}
/*****/
void wpisz(int n, int *w)
{
    ...
}
```

```
if (c=='k') {
    printf("Wpisz kolejne elementy ciagu \n\n");
    for (j=0; j<n; j++) {
        printf("a[%d] = ", j+1);
        scanf("%d", w++);
    }
}
else {
    printf(„\nWpisz ziarno ");
    scanf("%d", &seed); printf("\n\n");
    srand(seed); //stdlib.h
    for (j=0; j<n; j++) *(w++) = rand()%100; //stdlib.h
}
}
```

```
/******  
void druk (int k, int *w)  
{  
    int i;  
    for (i=0; i<k; i++) {  
        printf(" %6d", *(w++));  
        if ((i+1)%10==0) printf("\n");  
    }  
}
```