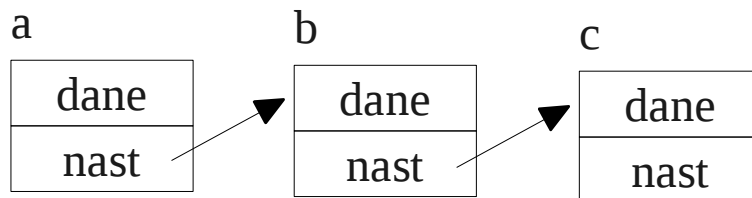


```

struct element {
    int dane;
    struct element *nast;
};
typedef struct element ELEMENT;
typedef ELEMENT *LISTA;

```

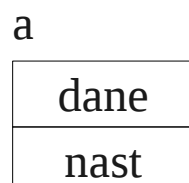


```

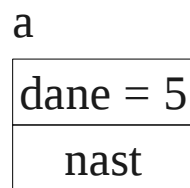
ELEMENT a, b, c;
a.dane = 1;
a.dane = 2;
a.dane = 3;
a.nast = &b;
b.nast = &c;
c.nast = NULL;

```

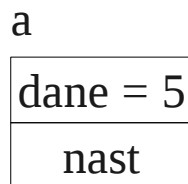
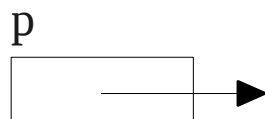
<pre> struct element { int dane; struct element *nast; }; struct element a; </pre>	<pre> int int a; </pre>
--	-------------------------



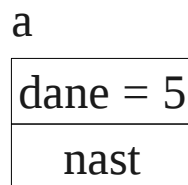
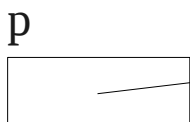
```
a.dane = 5;
```



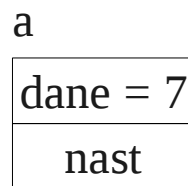
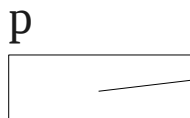
```
struct element *p;
```



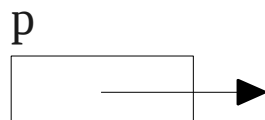
```
p = &a;
```



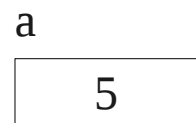
```
p->dane = 7;  
(*p).dane = 7;
```



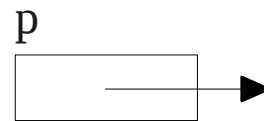
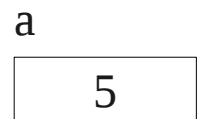
```
ELEMENT *p;
```



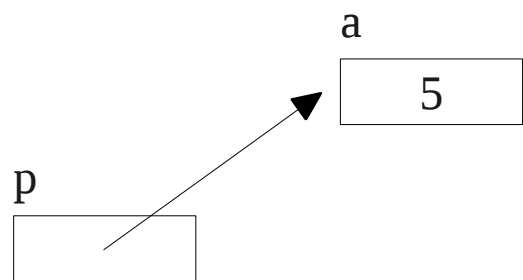
```
a = 5;
```



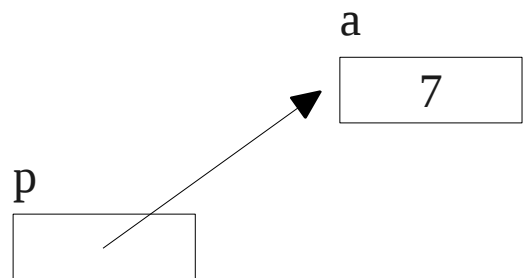
```
int *p;
```



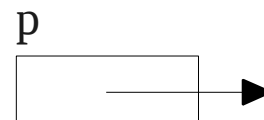
```
p = &a;
```



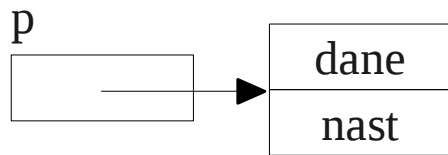
```
*p = 7;
```



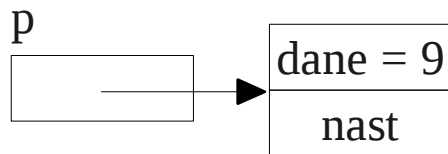
```
int *p;
```



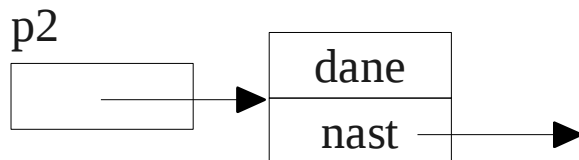
```
p = (ELEMENT*)
malloc(sizeof(ELEMENT));
```



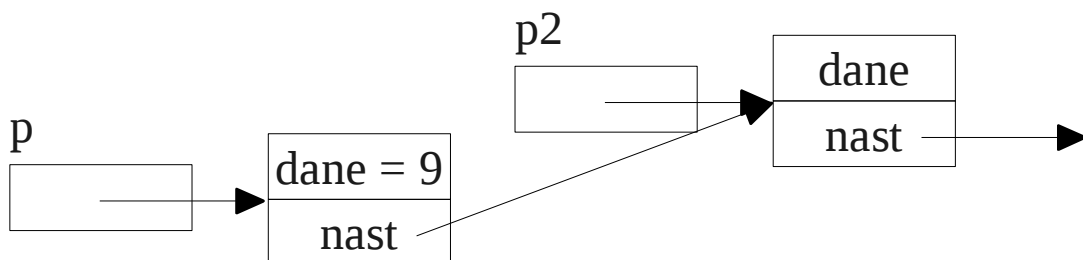
```
p->dane = 9;
```



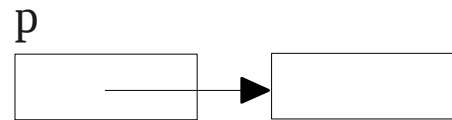
```
ELEMENT *p2 = (ELEMENT*)
malloc(sizeof(ELEMENT));
```



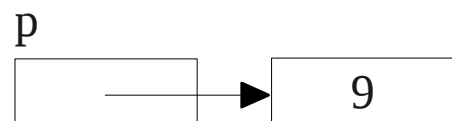
```
p->nast = p2;
```



```
p = (int*)
malloc( sizeof(int)
);
```

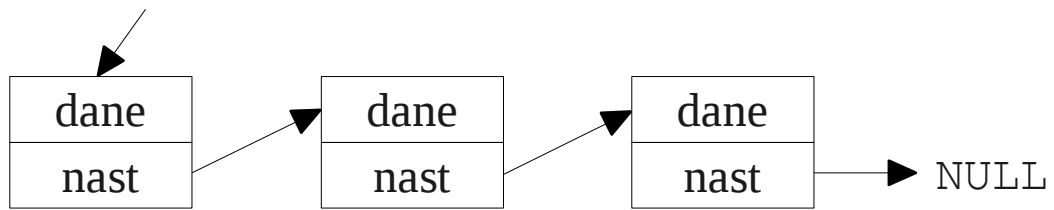


```
*p = 9;
```

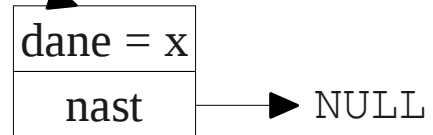


```
free(p);
```

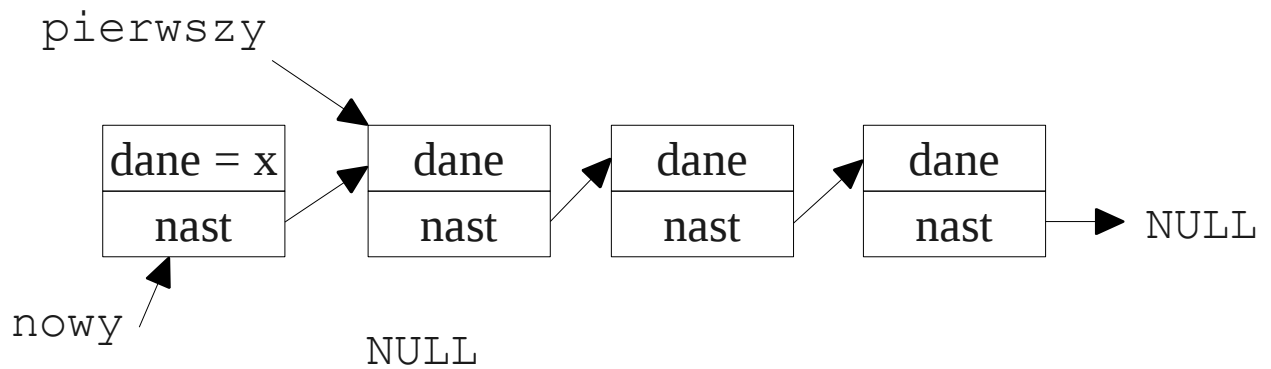
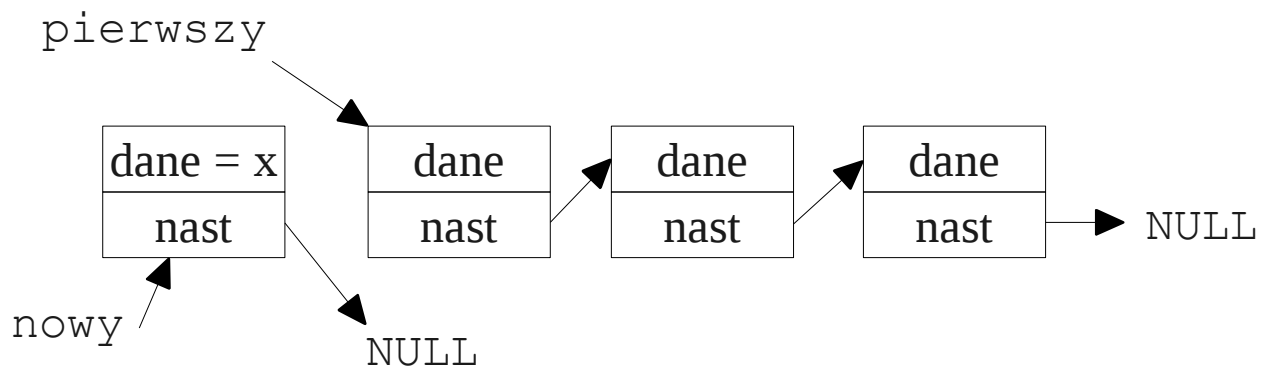
ELEMENT* poczatek;



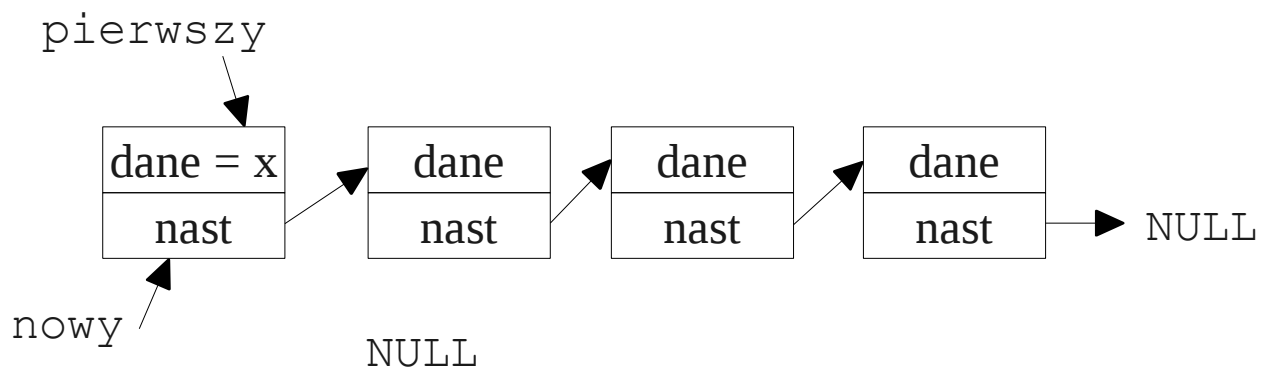
```
void dodaj(int x) {  
    ELEMENT *nowy;  
    nowy = (ELEMENT*)malloc(sizeof(ELEMENT));  
    nowy->dane = x;  
    nowy->nast = NULL;  
    ...  
    ...  
    ...  
}
```



przypadek pierwszy: na początek listy



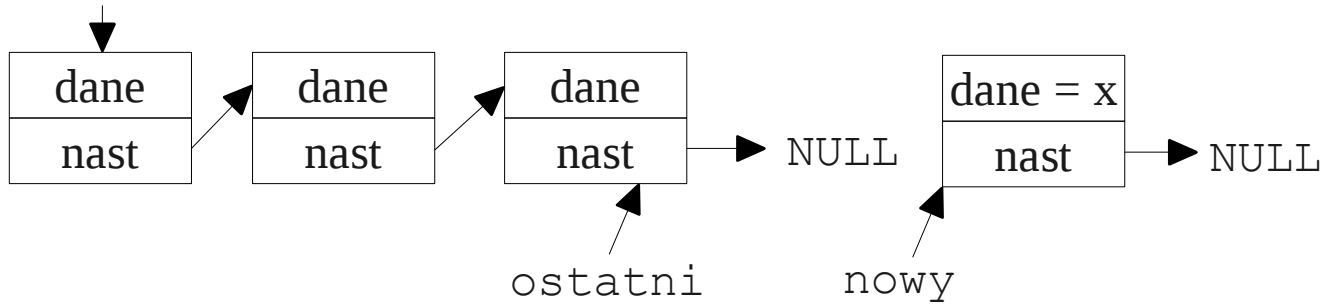
nowy->nast = pierwszy;



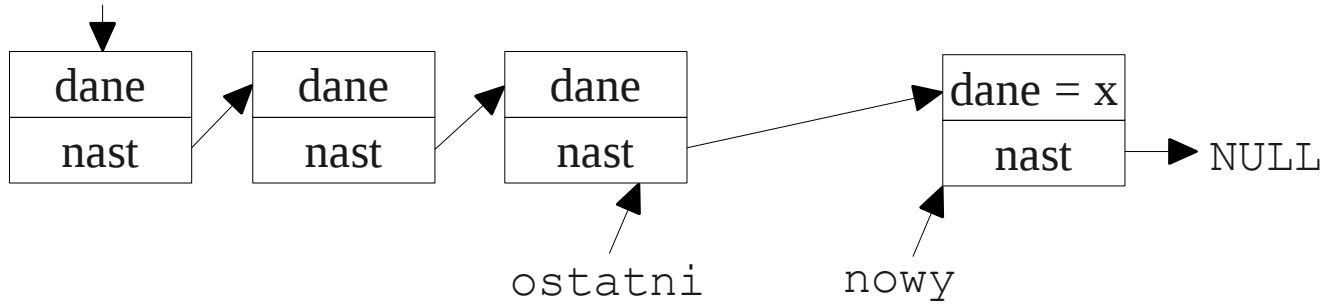
pierwszy = nowy;

przypadek drugi: na koniec listy

pierwszy

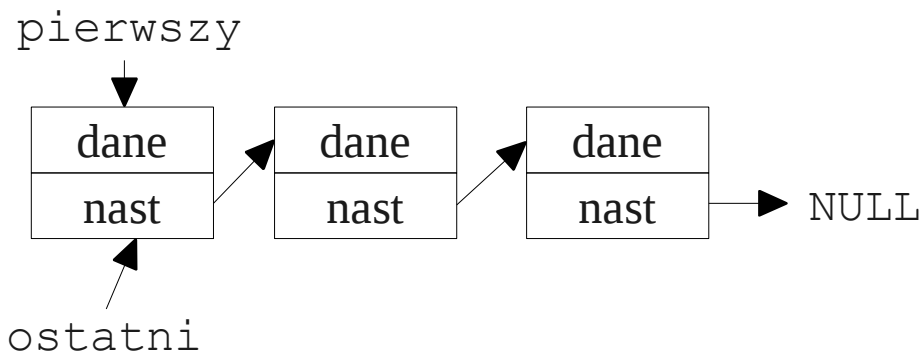


pierwszy

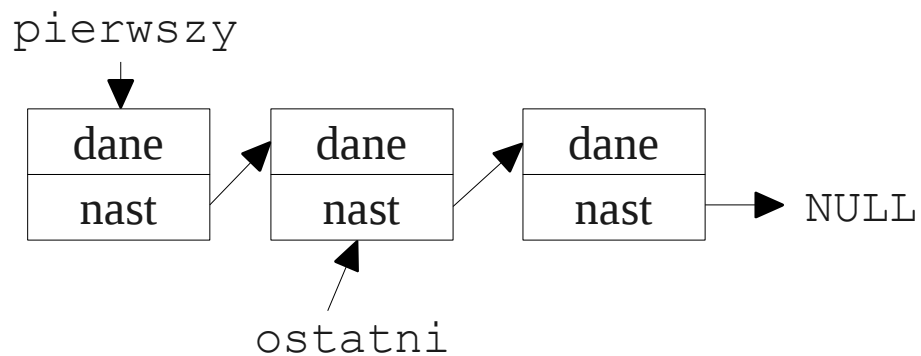


ostatni->nast = nowy;

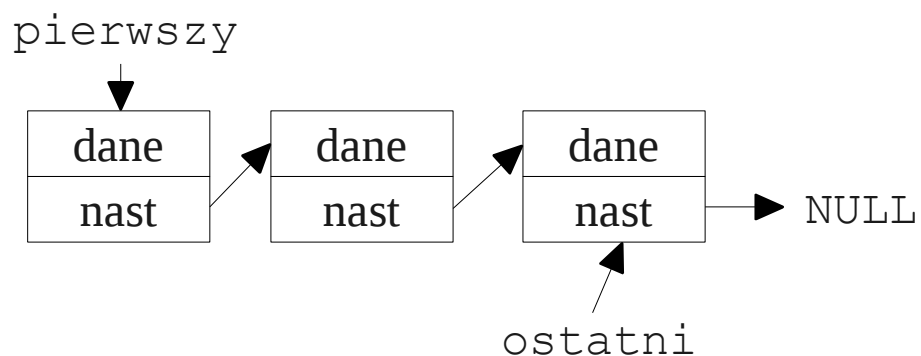
a jak znaleźć ostatni?



ostatni = pierwszy;

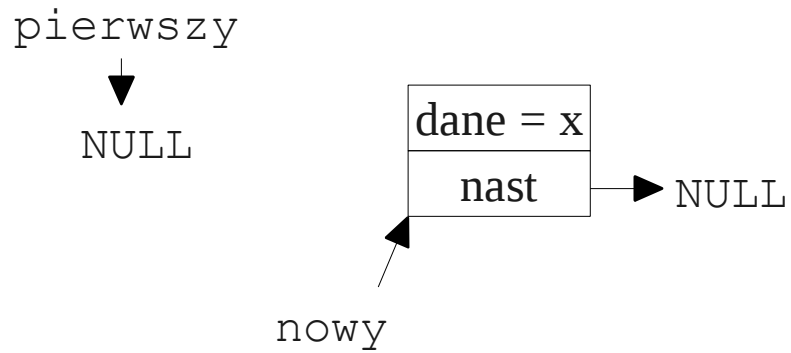


```
while(...) {  
    ostatni = ostatni->nast;  
}
```

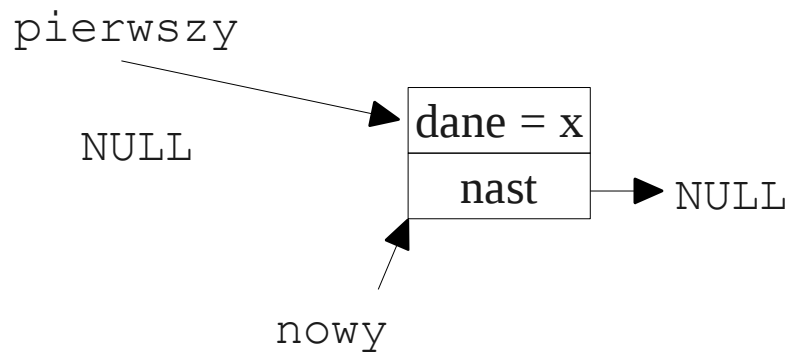


```
while(ostatni->nast != NULL) {  
    ostatni = ostatni->nast;  
}
```

przypadek specjalny: lista pusta

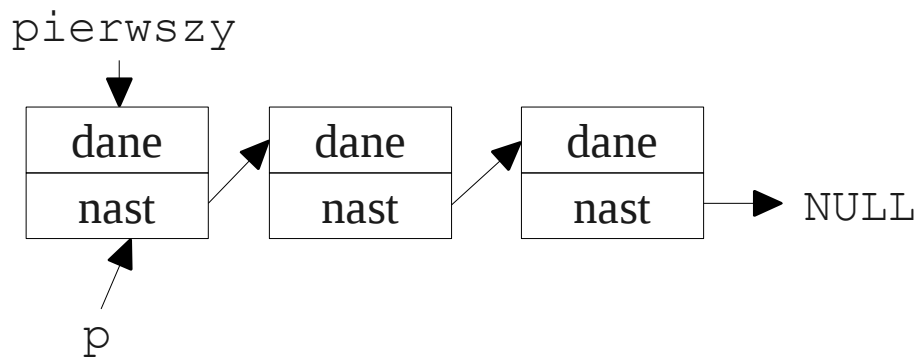


```
if(pierwszy == NULL) {  
    ...  
}
```

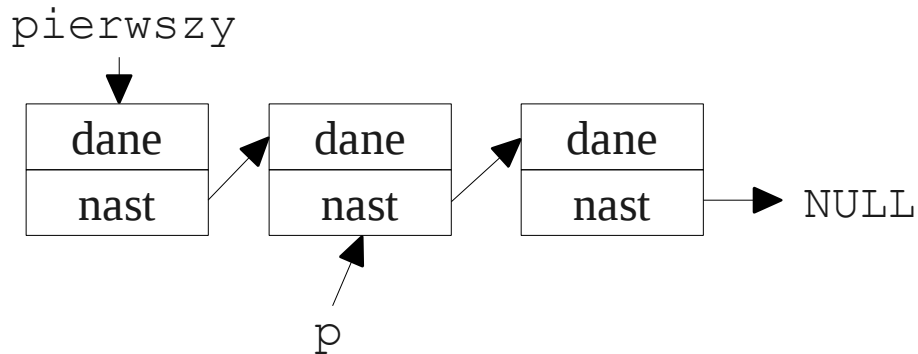


```
if(pierwszy == NULL) {  
    pierwszy = nowy;  
}
```

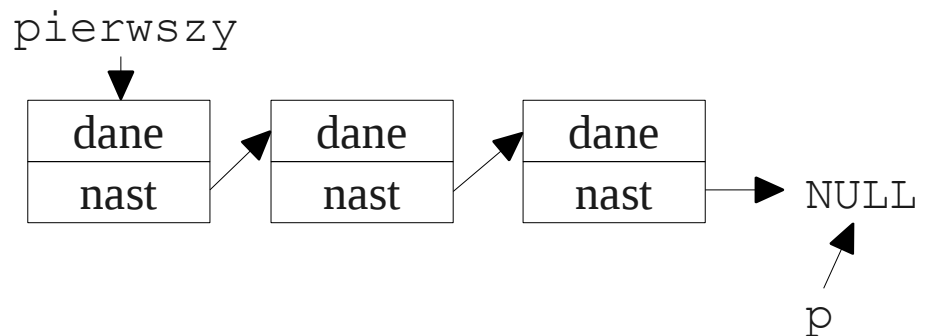
przeglądanie listy



```
ELEMENT* p = pierwszy;  
while(...) {  
    printf("%d\n", p->dane);  
}
```



```
while(...) {  
    printf("%d\n", p->dane);  
    p = p->nast;  
}
```



```
while(p != NULL) {  
    printf("%d\n", p->dane);  
    p = p->nast;  
}
```