

Listy jednokierunkowe – jest to pewna złożona konstrukcja, która w sposób elastyczny pozwala nam trzymać elementy określonego typu. W odróżnieniu od tablic nie określamy z góry jego rozmiaru, więc z punktu widzenia programisty konstrukcja jest bardziej bezpieczna.

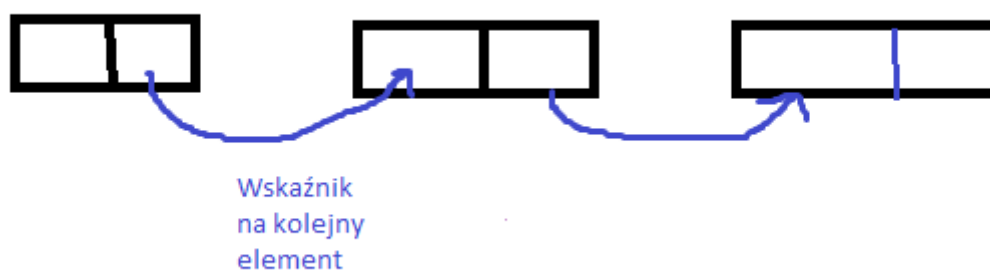
Idea implementacji w języku C polega na stworzeniu struktury elementy o składni:

```
struct element
{
    int i;
    struct element * next;
};
```

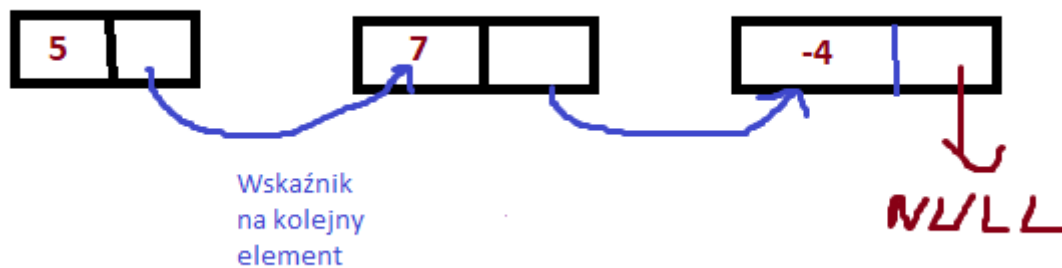
Pole i to wartość konkretnego elementu. Pole next to wskaźnik na następnik. Przy tak elastycznej konstrukcji nie mamy zawsze pewności, w której „miejsce” w pamięci trafi kolejny element. Standardowo też powinniśmy na każdy element zarezerwować pamięć (poprzez funkcję malloc). Usuwać element możemy to wykonać zwalniając pamięć metodą free.

Mamy podstawowe dwa rodzaje list: bez głowy i z głową. W przypadku listy z głową tworzymy pusty element „głowę” tak, aby wykonując operację na liście zawsze mieć stały „adres”/wskaźnik na początek listy (tak jak w przypadku tablic wskaźnik na listę to inaczej wskaźnik na pierwszy element). Co zyskujemy? Warto rozważyć sobie sytuację, kiedy mamy jakąś listę i chcemy dodać element na początek.

Rozważmy listę:



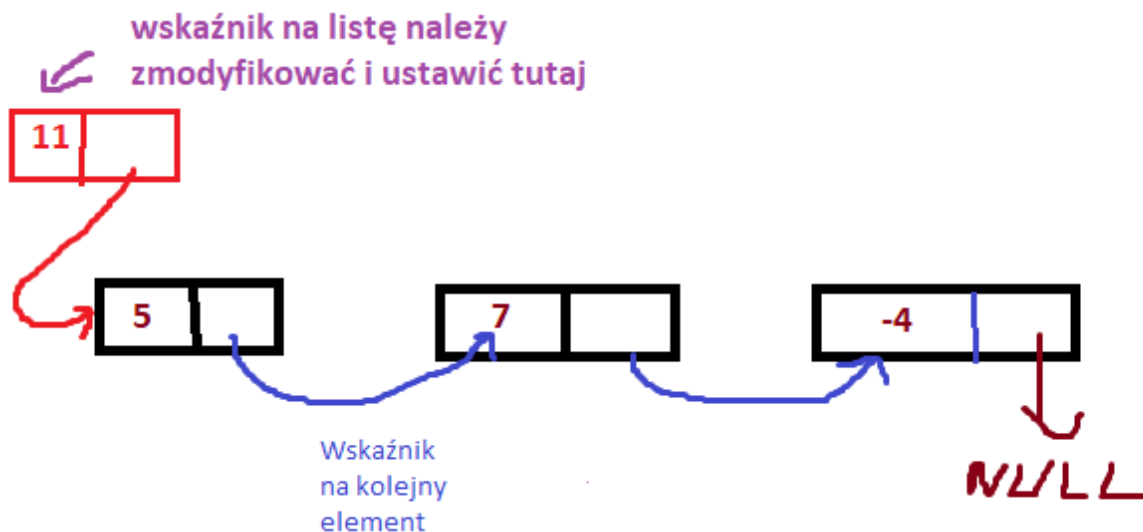
Lista bez głowy mogłaby wyglądać tak:



Dodanie na początek wymaga zatem wykonania następujących operacji (bez uwzględnienia sytuacji kiedy bazowa lista jest pusta):

- 1) Rezerwacja pamięci na nowy element
- 2) Ustawienia wartości pola i na nowym elemencie z punktu 1
- 3) Pole next nowego elementu z punktu 1 jest ustawiane jako adres pierwszej elementu początkowej listy
- 4) Należy zmienić wskaźnik całej „listy” wskazując jako adres „nowy” element z punktu 1.

Obrazowo:



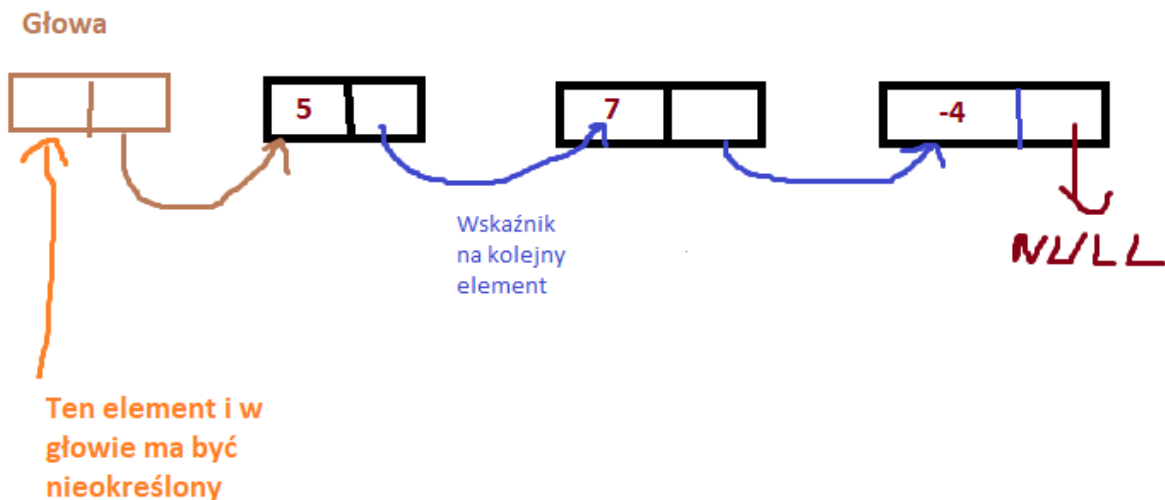
Kod był realizowany w jednym z zadań

[https://github.com/pjastr/ProgramowanieStrukturalne/blob/master/7 3 3/main.c](https://github.com/pjastr/ProgramowanieStrukturalne/blob/master/7%203%203/main.c)

Ew można to zrobić bez funkcji wykonując kolejne instrukcje w main:

<https://gist.github.com/pjastr/c4623127a08d172c3304af4ae34093eb>

Rozważmy teraz przypadek listy z głową

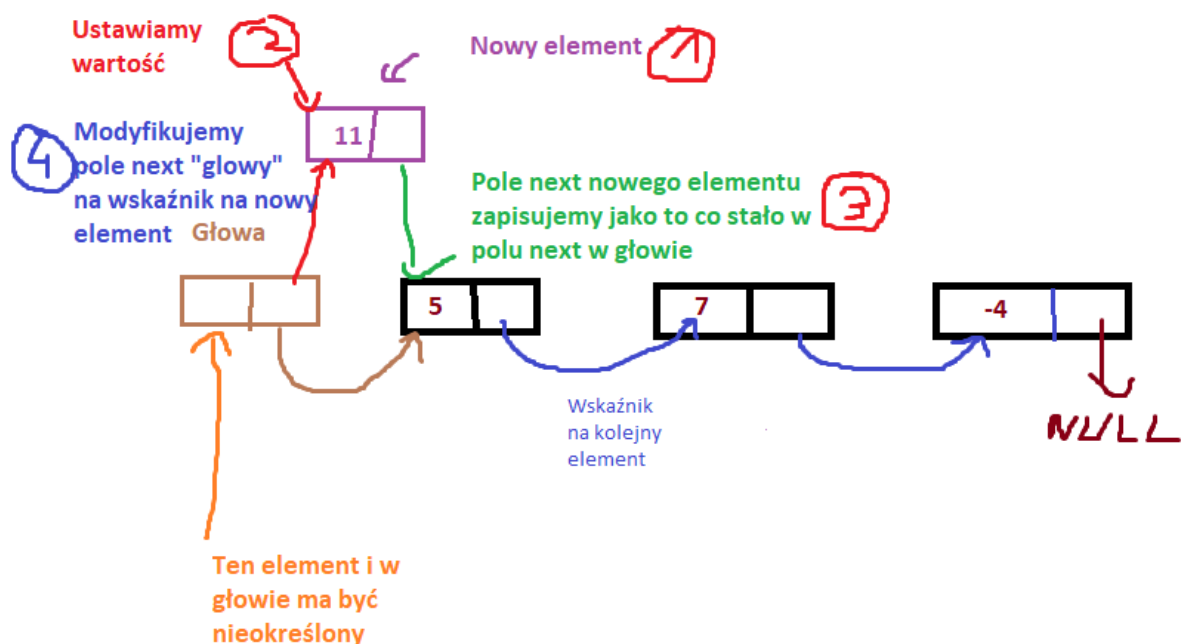


Dodanie na początek (przy założeniu że bieżąca lista nie jest pusta) polega na wykonaniu operacji:

- 1) Rezerwujemy pamięć na nowy element
- 2) Ustawiamy wartość jako pole i
- 3) Pole next ustawiamy jako to co znajduje się w „głowie” w polu next
- 4) Modyfikujemy pole next w „głowie” ustawiając je jako wskaźnik na nowy element z punktu 1

Ważne: nie możemy zmienić kolejności punktu 3 i 4 bo zmieni to sens.

Obrazowo:



Kod z użyciem funkcji był w jednym z zadań

https://github.com/pjastr/ProgramowanieStrukturalne/blob/master/7_3_12/main.c

Bez funkcji, wykonując jedynie instrukcje w main:

<https://gist.github.com/pjastr/239641b4286caa767cccbea8e860e6bf>

Co zyskujemy? To zależy od kontekstu w którym używamy i jakie operacje mamy. W wielu sytuacjach zmiana wskaźnika „początku” przy liście bez głowy jest operacją, która zwiększa tzw. złożoność obliczeniową (https://pl.wikipedia.org/wiki/Z%C5%82o%C5%BCono%C5%9B%C4%87_obliczeniowa).

Dokładniejsze szczegóły będą umówione na Algorytmach i Strukturach Danych. Na dziś można sobie myśleć, że jest to „szybsze” (to też zależy od języka programowania).