ADT LIST

Lista – ciąg 0 lub więcej elementowy, złożony z elementów tego samego typu:

$$a_1, a_2, \ldots, a_n \quad n>=0;$$

- *n* - długość listy (length)
- a_1 – pierwszy element listy (first)
- a_n – ostatni element listy (last)
- Jeśli *n*=0 to lista jest pusta (empty)

ADT LIST

$$a_1, a_2, \ldots, a_n \quad n>=0;$$

- Mówimy, że a_i występuje na pozycji i
- Mówimy, że a_i poprzedza a_{i+1}
- Mówimy, że a_i następuje po a_{i-1}
- Przyjmuje się istnienie elementu za ostatnim elementem listy. Funkcja **END(l)** zwraca tą pozycję

ADT LIST

Operacje na liście:
(ogólnie zdefiniowane, pomijające typ listy)

- **INSERT(x,p,L)** – wstaw *x* na pozycję *p* na liście *L* Jeśli *p* jest pozycją END(L), to lista a_1 ,a_2,...,a_n staje się listą a_1 , a_2 ,...,a_n ,x. Jeśli w *L* nie ma pozycji *p*, to operacja daje rezultat nieokreślony
- **LOCATE(x,L)** zwraca pozycję pierwszego wystąpienia elementu *x* w liście *L*. Jeśli x nie występuje w liście to zwraca END(L)
- **RETRIEVE(p,L)** – zwraca element występujący w *L* na pozycji *p*.

ADT LIST

Operacje na liście:

- **DELETE(p,L)** – usuwa element na pozycji *p* z listy. Gdy w *L* nie ma pozycji *p* lub *p*=END(L) to wynik jest nieokreślony.
- **NEXT(p,L)** – zwraca pozycję następną w stosunku do *p* w *L*. Jeśli *p* jest pozycją ostatniego elementu w liście *L* to next(p,L)=END(L). NEXT(p,L) jest nieokreślony gdy *p*=END(L).
- **PREVIOUS(p,L) -** zwraca pozycję poprzednią w stosunku do *p* w *L*. Gdy *p*=FIRST(L) to PREVIOUS(p,L) jest nieokreślone.

(gdy nie ma elementów w liście to Next i Previous niekreślone)

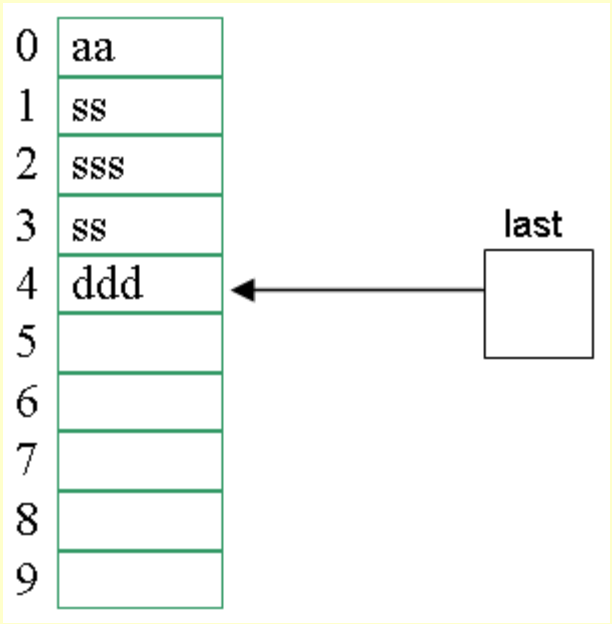
ADT LIST

Operacje na liście:

- **MAKENULL(L).**Czyni listę pustą i zwraca pozycję END(L).
- **FIRST(L)** – zwraca pozycję pierwszego elementu w L.Jeśli L pusta, to zwraca END(L)
- **PRINT_LIST(L)** – wypisuje elementy w kolejności występowania

ADT LIST

Tablicowa implementacja listy:



Last – indeks ostatniego elementu listy

ADT LIST - reprezentacja tablicowa listy

Tablicowa implementacja listy:

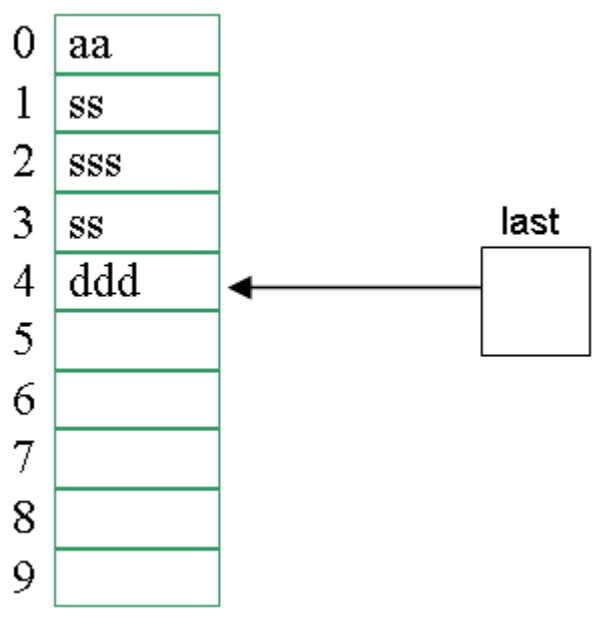
```
const int maxlength=10;
struct list
{
    elementtype elements[maxlength];
    int last;          //indeks ostatniego elementu listy
}
int position; // indeks danego elementu listy, pozycja elementu
```

ADT LIST - reprezentacja tablicowa listy

pozycja elementu- indeks elementu w tablicy (numer komórki w której się znajduje)

First() - zwraca pozycję elementu pierwszego -(indeks 0)

END() - zwraca pozycję za ostatnim elementem (czyli last+1)



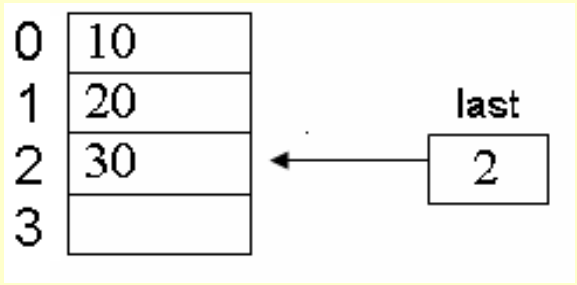
ADT LIST - reprezentacja tablicowa listy

Tablicowa implementacja listy:

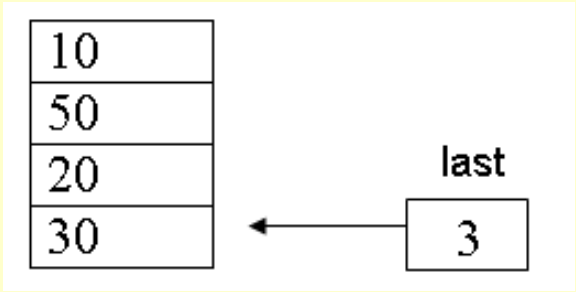
Lista pusta: last=-1

ADT LIST - reprezentacja tablicowa listy

Przykład działania operacji Insert w tej reprezentacji.
Załóżmy, mamy tablicę czteroelementową (maxlength=4) a nasza lista składa się z elementów 10,20,30



Wywołujemy procedurę Insert(50,1,l)- chcemy wstawić element 50 na pozycję 1. (do komórki tablicy o indeksie 1)

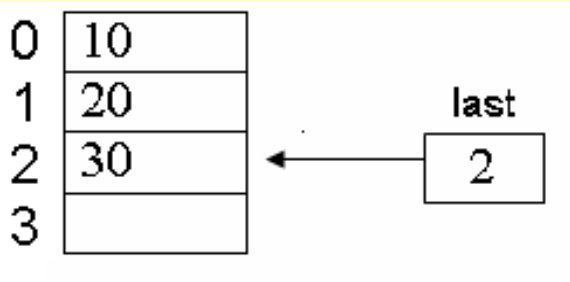


Operacja Insert przesunęła element 30 z pozycji 2 na pozycję 3
oraz element 20 z pozycji 1 na pozycję 2, aby zrobić miejsce na element 50 na pozycji 1.

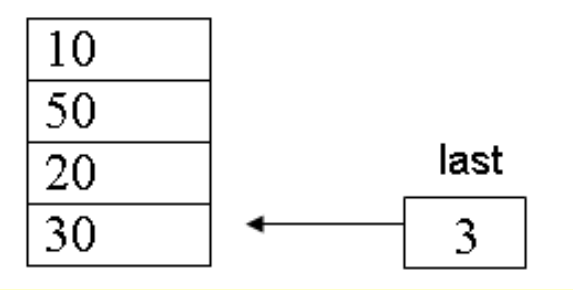
ADT LIST - reprezentacja tablicowa listy

Insert(x,p) - wstawia x do komórki o numerze p (o ile się da):

-sprawdź, czy jest miejsce w tablicy
-sprawdź, czy pozycja p jest poprawna
-przesuń elementy w tablicy, aby komórka p była pusta
-wstaw x do komórki p
-zwiększ last
-zwróć true gdy operacja wstawiania się powiedzie, false wpp



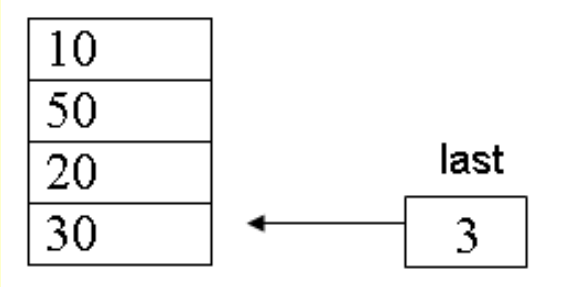
Insert(50,1,l)



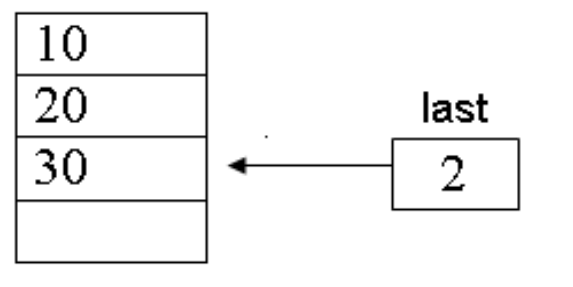
ADT LIST - reprezentacja tablicowa listy

Przykład działania operacji Delete :

Załóżmy, mamy tablicę czteroelementową (maxlength=4) a nasza lista składa się z elementów 10,50,20,30



Wywołujemy procedurę Delete(1,l)- chcemy usunąć element z pozycji 1.
Lista wygląda następująco:

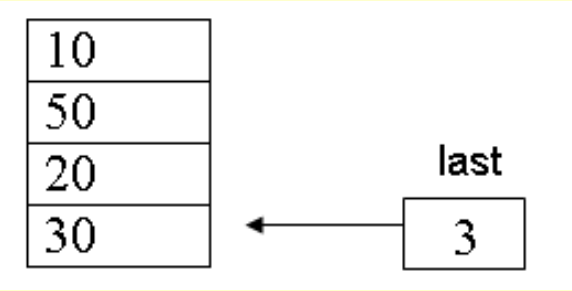


Operacja Delete przesunęła element 20 z pozycji 2 na pozycję 1.
oraz element 30 z pozycji 3 na pozycję 2

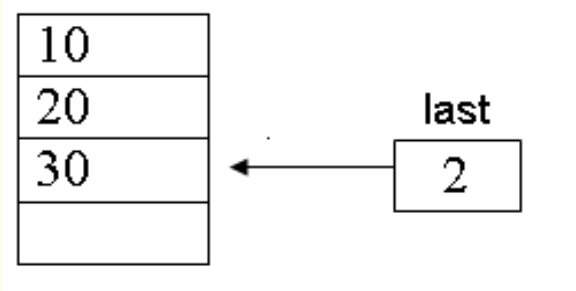
ADT LIST - reprezentacja tablicowa listy

Delete(p) - usuwa element z komórki o numerze p:

-sprawdź, czy pozycja p jest poprawna
-przesuń elementy w "górę" tablicy
-zmniejsz last
-zwróć true gdy operacja usuwania się powiedzie, false wpp



Delete(1,l)



ADT LIST - reprezentacja tablicowa listy

Next(p) -zwraca indeks następnego elementu po p (czyli p+1):

-jeśli istnieje element następny, to zwróć jego indeks
-jeśli nie istnieje element następny, to zwróć -1

Previous(p) - zwraca indeks poprzedniego elementu w stosunku do p (czyli p-1)
-jeśli istnieje element poprzedni, to zwróć jego indeks
-jeśli nie istnieje element poprzedni, to zwróć -1

ADT LIST - reprezentacja tablicowa listy

Locate(x) - zwraca pozycję elementu x w liście (indeks komórki), jeśli x występuje w tablicy.
Zwraca END() (pozycję za ostatnim elementem) gdy x nie występuje w tablicy

Retrieve(p) - zwraca element znajdujący się w liście na pozycji p (w komórce p), jeśli pozycja p jest poprawna
Zwraca MIN wpp, gdzie MIN to wartość która na pewno nie wystąpi w naszej liście. Gdy przechowujemy w liście liczby integer np. MIN=-10000000

ADT LIST - reprezentacja tablicowa listy

Zadanie:
Zaimplementuj podstawowe operacje na liście
w reprezentacji tablicowej (plik ListaTablicowa.cpp)
Przetestuj korzystając z testu w pliku ListaTest.cpp