Aplicatie la liste inlantuite

Reprezentarea polinoamelor si operatii cu polinoame

Reprezentarea polinoamelor

$$P(X) = 3X^2 + 2X + 1$$

$$Q(X) = 100X^{2010} + X^{1000} + X^2 + 5$$

Cum sa le reprezentam daca de exemplu dorim sa scriem un program care aduna doua polinoame?

Reprezentarea polinoamelor

- Fiecare polinom se poate scrie ca o suma de termeni.
- Fiecare termen este bine definit prin coeficient si gradul termenului

$P(X) = 3X^2 + 2X + 1$

P:

3	2	2	1	1	0

$Q(X) = 100X^{2015} + X^{1000} + X^2 + 5$

Q:

100	2015	0	2014	0	2013	...

1	1000	0	999	...	0	3

1	2	0	1	5	0

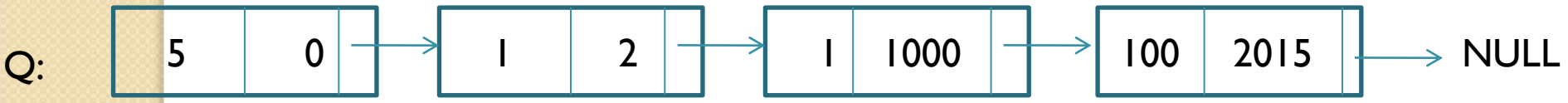
Reprezentarea polinoamelor

- Fiecare termen nenul se va reprezenta ca un nod cu componentele:
  - Coef
  - Grad
  - Link (legatura la urmatorul termen nenul)
- Numai termenii nenuli vor fi reprezentati.
- Termenii se vor aseza in lista in ordinea crescatoare a gradelor.

Reprezentarea polinoamelor

$$P(X) = 3X^2 + 2X + 1$$

$$Q(X) = 100X^{2015} + X^{1000} + X^2 + 5$$



Adunarea polinoamelor

$$P(X) = X + X^3 - X^5$$

$$Q(X) = 8 + 2X^2 + X^3 + X^4 + X^5$$

$$(P+Q)(X) = 8 + X + 2X^2 + 2X^3 + X^4$$

Adunarea polinoamelor

$$P(X) = a_0 + a_1 X + a_2 X^2 + \ldots + a_n X^n \qquad a_n \neq 0$$

$$Q(X) = b_0 + b_1 X + b_2 X^2 + \ldots + b_m X^m \qquad b_m \neq 0$$

$$(P+Q)(X) = c_0 + c_1 X + \ldots + c_p X^p$$

$$c_i = \begin{cases} a_i + b_i & \text{daca } 0 \leq i \leq \min(m,n) \\ a_i & \text{daca } n > m \text{ si } m < i \leq n \\ b_i & \text{daca } m > n \text{ si } n < i \leq m \end{cases}$$

Algoritmul de adunare
P, Q date => P = P + Q

- Plecand de la lista polinomului P, daca va fi necesar,
  - vom adauga in lista noduri,
  - vom sterge noduri si
  - vom modifica coeficienti,

astfel incat lista polinomului P va deveni lista polinomului P+Q.

In final, lista polinomului Q va fi ștearsă.

Algoritmul de adunare
Variabile

- Folosim variabilele:

  - $HEAD_P$ pointer la lista lui P și

  - $HEAD_Q$ pointer la lista lui Q

- Folosim variabilele suplimentare:

  - $iter_P$ și $iter_Q$, pointeri ce parcurg listele celor doua polinoame

  - preced puncteaza la nodul dinaintea lui $iter_P$.

Algoritmul de adunare-Descriere

// Initializare

$iter_Q$ = $HEAD_Q$, $iter_P$ = $HEAD_P$, preced=NULL

while ($iter_Q$ ≠NULL and $iter_P$ ≠ NULL) // Atat timp cat nici una din liste nu s-a epuizat

i = $iter_P$ -> grad, j = $iter_Q$ -> grad

a= $iter_P$ -> coef, b= $iter_Q$ -> coef

if i = j then // termeni de acelasi grad

if a+b ≠ 0 then // coeficientul sumei termenilor este nenul

$iter_P$ -> coef = a + b

// avansează in lista lui P

preced= $iter_P$

$iter_P$ = $iter_P$ ->link

else // sterge din lista lui P
//termenul curent
preced ->link = $iter_P$ -> link
temp = $iter_P$
$iter_P$= $iter_P$ -> link
delete temp.

endif
$iter_Q$=$iter_Q$->link //avansează in lista Q
else if j < i then //insereaza un nou nod in
lista P inaintea nodului
curent in P ca o copie a
nodului curent în Q.
Aloca memorie pentru un nod nou.
temp = pointer la noul nod.

```
if temp = NULL then OVERFLOW
                                STOP
        endif
        if preced ≠ NULL then
                        preced ->link =temp
        else // se insereaza la inceputul
             listei lui P
             HEAD_P=temp
        endif
        preced = temp
        temp ->link = iter_P
        temp ->coef = b
        temp ->grad =j
        // avansează in lista lui Q
        iter_Q= iter_Q->link
```

```
                else // avansează in lista lui P
                        preced= iterP
                        iterP = iterP ->link
                    endif
                endif
Endwhile
// Daca lista lui Q nu s-a terminat, dar a lui P s-a terminat se adauga la lista lui P ceea ce a mai ramas din lista lui Q
while (iterQ ≠ NULL )
        Aloca memorie pentru un nod nou.
        temp = pointer la noul nod.
        if temp = NULL then OVERLOW; STOP
        endif
```

Algoritmul de adunare- ultima parte

preced ->link =temp

temp ->link = NULL

temp ->coef = $iter_Q$ -> coef

temp ->grad = $iter_Q$ -> grad

// avansează in lista lui Q

$iter_Q$= $iter_Q$->link

preced= temp

- endwhile