

**Soluția problemei Expresie**

**Autor. prof. Eugen Nodea**  
**C. N. „Tudor Vladimirescu” Tg. Jiu**

**Cerința 1:**

**nr**-numărul de numere ce se află în expresie se determină extrem de ușor acesta fiind egal cu numărul de virgule conținute de expresie **+1**  
**nr=nv+1**

**Cerința 2:**

**Soluție 1 – folosind stive (Eugen Nodea)**

**st** - stiva în care reținem **k**-șirul  
**r** - stiva în care reținem pe ce poziție se deschide o paranteză deschisă, **kr** vârful stivei  
**d** - stiva în care reținem pe ce poziție se deschide o paranteză dreaptă, **kd** vârful stivei

Algoritmul este următor:

- fie **k** vârful stivei **st**
- dacă întâlnim o paranteză rotundă deschisă, reținem poziția de început a unui **k**-șir delimitat de paranteze rotunde în stiva **r[++kr]=k+1**
- dacă întâlnim o paranteză dreaptă deschisă, reținem poziția de început a unui **k**-șir delimitat de paranteze drepte în stiva **d[++kd]=k+1**
- dacă întâlnim numere le adăugăm în stivă (**st[++k]=x**)
- atunci când întâlnim o paranteză rotundă închisă, vom determina secvența de sumă maximă pentru **k**-șirul de valori cuprins între indicii **r[kr]**, ..., **k**, și actualizăm stivele  
Algoritmul de determinare a secvenței de sumă maximă este clasic:
- atunci când întâlnim o paranteză dreaptă închisă, vom determina mediana **k**-șirului de valori cuprins între indicii **d[kd]**, ..., **k**, și actualizăm stivele  
**x=d[kd]** ;  
**y=quickselect(st,x,k,(k+x)/2)** ;  
**k=x; --kd; st[k]=x;**
- Pentru aflarea valorii expresiei se va face suma elementelor din stivă

**quickselect()** - algoritmul de determinare a mediane unui vector (**k**-șir) este asemănător alg. de sortare rapidă **quicksort**. Algoritmul alege ca „pivot” o anumită poziție, în cazul nostru poziția mediană  $[(k+1)/2]$ , și încearcă să aducă pe aceea poziție valoarea din șirul ordonat. După execuția subprogramului elementele aflate la stânga medianei sunt mai mici, dar nu obligatoriu ordonate, iar elementele aflate în dreapta medianei sunt mai mari.

Complexitatea algoritmului de determinare a secvenței de sumă maximă este **O(kmax)**

Complexitatea algoritmului de determinare a mediane este **O(kmax)**.

Există un algoritm mai performant decât **quickselect()**, algoritmul **binmedian()**.

**Referințe:**

[http://en.wikipedia.org/wiki/Selection\\_algorithm](http://en.wikipedia.org/wiki/Selection_algorithm)

## Soluția 2 (prof. Dan Pracsu)

Descompunem expresia în atomi, prin atomi înțelegând:

- număr
- paranteză rotundă deschisă
- paranteză rotundă închisă
- paranteză pătrată deschisă
- paranteză pătrată închisă

Vom utiliza o stivă de numere întregi în care se vor depune unii din atomi: numerele (care sunt cuprinse între – 99 și 99), parantezele rotunde deschise (codificate prin valoarea 1000) și parantezele pătrate deschise (codificate prin valoarea 2000).

Se parcurge expresia și se completează stiva cu atomi. La identificarea unei paranteze rotunde închise, se calculează suma maximă  $S_{max}$  a secvenței de elemente care se extrag din vârful stivei până la întâlnirea valorii 1000. Această valoare  $S_{max}$  se depune în vârful stivei în locul valorii 1000. Complexitatea determinării secvenței de sumă maximă este  $O(n)$ , unde  $n$  este numărul de numere.

La identificarea în expresie a unei paranteze pătrate închise, se extrag de pe stivă toate numerele până la întâlnirea valorii 2000. Pentru aceste numere trebuie să se determine valoarea mediană. Pentru aceasta utilizăm o sortare (funcția **sort** din STL, sau **QuickSort**, sau **MergeSort**) de complexitate  $O(n \log n)$ . Valoarea mediană se depune apoi în vârful stivei, în locul valorii 2000.

La terminarea parcurgerii expresiei, în stivă vor rămâne doar unul sau mai multe numere cărora trebuie să li se determine suma. Această sumă reprezintă și rezultatul evaluării expresiei.

Complexitatea totală pentru această abordare este  $O(n \log n)$ , suficient pentru punctaj maxim.

## Soluția 3 - recursivitate indirectă (Constantin Gălățan)

Expresia dată poate fi formată din una sau mai multe subexpresii. Acestea pot fi la rândul lor: termeni simpli (numere), sume, intervale pentru care se cere valoarea mediană, intervale pentru care se cere suma, intervale pentru care se cere subsecvența de sumă maximă sau altă subexpresie.

Se definesc funcțiile **Suma()**, **Mediana()**, **Secvența()**.

Se parcurge șirul de caractere. În funcție de valoarea primului caracter al șirului, expresia se parsează astfel:

- Dacă primul caracter este '[' atunci expresia este un interval pentru care se va calcula valoarea mediană sau este o sumă al cărei prim termen este o mediană. În ambele cazuri se apelează funcția **Mediana()** începând cu poziția următoare din șir. Se rețin termenii medianei într-un vector pe măsură ce aceștia sunt calculați. La întâlnirea caracterului ']', funcția **Mediana()** returnează valoarea medianei acestui vector. Pentru calculul medianei am folosit algoritmul **nth\_element()**, care determină în timp liniar poziția medianei.
- Dacă primul caracter este '(' atunci se apelează funcția **Secvența()** începând cu poziția următoare din șir. Aceasta reține într-un vector valorile termenilor secvenței pe măsură ce aceștia sunt calculați. La întâlnirea caracterului '(' se apelează **Suma()**, iar la întâlnirea unui caracter '[' se apelează **Mediana()**.
- Dacă primul caracter al expresiei este '-' sau o cifră, atunci se apelează **Suma()**. Această funcție apelează la rândul său **Secvența()** sau **Mediana()** după cum caracterul curent este '(' sau '['.

Complexitatea:  $O(n)$