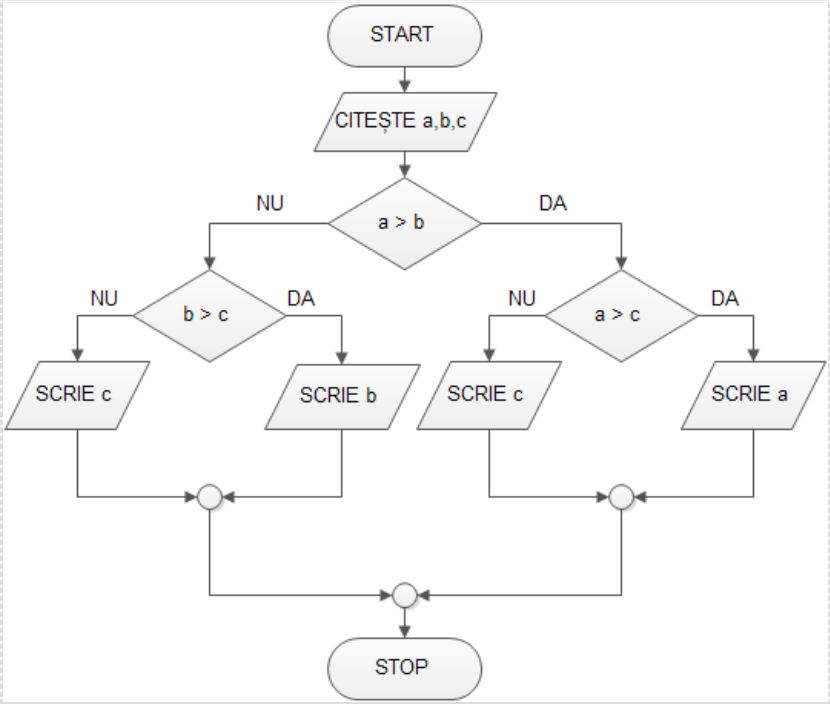
Algoritmi reprezentați prin scheme logice

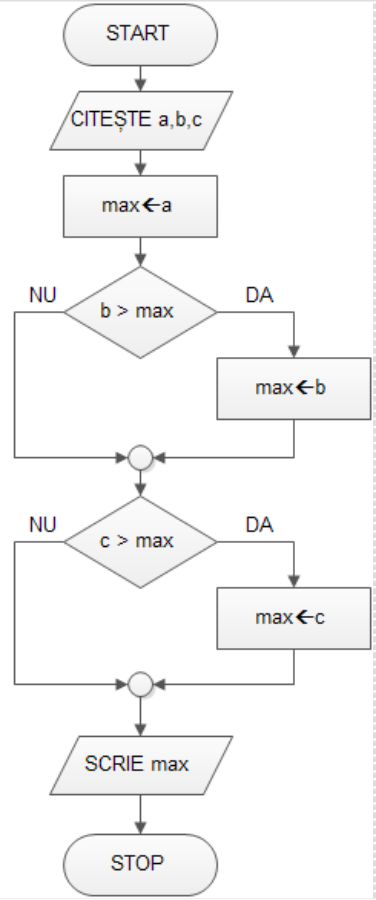
Maximul a trei numere (varianta 1)

- O primă variantă de rezolvare a problemei găsirii celui mai mare dintre 3 numere citite a, b și c este algoritmul reprezentat prin schema logică alăturată
- Se compară mai întâi primele 2 numere (a și b) și se continuă, în funcție de valoarea de adevăr a expresiei a>b pe ramura corespunzătoare: se compară cel mai mare dintre a și b cu c



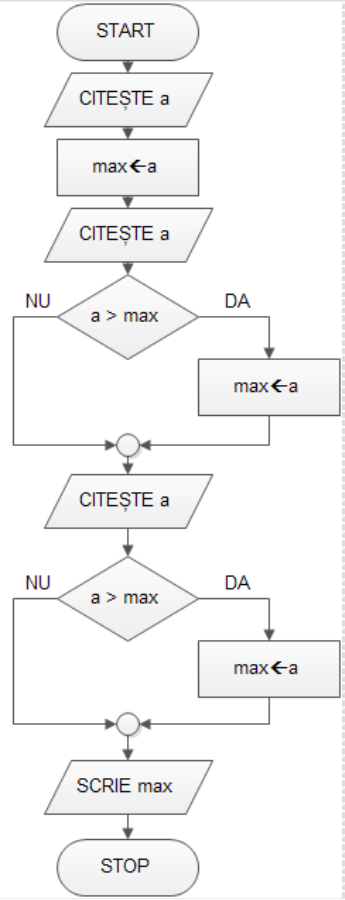
Maximul a trei numere (varianta 2)

- O altă variantă de rezolvare este cea din schema alăturată;
- Se folosește o variabilă suplimentară max, în care se păstrează cel mai mare număr de până la momentul respectiv;
- Avantajul cel mai important este reducerea riscurilor unei erori de programare;
- În locul unui bloc de decizie complex, care avea la rândul său subordonate alte două blocuri, schema alăturate folosește două blocuri de decizie simple;
- Acesta e un principiu pe care îl vom aplica și în viitor: preferăm să rezolvăm mai multe subprobleme simple în locul uneia complicate.



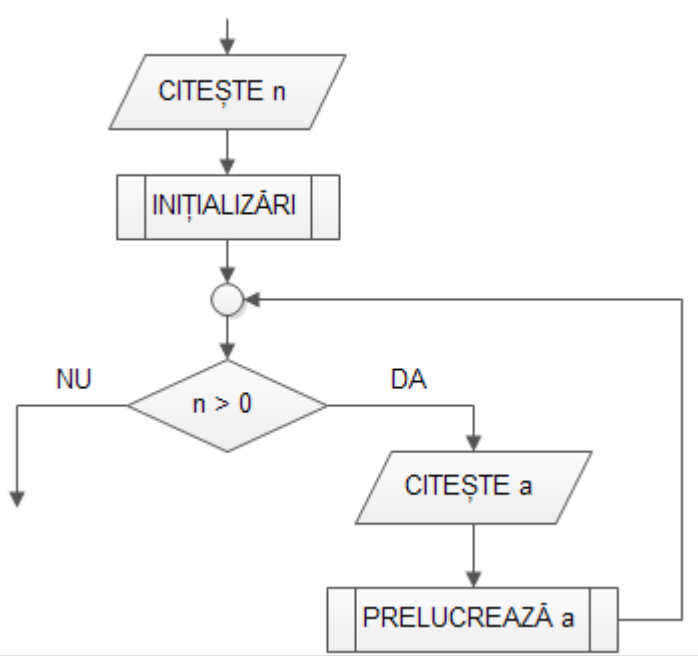
Maximul a trei numere (varianta 2’)

- Schema logică alăturată seamănă foarte bine cu cea anterioară;
- De data aceasta folosim însă o singură variabilă a, în care reținem pe rând valorile tuturor celor 3 numere citite (citirea datelor de intrare se face în 3 pași);
- Principiul de funcționare este asemănător cu cel de la Ping Pong (“învingătorul rîmâne la masă”);
- Variabila max corespunde jucătorului care se află deja la masă, care este cel mai bun dintre cei care au jucat până acum;
- Pe rând vin alți jucători (CITEȘTE a) care se confruntă cu cel aflat deja la masă;
- Dacă îl învinge (a>max), îi iau locul (max ← a);
- Un avantaj important al acestei versiuni este acela că poate fi ușor adaptat pentru a calcula maximul a oricâte numere citite.



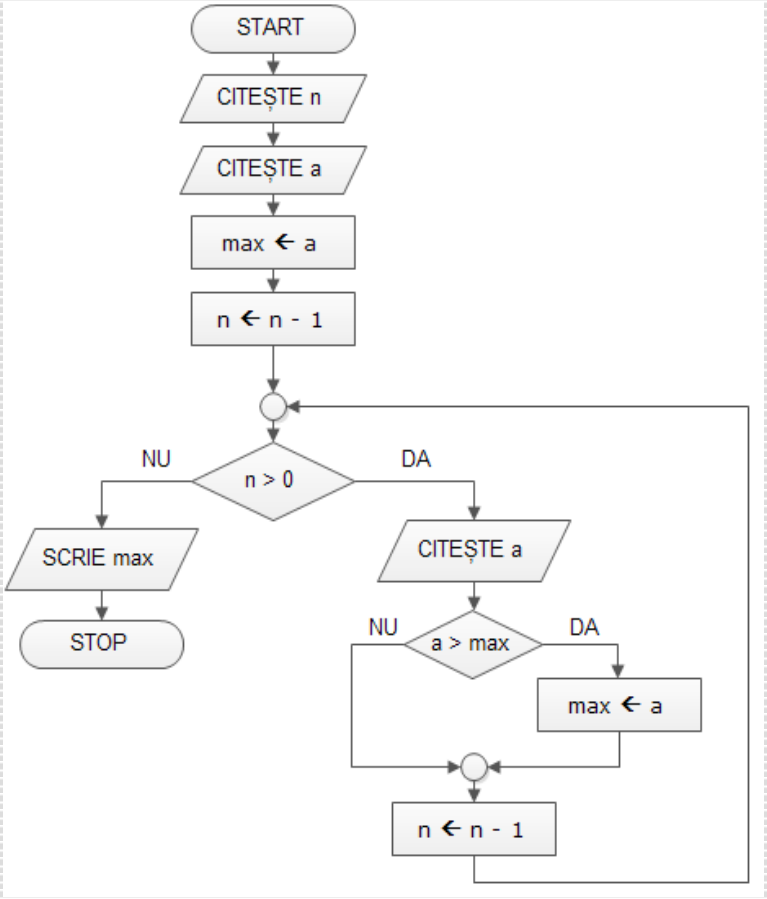
Forma generală a unui algoritm care prelucrează n numere citite

- Schema alăturată corespunde unui algoritm general de prelucrare a unui șir de n numere;
- Mai precis, se citește mai întâi n, reprezentând lungimea șirului (numărul elementelor acestuia), iar apoi, în cadrul buclei repetitive, câte un element, a, al șirului, care este prelucrat imediat după citire (prelucrarea diferă de la o problemă la alta).



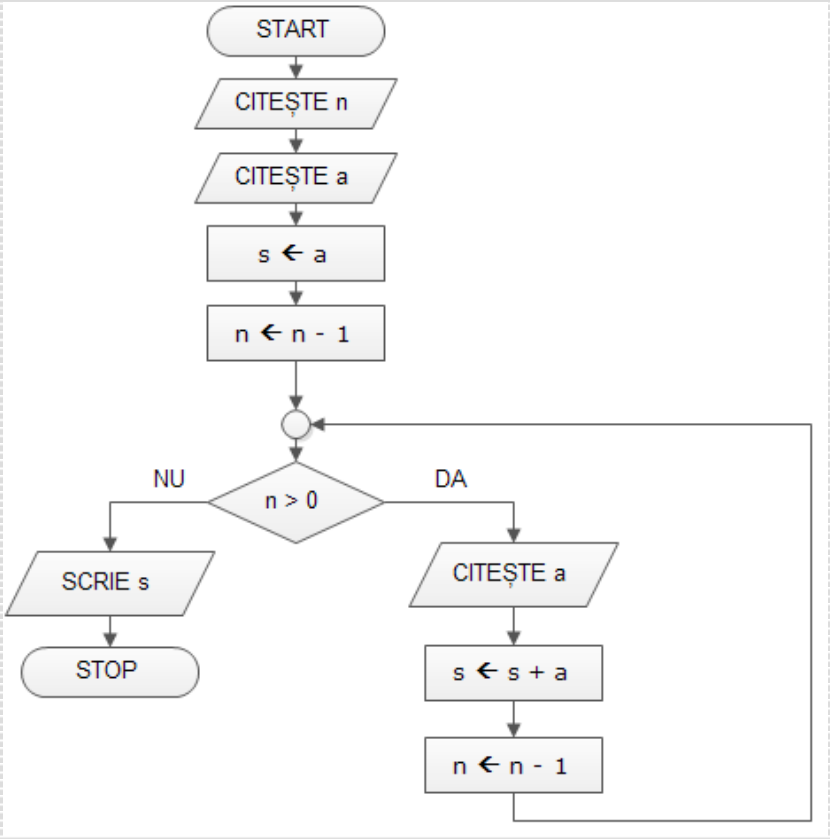
Algoritm pentru calculul maximului a n numere (n oarecare, citit)

- Algoritmul alăturat citește n și apoi n numere, care sunt reținute pe rând în variabila a și determină maximul lor;
- De fapt, pe parcursul rulării algoritmului semnificația valorii reținute de n este “câte numere mai avem de citit și prelucrat”;
- După citirea și prelucrarea fiecărui număr, n scade cu 1;
- Principiul de funcționare este din nou “învingătorul rămâne la masă”;
- Algoritmul combină ideile slide-urilor precedente: respectă șablonul pentru prelucrarea a n numere, iar fiecare număr este tratat la fel ca în cazul determinării maximului a 3 numere (e comparat cu maximul de până la acel moment și în cazul în care e mai mare ca acesta, îi ia locul).



Algoritm pentru calculul sumei a n numere (n oarecare, citit)

- Un alt exemplu de prelucrare a unui șir de n numere (cu n citit înainte) este calculul sumei elementelor șirului;
- Schema logică alăturată diferă de cea pentru determinarea maximului doar în ceea ce privește prelucrarea termenului curent a.



Algoritm pentru calculul sumei a n numere – varianta 2

- O variantă mai simplă de rezolvare a problemei anterioare se bazează pe faptul că 0 este element neutru pentru adunare;
- Dacă inițializăm s cu 0, atunci putem introduce prelucrarea primului element al șirului în bucla repetitivă, ca în schema alăturată.