


```

        nrPunctajMaxim ← 1                                { avem primul concurent cu acest punctaj maxim }
    altfel
        Dacă punctajeCurente[i] = punctajeCurente[pozMax] și pozițiiȚintă[i] =
        pozițiiȚintă[pozMax] atunci
            nrPunctajMaxim ← nrPunctajMaxim + 1          { am găsit încă un concurent cu punctajul maxim }
        SfDacă
    SfDacă
    SfPentru
    returnează pozMax
SfSubalgoritm

```

Următorul subalgoritm determină scorul final, adică ordinea concurenților, descrescătoare după punctajele obținute la cele *m* baraje. Prelucrarea se termină în momentul în care poziția concurentului cu punctaj maxim are valoarea 0.

```

Subalgoritm determRez(scor, punctajeCurente, poziiiTintă, n):
    { determinăm în vectorul scor rezultatul final al barajelor }

    Pentru i = 1, n execută:
        urmPozPosibilă[i] ← i
        { atunci când cineva concurează pentru o anumită poziție }
    SfPentru
        { va fi posibil să ajungă pe aceste poziții }
    pozMax ← maxim(punctajeCurente, poziiiTintă, n, nrPunctajMaxim)
    CâtTimp pozMax ≠ 0 execută:
        { căutăm concurenții care sunt încă în concurs și au punctaj maxim }
        Dacă nrPunctajMaxim = 1 atunci
            { dacă avem punctaj maxim unic }
            scor[urmPozPosibilă[poziiiTintă[pozMax]]] ← pozMax
            { concurentul de pe poziția pozMax, ocupă poziția corespunzătoare }
            urmPozPosibilă[poziiiTintă[pozMax]] ← urmPozPosibilă[poziiiTintă[pozMax]] + 1
            { dacă mai există concurent care concurează pentru acest loc va ajunge pe următoarea
poziție }
            poziiiTintă[pozMax] ← 0
            { concurentul de poziția pozMax și-a găsit locul }
            punctajeCurente[pozMax] ← -1
            { nu mai "are nevoie" de punctele obținute }
        altfel
            { cazul când există mai mulți concurenți cu același punctaj maxim }
            Pentru i = 1, n execută:
                { verificăm toți concurenții }
                Dacă punctajeCurente[i] = punctajeCurente[pozMax] și poziiiTintă[i] =
poziiiTintă[pozMax] atunci
                    { dacă și concurentul i are același punctaj maxim }
                    Dacă i ≠ pozMax atunci
                        { concurentul cu punctaj maxim se prelucrează separat pentru a nu pierde valoarea lui }
                    punctajeCurente[i] ← -1
                    { nu mai "are nevoie" de punctele obținute }
                    poziiiTintă[i] ← urmPozPosibilă[poziiiTintă[pozMax]]
                    { concurentul i va concura pe poziția liberă, imediat următoare }
                SfDacă
            SfDacă
            SfPentru
                punctajeCurente[pozMax] ← -1
                { concurentul cu punctaj maxim nu mai are nevoie de punctele lui }
                epozMax ← poziiiTintă[pozMax]
                { memorăm poziția țintă anterioară }
                poziiiTintă[pozMax] ← urmPozPosibilă[poziiiTintă[pozMax]]
                { poziția pe care ar putea ocupa acest concurent în următoarea etapă }
                urmPozPosibilă[epozMax] ← urmPozPosibilă[epozMax] + nrPunctajMaxim
                { următorul concurent care concurează pentru această poziție poate urma după acești
concurenți }
            SfDacă
            pozMax ← maxim(punctajeCurente, poziiiTintă, n, nrPunctajMaxim)
            { determinăm un nou
pozMax }
        SfCâtTimp
    SfSubalgoritm

```