

```

1 int porziona(int N, int mandorle[], int cannella[]) {
2
3     vector<int> bonta(N);
4     vector<int> sum(N+1);
5
6     //Il vettore bonta conterrà la differenza di cannella e mandorle
7     for (int i = 0; i < N; i++)
8         bonta[i] = (cannella[i]-mandorle[i]);
9
10    //Creazione somme cumulate
11    //all'indice 0 di sum è presente uno 0
12    partial_sum(bonta.begin(), bonta.end(), sum.begin()+1);
13
14
15    vector<int> minimi_parziali;
16    vector<int> massimi_parziali;
17
18    //Popolamento minimi parziali
19    //il primo valore sarà senz'altro 0
20    minimi_parziali.push_back(0);
21
22    for (int i = 1; i <= N; i++) {
23        // Se ci sono valori minori dell'ultimo presente nel vettore
24        if ( sum[i] < sum[minimi_parziali.back()])
25            //Mettiamo il loro indice in coda
26            minimi_parziali.push_back(i);
27    }
28
29    //Popolamento massimi parziali
30    massimi_parziali.push_back(N);
31    for (int i = N-1; i >= 0; i--) {
32        // Se ci sono valori maggiori dell'ultimo presente nel vettore
33        if (sum[i]>= sum[massimi_parziali.back()])
34            //Mettiamo il loro indice in coda
35            massimi_parziali.push_back(i);
36    }
37
38    //Variabili che indicano i due cursori
39    int fine = (int)massimi_parziali.size() - 1; // agirà sui massimi
40    int inizio = 0; // agirà sui minimi
41    int sol = 0;
42
43    for (; inizio < minimi_parziali.size() && fine != 0; inizio++) {
44
45        // Finchè possiamo scorrere avanti gli indici di fine, scorriamoli
46        // in modo tale da aver la fetta più grande a partire dal minimo_parziale[inizio]
47        // massimi_parziali[fine-1] indica il successivo massimo
48        while (fine > 0 && sum[massimi_parziali[fine - 1]] >= sum[minimi_parziali[inizio]]) fine--;
49
50        //Teniamo aggiornata la variabile con la soluzione massima
51        sol = max(sol, massimi_parziali[fine] - minimi_parziali[inizio]);
52    }
53
54    return sol;
55 }

```