

Esercizio 2 del 31/3/2020

Viene dato un main che legge in un array X una sequenza di interi che consiste di una serie di sottosequenze, ciascuna delle quali è costituita da alcuni valori (possibilmente anche 0 valori) qualsiasi e terminante con -1. La fine di tutte le sottosequenze è segnata da -2. Si osservi che, nel main, il -2 è usato come sentinella per terminare la lettura.

Esempio: Quella costituita da -2 è una sequenza valida che contiene 0 sottosequenze. Quella costituita da -1-1-2 è valida e consiste di 2 sottosequenze vuote (cioè per ciascuna c'è solo il -1 finale). Anche la seguente sequenza, 2 1 3 22 -1 -1 0 -1 33 1 -1 -2, è valida e contiene 4 sottosequenze: 2 1 3 22 -1, -1, 0 -1, e 33 1 -1. Nel seguito numereremo queste sottosequenze con interi crescenti, 0,1,2 eccetera. Osserviamo che nell'esempio precedente la sottosequenza 0 inizia all'indice 0 di X, la sottosequenza 1 all'indice 5, la sottosequenza 2 all'indice 6, la sottosequenza 3 all'indice 8, e per indici superiori a 3, X non ha più sottosequenza corrispondente.

Esercizio: si chiede di scrivere una funzione `int S(int *A, int n)` che soddisfi le seguenti pre e postcondizione:

PRE=(A contiene una sequenza di interi che termina con -2 e che consiste di una serie (possibilmente vuota) di sottosequenze ciascuna delle quali termina con -1, inoltre $n \geq 0$)

POST=(se A contiene la sottosequenza n ed essa inizia in `A[i]`, allora S restituisce i, altrimenti S restituisce -2)

Viene fornito un main che compie le operazioni di lettura e che invoca la funzione S che è da fare.

Correttezza: associare un invariante ad ogni ciclo di S e dimostrare che S è tale che se S inizia in uno stato del calcolo che rende vera la PRE e se S termina, allora termina in uno stato del calcolo che soddisfa la POST.