

Programmazione I (Tucci/Distasi)

PR1 INT MT/RD 10/11/2022

Modello: 1

Cognome: _____

Nome: _____

Matricola: _____

Email: _____

Regole del gioco: Compilare i dati personali prima d’incominciare. Una volta iniziata la prova, non è consentito lasciare l’aula. Usare questi stessi fogli (compreso il retro, dove necessario) per rispondere. *Buon lavoro!*

1. Specificare, progettare e scrivere una funzione

```
int copia_unici(int src[], int dest[], int size)
```

che prende come parametri due array d’interi `src[]` e `dest[]`, entrambi di taglia `size`, e copia all’inizio di `dest[]` gli elementi di `src[]` evitando le ripetizioni: se un elemento di `src[]` compare due o più volte, anche non consecutive, soltanto la sua prima occorrenza sarà copiata. La funzione restituisce il numero di elementi copiati.

Per esempio: se `src[] = {-6, 2, 0, 2, 13, -6, -6, 2, 0}`
e invochiamo `copia_unici(src, dest, 9)`
i primi 4 elementi di `dest[]` diventeranno `{-6, 2, 0, 13}`,
e la funzione restituirà 4.

La stesura delle specifiche (dati utilizzati, pre- e post-condizioni) e la progettazione top-down per raffinamenti successivi sono parte integrante della prova.

Risposte per il modello 1

1. Specificare, progettare e scrivere una funzione

```
int copia_unici(int src[], int dest[], int size)
```

che prende come parametri due array d'interi `src[]` e `dest[]`, entrambi di taglia `size`, e copia all'inizio di `dest[]` gli elementi di `src[]` evitando le ripetizioni: se un elemento di `src[]` compare due o più volte, anche non consecutive, soltanto la sua prima occorrenza sarà copiata. La funzione restituisce il numero di elementi copiati.

Per esempio: se `src[] = {-6, 2, 0, 2, 13, -6, -6, 2, 0}`

e invochiamo `copia_unici(src, dest, 9)`

i primi 4 elementi di `dest[]` diventeranno `{-6, 2, 0, 13}`,

e la funzione restituirà 4.

La stesura delle specifiche (dati utilizzati, pre- e post-condizioni) e la progettazione top-down per raffinamenti successivi sono parte integrante della prova.

Risposta Ecco una possibile codifica.

```
/* l'elemento ago compare in pagliaio[] entro indice n-1? SI/NO
 */
int appare_prima_di_nmo(int ago, int n, int pagliaio[], int size_pagliaio)
{
    int i;

    if (n > size_pagliaio)          // questo non dovrebbe mai succedere
    {
        my_error("appare_nei_primi_n(): ricerca oltre limiti: %d > %d\n",
                  n, size_pagliaio);
    }
    for (i = 0; i < n; i++)
    {
        if (pagliaio[i] == ago)    // appare! Trovato
        {
            return 1;
        }
    }
    return 0;                      // non appare
}

int copia_unici(int src[], int dest[], int size)
{
    int i, j, ncopiat, x;

    for (i = 0, j = 0; i < size; i++)
    {
        x = src[i];                // elemento da copiare
        if (!appare_prima_di_nmo(x, j, dest, size))
        {
            dest[j++] = x;
        }
    }
    ncopiat = j;
    while (j < size)                // azzeriamo elementi rimanenti
    {
        dest[j++] = 0;
    }
    return ncopiat;
}
```