

Programmazione I (Tucci/Distasi)

PR1 MT/RD 12/01/2022

Modello: 1

Cognome: _____

Nome: _____

Matricola: _____

Email: _____

Regole del gioco: Compilare i dati personali prima d'incominciare. Alla fine della prova, inviare un singolo file pdf (non immagini separate, non un link web) all'indirizzo email

prog1unisa.aula1@gmail.com

Buon lavoro!

1. Consideriamo una struttura dati per la rappresentazione di un libro.

```
typedef struct
{
    char autore[50];
    char titolo[100];
    int anno;
} Libro;
```

Scrivere una funzione

```
Libro * libri_nuovi(Libro a[], int n, int anno0, int * nnuovi)
```

che riceve come parametri un array `a[]` di `n` oggetti di tipo `Libro` e restituisce un nuovo array, contenente copie di tutti i libri in `a[]` pubblicati nell'anno `anno0` o successivamente. L'array restituito va allocato dinamicamente sulla base della dimensione strettamente necessaria. La funzione `libri_nuovi()` comunica al chiamante la dimensione dell'array restituito usando il parametro di output `nnuovi`.

Se nessun libro è abbastanza recente, `libri_nuovi()` restituisce `NULL`.

2. Scrivere una funzione

```
int copia_nuovi(FILE *fin, int nlibri, FILE *fout, int anno0)
```

che lavora su due file binari già aperti rispettivamente in lettura e scrittura, con descrittori `fin` e `fout`. Il file input `fin` contiene una serie di `nlibri` oggetti di tipo `Libro`. La funzione copia nel file output `fout` tutti i libri pubblicati nell'anno `anno0` o successivamente, e restituisce il numero di libri copiati, senza lasciare file aperti all'uscita o memoria allocata inaccessibile.

Usare la funzione `libri_nuovi()` scritta nell'esercizio precedente.

Risposte per il modello 1

1. Consideriamo una struttura dati per la rappresentazione di un libro.

```
typedef struct
{
    char autore[50];
    char titolo[100];
    int anno;
} Libro;
```

Scrivere una funzione

```
Libro * libri_nuovi(Libro a[], int n, int anno0, int * nnuovi)
```

che riceve come parametri un array `a[]` di `n` oggetti di tipo `Libro` e restituisce un nuovo array, contenente copie di tutti i libri in `a[]` pubblicati nell'anno `anno0` o successivamente. L'array restituito va allocato dinamicamente sulla base della dimensione strettamente necessaria. La funzione `libri_nuovi()` comunica al chiamante la dimensione dell'array restituito usando il parametro di output `nnuovi`.

Se nessun libro è abbastanza recente, `libri_nuovi()` restituisce `NULL`.

Risposta

Ecco una possibile soluzione.

```
void *xmalloc(size_t nbytes)    // malloc() con controllo errore
{
    void *result;

    if ((result = malloc(nbytes)) == NULL)
    {
        fprintf(stderr, "malloc(%lu) failed. Exiting.\n", nbytes);
        exit(-1);
    }
    return result;
}

Libro *copia_libro(Libro * dest, Libro * src)    // return dest, like strcpy()
{
    strcpy(dest->autore, src->autore);
    strcpy(dest->titolo, src->titolo);
    dest->anno = src->anno;
    return dest;
}
```

```

Libro *libri_nuovi(Libro a[], int n, int anno0, int *nnuovi)
{
    int i, j, contanuovi;
    Libro *new_array;

    for (i = 0, contanuovi = 0; i < n; i++)          // contiamo i libri "nuovi"
    {
        if (a[i].anno >= anno0)
        {
            contanuovi++;
        }
    }
    *nnuovi = contanuovi;
    if (contanuovi == 0)                            // se nessun nuovo, abbiamo finito
    {
        return NULL;
    }
    // else, creiamo spazio e copiamoci i libri nuovi
    new_array = (Libro *) xmalloc(sizeof(Libro) * contanuovi);
    for (i = 0, j = 0; i < n; i++)
    {
        if (a[i].anno >= anno0)
        {
            copia_libro(&new_array[j++], &a[i]);
        }
    }
    return new_array;
}

```

2. Scrivere una funzione

```
int copia_nuovi(FILE *fin, int nlibri, FILE *fout, int anno0)
```

che lavora su due file binari già aperti rispettivamente in lettura e scrittura, con descrittori `fin` e `fout`. Il file input `fin` contiene una serie di `nlibri` oggetti di tipo `Libro`. La funzione copia nel file output `fout` tutti i libri pubblicati nell'anno `anno0` o successivamente, e restituisce il numero di libri copiati, senza lasciare file aperti all'uscita o memoria allocata inaccessibile.

Usare la funzione `libri_nuovi()` scritta nell'esercizio precedente.

Risposta

Ecco una possibile soluzione.

```
int copia_nuovi(FILE * fin, int nlibri, FILE * fout, int anno0)
{
    Libro *tuttilibri, *nuovilibri;
    int ncopiati, nletti, nnuovi;

    tuttilibri = xmalloc(sizeof(Libro) * nlibri); // spazio per tutti i libri
    nletti = fread(tuttilibri, sizeof(Libro), nlibri, fin);
    if (nletti != nlibri)
    {
        fprintf(stderr, "Letti %d libri anziche' %d.\n", nletti, nlibri);
    }

    nuovilibri = libri_nuovi(tuttilibri, nletti, anno0, &nnuovi);
    ncopiati = fwrite(nuovilibri, sizeof(Libro), nnuovi, fout);
    if (ncopiati != nnuovi)
    {
        fprintf(stderr, "Copiati %d libri anziche' %d.\n", ncopiati, nnuovi);
    }

    // fare pulizia: liberare memoria ormai inutile, chiudere file
    free(tuttilibri);
    free(nuovilibri);
    fclose(fin);
    fclose(fout);
    return ncopiati;
}
```