

Programmazione I (Tucci/Distasi)

PR1 MT/RD 14/07/2022

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. Scrivere una funzione C

```
int * copy_neg_seq(int a[], int n, int *seqlen)
```

che ha come parametri un array `a[]` di numeri interi con la sua taglia `n` e restituisce un nuovo array contenente la prima sequenza di numeri negativi consecutivi che appare in `a[]`.

L'array restituito dev'essere allocato dinamicamente sulla base dell'effettivo numero di elementi necessario. Il parametro output `seqlen` rifletterà la taglia del risultato.

Ai fini di questo esercizio, anche un solo numero negativo costituisce una sequenza (di lunghezza 1). Se l'array `a[]` non contiene numeri negativi, la funzione restituisce `NULL` e imposta a zero `seqlen`.

Esempi:

Se `a[] = { 1, 2, 3, -4, -5, 6, 7 }`, il risultato sarà `{ -4, -5 }`.

Se `a[] = { 1, -2, 3, -4, -5, -6 }`, il risultato sarà `{ -2 }`.

Se `a[] = { 1, 2, 3, 4 }`, il risultato sarà `NULL`.

2. Con riferimento all'esercizio precedente, scrivere una funzione

```
void list_neg(FILE *fout, int a[], int n)
```

in cui, come prima, `a[]` è un array di `n` interi, mentre `fout` è un file binario già aperto in scrittura. La funzione scrive nel file la prima sequenza di numeri negativi consecutivi che appare in `a[]`.

Saranno preferite le soluzioni che usano la funzione scritta nell'esercizio precedente.

Risposte per il modello 1

1. Scrivere una funzione C

```
int * copy_neg_seq(int a[], int n, int *seqlen)
```

che ha come parametri un array `a[]` di numeri interi con la sua taglia `n` e restituisce un nuovo array contenente la prima sequenza di numeri negativi consecutivi che appare in `a[]`.

L'array restituito dev'essere allocato dinamicamente sulla base dell'effettivo numero di elementi necessario. Il parametro output `seqlen` rifletterà la taglia del risultato.

Ai fini di questo esercizio, anche un solo numero negativo costituisce una sequenza (di lunghezza 1). Se l'array `a[]` non contiene numeri negativi, la funzione restituisce `NULL` e imposta a zero `seqlen`.

Esempi:

Se `a[] = { 1, 2, 3, -4, -5, 6, 7 }`, il risultato sarà `{ -4, -5 }`.

Se `a[] = { 1, -2, 3, -4, -5, -6 }`, il risultato sarà `{ -2 }`.

Se `a[] = { 1, 2, 3, 4 }`, il risultato sarà `NULL`.

Risposta

Ecco una possibile soluzione.

```
#include <stdio.h>
#include <stdlib.h>

void *xmalloc(size_t nbytes)
{
    void *result;

    result = malloc(nbytes);
    if (nbytes != 0 && result == NULL)
    {
        fprintf(stderr, "xmalloc(%ld) failed. Bye.\n", nbytes);
        exit(-1);
    }
    return result;
}
```

```

int *copy_neg_seq(int *a, int n, int *seqlen)
{
    int *result;           // will be the returned array
    int start;             // where does the sequence start?
    int len;               // length of the current sequence
    int i;

    i = 0;
    while (i < n && a[i] >= 0)    // skip nonnegatives
    {
        i++;
    }
    // now we are either on a negative number or at the end of a[]
    // the commented code below is unnecessary
    // because the following while won't be executed at the end of a[]
    // if (i >= n)                // end of a[]
    // {
    //     *seqlen = 0;
    //     return NULL;
    // }
    // else: a[i] is negative
    start = i;               // sequence of negative numbers begins here
    len = 0;                 // count consecutive negative numbers
    while (i < n)             // if i>=n we're at the end, while is skipped
    {
        if (a[i] < 0)
        {
            len = len + 1;    // one more negative
        } else
        {
            // nonnegative: sequence ends here
            break;
        }
        i++;
    }
    result = xmalloc(len * sizeof(int));
    for (i = 0; i < len; i++)
    {
        result[i] = a[start + i]; // copy negative numbers into result[]
    }
    *seqlen = len;
    return result;
}

```

2. Con riferimento all'esercizio precedente, scrivere una funzione

```
void list_neg(FILE *fout, int a[], int n)
```

in cui, come prima, `a[]` è un array di `n` interi, mentre `fout` è un file binario già aperto in scrittura. La funzione scrive nel file la prima sequenza di numeri negativi consecutivi che appare in `a[]`.

Saranno preferite le soluzioni che usano la funzione scritta nell'esercizio precedente.

Risposta

Ecco una possibile soluzione.

```
#include <stdio.h>
#include <stdlib.h>

// Variation 1: don't use copy_neg_seq() from the previous exercise
void list_neg1(FILE * fout, int * a, int n)
{
    int i;

    i = 0;
    while (i < n && a[i] >= 0)    // skip nonnegatives
    {
        i++;
    }
    // now we are either on a negative number or at the end of a[]
    // if (i >= n), we're at the end of a[]
    // and the next while won't be executed.

    while (i < n)
    {
        if (a[i] < 0)
        {
            fwrite(&a[i], sizeof(int), 1, fout); // write one more negative
        } else
        {
            // nonnegative: sequence ends here
            break;
        }
        i++;
    }
    fclose(fout);
    return;
}

// Variation 2 (better): use copy_neg_seq() from the previous exercise
void list_neg2(FILE * fout, int * a, int n)
{
    int *negatives;
    int newsize;

    negatives = copy_neg_seq(a, n, &newsize);
    if (negatives != NULL)
    {
        fwrite(&negatives[0], sizeof(int), newsize, fout);
    }
    fclose(fout);
    return;
}
```