

INFORMATICA - CdL in FISICA

Esercitazione del 19 maggio 2011

N.B.: Negli esercizi di programmazione, viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione (come, ad esempio, `continue`, `break` e istruzioni di `return` all'interno di cicli che ne provochino l'uscita forzata).

ESERCIZIO 1

Definire una procedura che, dato un vettore di interi distinti di dimensione non nulla e la sua lunghezza, lo modifica facendo in modo che:

- gli elementi pari precedano gli elementi dispari
- nel vettore finale l'ordine relativo degli elementi pari sia immutato rispetto al vettore iniziale
- nel vettore finale l'ordine relativo degli elementi dispari sia immutato rispetto al vettore iniziale

Nella soluzione non possono essere utilizzati vettori di supporto né strutture dinamiche.

Dato ad esempio il vettore iniziale

15	4	3	7	8	2	9	23	40	10
----	---	---	---	---	---	---	----	----	----

la procedura lo deve modificare come segue:

4	8	2	40	10	15	3	7	9	23
---	---	---	----	----	----	---	---	---	----

ESERCIZIO 2

Definire una funzione che, dato un array di interi e la sua lunghezza (che si suppone non nulla), verifica se il vettore contiene o meno tutti elementi distinti tra loro. Nella definizione di tale funzione l'unico costrutto iterativo utilizzabile è il costrutto **while**. Si giustifichi poi la correttezza della soluzione proposta.

ESERCIZIO 3

Definire una funzione **ricorsiva** che, dati un vettore *vet* di interi, la sua dimensione *dim* ed un intero *x* restituisce il numero di occorrenze di *x* in *vet*.

Dati ad esempio il vettore

15	4	9	7	8	2	9	23	40	10
----	---	---	---	---	---	---	----	----	----

lungo 10, e l'intero 9, la funzione restituisce il valore 2. Dati lo stesso vettore e l'intero 200 la funzione restituisce 0.

ESERCIZIO 4

Dato un vettore *v* di elementi di tipo *T*, la *distanza* tra due elementi $v[i]$ e $v[j]$, con $i \leq j$ è data dall'intero non negativo $j - i$. Si definisca una funzione che, dato un vettore di caratteri, la sua dimensione, ed un intero $k > 0$ restituisce *true* se nel vettore esistono due elementi uguali tra loro e distanti *k*, e restituisce *false* altrimenti. Nella soluzione, l'unico costrutto iterativo utilizzabile è il comando **while**. Si giustifichi poi la correttezza della soluzione proposta.

ESERCIZIO 5

Data la seguente definizione

```
void p(int x, int y)
{
  int a, b, *h, *k;
  a = (x + y);
  b = (x * y);
  h = &a;
  k = &b;
  if ((x % y) != 0)
  { h = k;
    k = &a; }
  else
  { *k = (*k) + (*h);
    *h = (*k) + (*h); }
  printf("%d %d %d %d\n", a, b, *h, *k);
}
```

dire quali valori vengono stampati in corrispondenza delle due chiamate:

i) p(5, 5)

ii) p(8, 3)

ESERCIZIO 6

Definire in modo **ricorsivo** una procedura che, dato un array di caratteri e la sua dimensione, rimpiazza nell'array tutti i caratteri non alfabetici con il carattere '.' e restituisce il numero di rimpiazzamenti effettuati.

Ad esempio, dato il vettore

'f'	'\$'	'}'	'a'	'c'	'2'	'i'	'2'	'l'	'e'
-----	------	-----	-----	-----	-----	-----	-----	-----	-----

la procedura lo deve modificare come segue:

'f'	'.'	'.'	'a'	'c'	'.'	'i'	'.'	'l'	'e'
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

e deve restituire il valore 4.

ESERCIZIO 7

Scrivere una funzione che calcola l' n -esimo numero di Fibonacci senza usare la ricorsione.