

# Esame di Tecniche di Programmazione

13 febbraio 2015

Istruzioni per lo svolgimento della prova:

- Inserire nel file **studente.txt** i dati personali richiesti, quindi *salvare e chiudere* il file.
- All'esercizio *i* è associata la directory **esercizio*i***, contenente i file necessari allo svolgimento dell'esercizio. Ciascun esercizio deve essere svolto modificando *esclusivamente* i file **esercizio.h** (se necessario) ed **esercizio.c** presenti nella directory corrispondente. Tali file *non devono essere spostati* dalla directory in cui si trovano. *Nessun* altro file deve essere modificato. Non devono essere creati ulteriori file.
- Il file contenente il programma principale di ciascun esercizio è **test.c**. Tale file deve essere compilato usando il **makefile** fornito nella directory ed il corrispondente eseguibile (**test**) deve essere lanciato per eseguire i test di correttezza e conoscerne l'esito. Il file **test.c** *non deve essere modificato*.
- Verranno corretti esclusivamente gli esercizi la cui compilazione dia esito positivo. In caso contrario verrà assegnato punteggio nullo.
- È permesso consultare la documentazione (in inglese) della libreria standard C, raggiungibile all'indirizzo riportato nel file **link-documentazione-C.txt**.
- Per eventuali problemi o dubbi, *chiedere al docente*.

## Esercizio 1

Implementare in C la funzione:

- `int* accumulaArray(char* nomefile, int* n)`

che, presi in input il nome di un file di testo (**nomefile**) contenente valori interi ed un puntatore **n** ad intero, restituisca un array di interi la cui prima componente è pari al primo valore contenuto nel file, la seconda alla somma dei primi due valori nel file, e così via fino all'ultima componente, pari alla somma di tutti i valori nel file. La funzione deve inoltre assegnare alla variabile puntata da **n** la dimensione dell'array restituito. I valori nel file sono separati da spazi bianchi o caratteri di ritorno a capo.

**Esempio** Con il seguente file di input:

```
10 20
1
2
37 14
```

La funzione deve restituire l'array: [10, 30, 31, 33, 70, 84].

## Esercizio 2

L'operazione *rotazione di  $k$  posizioni* di una lista consiste in una traslazione a sinistra di  $k$  posizioni di tutti gli elementi della lista, considerando l'ultimo elemento della lista a sinistra del primo. Ad esempio, il risultato della rotazione di 3 posizioni della lista  $[2, 4, 6, 7, 4, 8]$  è la lista  $[7, 4, 8, 2, 4, 6]$ . Usando il tipo di dato `TipoSCL` (definito nel file `esercizio.h`) per la rappresentazione di strutture collegate lineari contenenti elementi di tipo `int`, realizzare in C la funzione

- `void ruota(TipoSCL* pscl, int k)`

che, presi in input un puntatore `pscl` ad una lista di interi ed un intero `k`, ruoti la SCL puntata di `k` posizioni.

NOTA: Assumere  $k \geq 0$ .

## Esercizio 3

Implementare in C la funzione **ricorsiva**:

- `int diagonalePrincipale(int** mat, int n)`

che, presa in input una matrice quadrata di interi `mat`, di dimensione `n*n`, restituisca il valore ottenuto sommando i valori della diagonale principale (ovvero i valori aventi indici di riga e colonna coincidenti). Si assuma che la matrice sia non vuota. **Non è consentito l'uso di istruzioni di ciclo, né nel corpo della funzione né in eventuali funzioni ausiliarie.**

**Esempio** Con la seguente matrice di input:

$$\text{mat} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

L'invocazione `diagonalePrincipale(mat,3)` deve restituire il valore  $1 + 5 + 9 = 15$ .