

FONDAMENTI DI INFORMATICA II° PROVA INTERMEDIA

COMPITO A *tempo a disposizione 120 minuti*

Esercizio A1: ricorsione e file (10 pt)

Realizzare un'applicazione che consiste di cinque funzioni.

- **(5 pt)** Una funzione **ricorsiva** *tuttiAlternati* che verifica se, in un array di numeri interi, i numeri positivi e negativi si alternano. Ad esempio, la funzione *tuttiAlternati* deve restituire il valore 1 se l'array parametro è [-3,6,-4,1]. Deve restituire valore 1 anche se l'array parametro è [2,-1,4]. La funzione *tuttiAlternati* ha due parametri: l'array e la sua lunghezza. Nel caso in cui si desideri utilizzare un terzo parametro per realizzare la ricorsione, deve essere definita un'ulteriore funzione *tuttiAlternatiRic* che ha tre parametri e che realizza la ricorsione. In tal caso la funzione *tuttiAlternati* invoca la funzione *tuttiAlternatiRic*.
- **(1 pt)** Una funzione *scriviNumeri* che scrive i valori di un array di interi su di un file. La funzione riceve come parametro l'array e la sua lunghezza. Il file si deve chiamare interi.dat (nel caso sia binario) oppure interi.txt (nel caso in cui sia testuale). La scelta se utilizzare un file binario oppure un file di testo è lasciata allo studente.
- **(2 pt)** Una funzione *quantiNumeri* che determina quanti numeri interi sono memorizzati su un file dal nome interi.dat (nel caso in cui si utilizzi un file binario) oppure interi.txt (nel caso in cui si utilizzi un file testuale).
- **(2 pt)** Una funzione *leggiNumeri* che legge una sequenza di interi da un file dal nome interi.dat (nel caso in cui si utilizzi un file binario) oppure interi.txt (nel caso in cui si utilizzi un file testuale) e li memorizza su un array che è ricevuto come parametro.
- Una funzione *main* che è già stata implementata in un file **a1.c** (ad eccezione dei messaggi finali per l'utente) il cui codice può essere scaricato e copiato da moodle (il file contiene anche le intestazioni delle altre funzioni). La funzione chiede all'utente se vuole inserire una sequenza di interi, oppure leggere la sequenza che è memorizzata sul file. Dopo aver memorizzato tale sequenza in un array, viene verificato se nell'array numeri positivi e negativi compaiono in maniera alternata. Questa informazione viene quindi stampata per l'utente.

Esercizio A2: liste (12 pt)

Realizzare un'applicazione per gestire una lista di segmenti orizzontali. L'applicazione deve gestire ciascun segmento come una struttura con due campi, che rappresentano il punto del piano che è l'estremo sinistro del segmento e la sua lunghezza (un valore reale). L'estremo sinistro del segmento è a sua volta una struttura con due campi, che rappresentano le coordinate del punto nel piano (due valori reali).

- **(1pt)** Definire delle strutture per rappresentare un punto, un segmento, ed un nodo della lista.
- **(1,5pt)** Definire una funzione che visualizza una lista di segmenti, stampando per ciascun segmento le coordinate dell'estremo sinistro, le coordinate dell'estremo destro e la lunghezza del segmento.
- **(5,5pt)** Definire una funzione che cancella il primo segmento della lista la cui lunghezza è pari a 12 o, se tale segmento non esiste, stampa il messaggio "non esiste nessun segmento lungo 12".
- **(4pt)** Definire una funzione che inserisce un segmento in coda alla lista.
- Definire una funzione main che crea la lista e gestisce l'interazione con l'utente. La funzione main è già parzialmente implementata in un file **a2.c** che può essere scaricato da Moodle.