

specifica PROGETTO B

Istruzioni lettura specifica

Per ogni specifica, in linea, troverete il punteggio massimo ottenibile. La sua implementazione priva di errori comporterà il punteggio massimo, la sua mancanza il punteggio minimo [0]. Eventuali errori saranno valutati di conseguenza ma non potranno mai portare a valori inferiori a zero.

Obiettivo del progetto

Il programma in oggetto dovrà permettere alle Agenzie Immobiliari la gestione e ricerca di immobili sulla base delle loro caratteristiche. Devono essere quindi permesse operazioni di inserimento, modifica, eliminazione, ricerca, ordinamento ed esportazione dei dati inerenti ai soggetti memorizzati nel sistema.

Data consegna specifiche: 03/02/2021 00:00

Data termine consegne: 30/09/2021 23:59

Definizione strutture ed enumerazioni

Nome struttura: **Immobile** [2pt]

Ogni immobile registrato dovrà contenere (esclusivamente) le seguenti informazioni:

- a) Prezzo (int)
- b) Metri quadri (float)
- c) Numero vani (intero)
- d) Posizione (struttura, vedi sotto)
- e) Codice identificativo immobile (vettore 9 caratteri -lettere o cifre- NON deve contenere il Terminatore di stringa)
- f) Numero verande (intero)
- g) Tipologia (enumerazione, vedi sotto)
- h) Ascensore (booleano)
- i) Posto auto (booleano)
- j) Scantinato (booleano)
- k) Giardino (booleano)
- l) Uso Immobile (enumerazione, vedi sotto)
- m) Stato Immobile (enumerazione, vedi sotto)

Nome struttura: **Posizione** [1pt]

Ogni posizione dovrà contenere (esclusivamente) le seguenti informazioni:

- a) Regione (stringa 32 caratteri, compreso terminatore di stringa)
- b) Comune (stringa 32 caratteri, compreso terminatore di stringa)
- c) Via (stringa 32 caratteri, compreso terminatore di stringa)
- d) Numero civico (intero)

Nome enumerazione: **Tipologia** [1pt]

L'enumerazione dovrà contenere i seguenti valori: indipendente, semi-indipendente, appartamento, altro.

Nome enumerazione: *UsoImmobile* [1pt]

L'enumerazione dovrà contenere i seguenti valori: abitativo, ufficio, locale commerciale.

Nome enumerazione: *StatoImmobile* [1pt]

L'enumerazione dovrà contenere i seguenti valori: nuovo, ristrutturato, parzialmente ristrutturato, da ristrutturare, rudere.

Vincoli:

- Il codice identificativo deve sempre contenere 9 caratteri ed ha un valore alfanumerico formato dalle prime due lettere della regione, le prime due lettere del comune e 5 valori numerici. Esempio per un immobile sito nel comune di Cagliari un codice potrebbe essere "SACA10114"
- La regione, il comune e la via possono solo lettere (maiuscole o minuscole) e spazi
- Il prezzo è indicato in euro e deve essere compreso tra 0 e 1e+7
- I metri quadri devono essere compresi tra 1 e 10.000
- Il numero civico è un intero e può avere un massimo di 3 cifre
- Il valore booleano true indica la presenza dell'ascensore, false altrimenti.
- Il valore booleano true indica la presenza del posto auto, false altrimenti.
- Il valore booleano true indica la presenza di uno scantinato, false altrimenti.
- Il valore booleano true indica la presenza di uno giardino, false altrimenti.

L'elenco di Immobili dovrà essere gestito con un array dinamico e devono essere implementate le seguenti funzionalità:

- Creare una procedura inserisciImmobile() [4pt] che riceva in ingresso un puntatore ad un Immobile e permetta l'inserimento dei dati di un immobile. La procedura deve effettuare i controlli (direttamente nella procedura o utilizzando subroutine esterne – scelta consigliata) necessari per far rispettare i vincoli elencati in precedenza, in più:
 - Tutti i campi sono obbligatori;
 - I valori di Tipologia, UsoImmobile e StatoImmobile possono essere scelti solo tra i valori delle enumerazione perciò quando verrà chiesto all'utente di inserire un valore corrispondente all'enumerazione dovranno essere stampate le opzioni disponibili con i rispettivi valori interi associati.
 - In caso di inserimento errato l'utente dovrà inserire nuovamente il valore richiesto, per ogni tipo di informazione richiesta.
- Creare una procedura generaImmobile() [2pt] che riceva in ingresso un puntatore ad un Immobile e generi casualmente tutte le informazioni relative all'immobile seguendo i vincoli riportati in precedenza.
- Creare una procedura modificaImmobile() [1pt] che riceva in ingresso un puntatore ad un Immobile e permetta di modificare le informazioni sull'immobile (ovviamente sempre seguendo i vincoli) chiedendo iterativamente all'utente se intende modificare o confermare un campo
- Creare una procedura stampaImmobile() [2pt] che riceva in ingresso un puntatore ad un Immobile e stampi tutte le informazioni, in modo ordinato e leggibile relative all'immobile

- I valori corrispondenti alle enumerazioni devono essere stampati sempre in formato letterale e non in formato intero.
- Creare una procedura stampaListaImmobili() [1pt] che riceva in ingresso l'elenco degli immobili e stampi, sfruttando la procedura precedente, tutte le informazioni, sugli immobili
- Creare una funzione cercaImmobili() [5pt] che riceva in ingresso l'elenco degli immobili, chieda all'utente di inserire il campo necessario per fare la ricerca:
 - ricerca per Regione
 - ricerca per Tipologia
 - ricerca per StatoImmobile
 e restituisca tutti gli immobili con le caratteristiche ricercate
- Creare una procedura ordinaPrezzo() [3pt] che riceva in ingresso l'elenco degli immobili e li riordini in base al prezzo e dovrà essere chiesto all'utente se vuole ordinare i risultati in ordine crescente o decrescente
- Creare una funzione vendiImmobile() [2pt] che riceva in ingresso l'elenco degli immobili e l'indice dell'immobile venduto e lo elimini dall'elenco degli immobili
- Creare una procedura salvaImmobili() [2pt] che riceva in ingresso l'elenco degli immobili e il nome di un file binario e che permetta di salvare tutti gli immobili sul file
- Creare una funzione caricaImmobili() [2pt] che riceva in ingresso l'elenco degli immobili e il nome di un file binario e che permetta di caricare una serie di immobili dal file

Documentazione

Tutto il progetto dovrà essere commentato in tutte le sue parti. In particolare, dovrà essere presente un commento prima dell'implementazione di ogni subroutine.

Cause di non accettazione del progetto

Il progetto non verrà corretto nei seguenti casi:

- Il progetto non compila (anche a causa di semplici errori di sintassi e/o utilizzo di librerie non standard);
- Non è presente alcun commento;
- Viene consegnato dopo il termine previsto (anche se 1 minuto dopo la scadenza);
- Contiene al suo interno codice copiato da altri, ad eccezione del codice realizzato e fornito dai docenti durante tutto il corso;
- Se contiene tutto il codice in un singolo file (ad es. main.c) o libreria;
- Se realizzato su ambiente diverso da CLion.

Il voto minimo per il superamento dell'esame deve essere maggiore o uguale a 15/30