

**ES1.** Scrivere un programma C che apre un file di testo "testo.txt" contenente parole ciascuna di al massimo 15 caratteri, separate da spazi. Il programma trova tutte le parole più lunghe di 4 caratteri e ripetute due volte di seguito nel testo e le stampa a video. Gestire opportunamente tutti gli eventuali errori.

Ad esempio, se il file "testo.txt" contiene il testo:

bisogna studiare passo passo e adagio adagio per ottenere un voto alto alto

Il programma visualizzerà in output:

passo adagio

Come test, forniamo i file testo1.txt, testo2.txt, testo3.txt, i cui output corrispondenti sono qui elencati:

testo1.txt -> passo adagio

testo2.txt -> quando occupato

testo3.txt -> quattro cinque sette

**ES2.** Si considerino le seguenti strutture dati che permettono di rappresentare un componente di un mobile Ikea (per esempio: vite, pannello, anta, ...), un magazzino di componenti e le istruzioni per costruire un mobile (per esempio: cucina, armadio, ...).

```
#define MAXS 100
#define MAXC 200

typedef struct {
    int codice;
    char nome[MAXS+1];
    int quantita;
} componente_t;

typedef struct {
    char nome[MAXS+1];
    char indirizzo[MAXS+1];
    componente_t componenti[MAXC];
    int n_componenti;
} magazzino_t;

typedef struct {
    int codice;
    char descrizione[MAXS+1];
    float costo;
    componente_t componenti[MAXC];
    int n_componenti;
} istruzioni_t;
```

1. Scrivere un sottoprogramma C che riceve come parametri un magazzino di componenti MAG, un manuale di istruzioni MAN ed un numero intero X. Il sottoprogramma verifica se è possibile costruire X mobili relativi al manuale MAN ricevuto come parametro e prelevando i componenti dal magazzino MAG; in caso affermativo il sottoprogramma restituisce 1 altrimenti 0.
2. Scrivere un programma che riceve come argomenti i nomi di due file binari, contenenti rispettivamente i dati di un libretto di istruzioni (una variabile di tipo istruzioni\_t) e di un magazzino (una variabile di tipo magazzino\_t), ed un numero N. Il programma legge dai file i dati salvandoli in due variabili di tipo istruzioni\_t e magazzino\_t, ne visualizza i due contenuti ed esegue la chiamata al sottoprogramma precedentemente definito e stampa a video il risultato dell'elaborazione. Il programma deve gestire opportunamente tutti i possibili casi di errore.

Come test, forniamo i file istrN.bin, magN.bin, con N=1,2,3. Qui di seguito mostriamo gli output omettendo le stampe relative a codici, descrizioni e quantità, lasciando quindi solo l'output del sottoprogramma.

./ikea istr1.bin mag1.bin 1

Output: 1

./ikea istr1.bin mag1.bin 2

Output: 0

./ikea istr2.bin mag2.bin 0

Output: 1

./ikea istr2.bin mag2.bin 1

Output: 0

./ikea istr3.bin mag3.bin 3

Output: 1

./ikea istr3.bin mag3.bin 4

Output: 0

./ikea istr2.bin mag3.bin 0

Output: 0

**ES3.** Scrivere un sottoprogramma che riceve come parametri due interi n e b, con b sicuramente compreso tra 1 e 9 estremi inclusi. Il sottoprogramma visualizza tutti i numeri, in ordine crescente, che possono essere generati in base b su n cifre.

Per esempio se il sottoprogramma riceve in ingresso i valori n=3 e b=2, stampa a video "000, 001, 010, 011, 100, 101, 110, 111".

Scrivere un programma che chiede all'utente due valori interi n e b ed invoca il sottoprogramma sopra definito.

NOTA: nel caso si realizzi un sottoprogramma ricorsivo è possibile passare anche ulteriori parametri ritenuti necessari.

**ES4.** Considerando la struttura dati ed i sottoprogrammi contenuti nei file lista.c e lista.h, implementare un sottoprogramma che riceve come parametri una lista dinamica di interi l1 ed un array di interi sel. Assumendo che sel contenga solo valori 0 e 1 e che sia di dimensione pari a l1, il sottoprogramma costruisce e restituisce una nuova lista l2 che contiene solo gli elementi di l1 che si trovano nella lista in una posizione corrispondente agli 1 del vettore sel.

Esempio: se il sottoprogramma riceve in ingresso

```
l1 = 3 -> 5 -> 8 -> 9 -> 12
```

```
vet = [0, 1, 0, 1, 1]
```

dovrà ritornare la lista

```
l2 = 5 -> 9 -> 12
```

Scrivere un programma che chiede all'utente

1. una sequenza di lunghezza indefinita di numeri interi chiesta all'utente e terminata dal valore 0 (che non fa parte della sequenza) per popolare una lista dinamica
2. un array di valori interi 0 e 1 di lunghezza pari a quella della lista (array da allocare dinamicamente!)

Il programma invoca il sottoprogramma sopra definito, visualizza la lista restituita e libera tutta la memoria allocata.

Gestire opportunamente tutti gli eventuali errori.