

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * tema d'esame (2017-09-13)  
 */
```

```
/* Si implementino due sottoprogrammi C.  
 Il primo sottoprogramma, chiamato List2File, esegue quanto segue:  
 1. riceve in ingresso la testa di una lista dinamica, ciascun elemento della quale contiene (come campo dati) una struct  
 composta da un intero e un char;  
 2. crea su disco fisso un file binario chiamato "Contents.dat";  
 3. memorizza nel file Contents.dat tutte le struct contenute nella lista dinamica, nello stesso ordine;  
 4. chiude il file binario.
```

```
Il secondo sottoprogramma, chiamato File2List, esegue quanto segue:  
 1. apre da disco fisso il file binario chiamato "Contents.dat";  
 2. legge le struct contenute nel file e le memorizza (nello stesso ordine oppure in ordine inverso) in una lista dinamica i  
 cui elementi hanno lo stesso tipo di quella che il sottoprogramma List2File riceve in ingresso;  
 3. chiude il file;  
 4. restituisce al programma chiamante la testa della lista.
```

Oltre ad implementare i due sottoprogrammi, lo studente definisca i tipi di dato necessari e specifici quali direttive #include sono necessarie. Non Ã" richiesto gestire i casi in cui le operazioni di scrittura e lettura file non vanno a buon fine.

```
*/  
#include <stdio.h>  
#include <stdlib.h>  
#define NOMEFILE "./file-es12_04/Contents.dat"  
  
/* #define DEBUG */  
/* commentare via questa direttiva a debugging completato */  
  
typedef struct  
{  
    char C;  
    int I;  
} Tipodato;  
  
typedef struct EL
```

```
{
    Tipodato Dato;
    struct EL *Prossimo;
} ElemLista;

void List2File(ElemLista *Testalista)
{
    FILE* Stream;
    /* stream nel quale scrivere i dati */
    ElemLista *ElemCorr;
    /* punta all'elemento corrente della lista */

    #ifdef DEBUG
        printf("\n###DEBUG: inizio List2File");
    #endif

    Stream = fopen(NOMEFILE, "wb");

    ElemCorr = Testalista;

    while (ElemCorr != NULL)
    {
        fwrite(&(ElemCorr->Dato), sizeof(Tipodato), 1, Stream);
        ElemCorr = ElemCorr->Prossimo;
    }

    fclose(Stream);
}

ElemLista *File2List()
{
    FILE* Stream;
    /* stream dal quale leggere i dati */
    Tipodato DatoCorr;
    /* dato letto da file, da inserire nella lista */
    ElemLista *Testa, *NuovoElem;
    /* testa di una lista dinamica di prova, elemento corrente della lista
    durante la sua creazione */

    #ifdef DEBUG
```



```
ElemCorr = ElemCorr->Prossimo;
}
}
int main()
{
    int Contadati;
    /* usato per generare dati di prova */
    ElemLista *Testa, *NuovoElem;
    /* testa di una lista dinamica di prova, elemento corrente della lista
    durante la sua creazione */
    Testa = NULL;

    /** genera una lista di prova **/
    for (Contadati = 0; Contadati < 20; ++Contadati)
    {
        NuovoElem = malloc(sizeof(ElemLista));
        NuovoElem->Dato.C = 'a' + Contadati;
        NuovoElem->Dato.I = Contadati+1;
        NuovoElem->Prossimo = Testa;
        Testa = NuovoElem;
    }

    printf("\nlista originale:");
    Stampalista(Testa);

    List2File(Testa);
    Testa = File2List();

    printf("\n\nlista finale:");
    Stampalista(Testa);
    printf("\n\n");
    return 0;
}
```