ESERCIZIO (Domanda)
Supponiamo che sia int X = 1;
la chiamata di funzione

```
fprintf(file,"%d", X);
```

a. emette sul file un byte che corrisponde al codice ASCII del carattere 1;
b. emette sul file (per interi su 16 bit) due byte che rappresentano 1 in notazione binaria;
c. emette sul file un byte che corrisponde al codice ASCII del carattere X.

Soluzione
La risposta è a.

ESERCIZIO (sintesi)
Dato un file di testo **mesi.txt**, si supponga che sia costituito da righe ciascuna contenente una stringa (nome del mese) ed un intero (numero di giorni). Ad esempio:

```
gennaio   31
febbraio  28
marzo     31
aprile    30
```

Si stampino a video i nomi dei mesi che hanno 31 giorni.

Soluzione

```
#include <stdio.h>
#include <stdlib.h>

main() {
    int giorni; FILE* f;char nome[20];

    if ((f=fopen("mesi.txt", "r"))==NULL) {
        printf("Il file non esiste!");
        exit(1);
    }

    while(fscanf(f,"%s%d\n",     nome, &giorni) != EOF)
        if (giorni == 31)
          printf("%s\n", nome);

    fclose(f);
}
```

Esercizio (sintesi)

Dato un file di testo **`estratti.txt`**, si supponga che sia costituito da righe ciascuna contenente una ruota (nome della ruota del lotto) ed un intero (numero estratto). Ad esempio:

```
napoli    31
genova    28
napoli    60
```

Si scriva un programma C che prenda in ingresso il nome di una ruota e stampi tutti i numeri estratti su quella ruota.

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

main() {
    int numero; FILE* f;
    char ruota[20],miaruota[20];

    printf("Inserire ruota: ");
    scanf("%s",miaruota);

    if ((f=fopen("estratti.txt", "r"))==NULL) {
        printf("Il file non esiste!");
      exit(1);
    }
    while(fscanf(f,"%s %d\n",    ruota, &numero) != EOF)
        if (strcmp(ruota,miaruota)==0)
          printf("%d\t", numero);
    fclose(f);
}
```

Esercizio (sintesi)

Si vuole realizzare un programma che data da input una sequenza di N parole (di, al massimo, 20 caratteri ciascuna), li memorizzi in una struttura dati dinamica e poi stampi la loro lunghezza.

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef char parola[20];

main()
{    parola  *p;
     int i,  N;

     printf("Quante parole? ");
     scanf("%d", &N);

     /* allocazione del vettore */
     p=(parola *)malloc(N*sizeof(parola));

     /* lettura della sequenza */
     for(i=0; i<N; i++) scanf("%s", p[i]);
     for(i=0; i<N; i++) printf("\n%d", strlen(p[i]));

     free(p); /* deallocazione */
}
```

Esercizio (sintesi, 3 esercizi in uno)
Sia dato il file di testo "*dati.txt*" contenente i dati relativi agli studenti immatricolati al primo anno della Facoltà di Ingegneria.

In particolare, le informazioni sono memorizzate nel file "*dati.txt*" come segue:

ognuna delle linee del file contiene i dati relativi ad un nuovo studente ed in particolare:

- **Matricola**: un intero che indica il numero di matricola dello studente;
- **CdL**: un intero che indica il corso di laurea (CdL) dello studente (es. 2145);

Sia dato un secondo file binario "*indirizzi.bin*" che contiene, invece, l'indirizzo di ogni studente, e in particolare:

- **Matricola**: il numero di matricola dello studente;
- **Nome**: il nome dello studente;
- **Cognome**: il cognome dello studente;
- **Via**: una stringa che riporta la via di residenza dello studente;
- **Città**: una stringa che riporta la città di residenza dello studente;
- **CAP**: un intero che rappresenta il codice di avviamento postale dello studente.

Si scriva un programma in linguaggio C che:

1. A partire dai file "dati.txt" e "indirizzi.bin" costruisca una tabella T contenente, per ogni studente, Matricola, Nome, Cognome, Via, Città, CAP e CdL.

2. A partire dalla tabella T, e dato da input un intero C che rappresenta un CdL, stampi la percentuale di studenti (rispetto al numero totale delle matricole) iscritti al corso C. [Ad esempio, se il numero totale delle matricole e` 1000, e quello degli studenti iscritti a C e` 200, il programma stamperà "20%"]

3. Scriva su un terzo file di testo "bologna.txt", nome, cognome e numero di matricola di tutti gli studenti che abitano a Bologna.

Soluzione:

```
#include <stdio.h>
#include <string.h>
/* tipi di dato */
typedef struct {
      unsigned int matr;
      unsigned CDL;
}dati;

typedef struct {
      unsigned int matr;
      char nome[20];
      char cognome[30];
      char via[30];
```

```
      char citta[30];
      unsigned int CAP;
} indirizzo;

typedef struct {
      unsigned int matr;
      char nome[20];
      char cognome[30];
      char via[30];
      char citta[30];
      unsigned int CAP;
      unsigned int CDL;
} elemento;

typedef elemento tabella[10];

elemento riempiel( dati d, indirizzo i);

/*  le seguenti funzioni servono solo per predisporre e visualizzare il
file di  indirizzi:*/

void creafile(char *b);
void vedifile(char *b);
/* fine funzioni file */



main(){
      dati D;
      indirizzo I;
      elemento E;
      tabella T;
      FILE *f1, *f2;
      int i, trovato,ins=0, totC;
      unsigned int C;

      /*non necessario: creaz. del file binario */
      printf("creare il file (0/1)??");
      scanf("%d", &i);
      if (i==1)
            creafile("indirizzi.bin");
      else vedifile("indirizzi.bin");


/*domanda 1: costruzione della tabella */

      f1=fopen("dati.txt", "r");
      f2=fopen("indirizzi.bin", "rb");


      while (fscanf(f1,"%u%u", &D.matr, &D.CDL)>0){
            trovato=0;
            rewind(f2);
```

```
            while(fread(&I,sizeof(indirizzo),1,f2)>0 && !trovato)
                  if(I.matr==D.matr){ /*trovato l'indirizzo dello stud. D*/
                        trovato=1;
                        E=riempiel(D, I);
                        T[ins]=E;
                        ins++;
                  }
      }

      fclose(f1);
      fclose(f2);

/*domanda 2: stampa della percentuale degli iscritti a un dato corso */
      printf("Inserire il corso C: ");
      scanf("%u", &C);
      totC=0;
      for(i=0; i<ins; i++)
            if(T[i].CDL==C)
                  totC++;

      printf("\n Iscritti al corso %u: %f \%\n", C, (float)totC*100/ins);

/*domanda 3: scrittura di “bologna.txt" */
      f1=fopen("bologna.txt", "w");
      for (i=0; i<ins; i++)
         if (strcmp("bologna", T[i].citta)==0)
            fprintf(f1, "%s %s %u\n", T[i].nome, T[i].cognome, T[i].matr);
      fclose(f1);
}



elemento riempiel(dati d, indirizzo i){
      elemento e;
      /*copia in e il contenuto di d e di i*/
      e.matr=d.matr;
      e.CDL=d.CDL;
      strcpy(e.nome, i.nome);
      strcpy(e.cognome, i.cognome);
      strcpy(e.via, i.via);
      strcpy(e.citta, i.citta);
      e.CAP=i.CAP;
      return e;
}



void creafile(char *v){
      FILE *f; indirizzo e;int fine=0;

      f=fopen(v, "wb");
      printf("creazione di %s...\n", v);
```

```
      while (!fine){
            printf("matricola");
            scanf("%u", &e.matr);
            printf("\nCAP ? " );
            scanf("%u", &e.CAP);
            printf("\nCognome ? ");
            scanf("%s", &e.cognome);
            printf("\nNome ? ");
            scanf("%s", &e.nome);
            printf("\nCitta`? ");
            scanf("%s", &e.citta);
            printf("\nVia ? ");
            scanf("%s", &e.via);
            fflush(stdin);
            fwrite(&e, sizeof(indirizzo), 1, f);
            printf("\nFine (SI=1, NO=0) ? ");
            scanf("%d", &fine);
      }
      fclose(f);
}



void vedifile(char *v){
      FILE *f; indirizzo e;int fine=0;

      f=fopen(v, "rb");
      printf("Lettura di %s:\n", v);

      fread(&e, sizeof(indirizzo), 1, f);
      while (!feof(f)){
            printf("%u\t", e.matr);
            printf("%s\t", e.cognome);
            printf("%s\t", e.nome);
            printf("%s\t", e.via);
            printf("%s\n", e.citta);
            printf("%u\t", e.CAP);
            fread(&e, sizeof(indirizzo), 1, f);
      }
      fclose(f);
}
```

Esercizio (sintesi)

Dato un file binario **mesi.dat**, si supponga che contenga (in rappresentazione interna) strutture così configurate: una stringa (nome del mese) ed un intero (numero di giorni). Ad esempio:

```
gennaio   31
febbraio  28
marzo     31
aprile    30
```

Si memorizzi il contenuto del file in un vettore di strutture e si stampino a video i nomi dei mesi che hanno 31 giorni.

Soluzione

```
#include <stdio.h>
#include <stdlib.h>

main() {
    int i;
    struct mese {int giorni; char nome[20];} v[12];
    FILE* f;

    if ((f=fopen("mesi.dat", "rb"))==NULL) {
      printf("Il file non esiste!");
      exit(1);
    }

    while(fread(&v[i],sizeof(struct mese),1,f)>0){
      if (v[i].giorni == 31)
            printf("%s\n", v[i].nome);
      i++;
    }

    fclose(f);
}
```

Esercizio (sintesi) su liste di interi

Un file di testo (TEMP.DAT) contiene i dati relativi alle medie di tutti gli studenti che devono accedere ad una sessione di laurea.

Si realizzi un programma C che:

a) Costruisca in memoria centrale una lista che memorizzi, in modo ordinato crescente tali medie (intere) e la stampi.

b) Letti due valori interi da console min e max, utilizzando la lista, visualizzi il valore delle medie comprese fra min e max ed un opportuno messaggio se non ne esistono.

Possibile contenuto di TEMP.DAT

```
90
100
98
111
88
87
```

intervallo 88 101

stampa

```
88 90 98 100
```

È possibile utilizzare ***librerie C*** (ad esempio per stringhe) e si possono utilizzare le operazioni primitive presentate a lezione sull'ADT lista.

Possibile Schema di Soluzione

```
/* PROGRAMMA PRINCIPALE - file main.c */
#include <stdio.h>
#include <stdlib.h>
#include "list.h"

main(){
     element   e, min, max;
     list L=emptylist();;
     FILE *f1;
     int i;

     /* DOMANDA a */
     f1 = fopen("TEMP.DAT", "r");
     while (fscanf(f1, “%d, &e)!=EOF)
          L=insord(e, L);
     showlist(L);
     fclose(f1);

     /* DOMANDA b */
     printf(“Dammi i due estremi : “);
     scanf(“%d”, &min, &max);

     while (!empty(L)&& (head(L)<min))L=tail(L);

     if (empty(L)) printf(“nessun valore”);
     else {
          while (!empty(L) && (head(L)<max)){
               printf(“%d”, head(L));
               L=tail(L)
          };
     }
}
```

Esercizio (sintesi) su file e liste di interi

Un file di testo ARCHIVIO.TXT contiene i dati (***primo autore***, ***titolo***, ***numero*** di copie ***possedute***, ***numero*** di copie in ***prestito***) relativi ai differenti volumi conservati presso una biblioteca. Più precisamente, ogni riga del file contiene nell'ordine, separati da uno spazio bianco:

- ***autore*** (non più di 20 caratteri senza spazi intermedi);
- ***titolo*** (non più di 50 caratteri senza spazi intermedi);
- ***numero_possedute*** (da leggersi come intero);
- ***numero_prestito*** (da leggersi come intero).

Si realizzi un programma C che:

1. Legga il contenuto di ARCHIVIO.TXT e costruisca in memoria centrale un ***vettore V*** di strutture corrispondenti (si supponga che il file ARCHIVIO.TXT non possa contenere più di 30 righe). Si stampi a video il contenuto del vettore.

2. A partire da V, costruisca una ***lista L di interi*** contenente per ciascun volume il ***numero di copie disponibili*** nella biblioteca, ovvero la differenza fra il numero di copie possedute e il numero di copie in prestito. Si stampi a video il contenuto della lista L.

3. Utilizzando L per ottenere la somma delle copie disponibili e V per la somma delle copie possedute, calcoli ***il rapporto fra volumi disponibili e volumi posseduti***.

Oppure

3bis. Utilizzando la lista di interi L, stampi il numero di riga di ARCHIVIO.TXT relativo al volume con più copie disponibili. In caso di più volumi con pari numero di copie disponibili, qualunque riga relativa a questi ultimi è considerata una risposta corretta.

Ad esempio: contenuto di ARCHIVIO.TXT

```
Salinger IlGiovaneHolden 10 8
Wallace InfiniteJest 12 3
Carver Cattedrale 12 12
Baricco Seta 6 0
Hornby ComeDiventare 9 9
Sartre LaNausea 3 1
Robbins NaturaMorta 7 7
```

Stampa di L:

```
[2, 9, 0, 6, 0, 2, 0]
```

È possibile utilizzare ***librerie C*** (ad esempio per stringhe) e si devono utilizzare le librerie sulle liste presentate a lezione. Qualunque ***libreria utente*** addizionale eventualmente utilizzata va riportata nello svolgimento e consegnata.

Possibile Schema di Soluzione

Suddivido il programma nei seguenti file:

list.c	funzioni di libreria per la gestione di liste
list.h	header file associato a list.c
element.h	contiene la dichiarazione di element
mainLibri.c	contiene il programma principale

```
/* PROGRAMMA PRINCIPALE - file mainLibri.c */
#include <stdio.h>
#include <stdlib.h>
#include "list.h"
#define MAX 20

typedef struct{
      char autore[20];
      char titolo[50];
      int possedute;
      int prestito;
} volume;

main() {
     volume e; list L,L1; FILE *f;
     volume V[MAX]; int elementi=0,i,pos,max;
     int somma_possedute, somma_disponibili;
     L=emptylist();


/* DOMANDA 1 */
     f = fopen("ARCHIVIO.TXT", "r");
     if (f==NULL) {
          printf("Impossibile aprire file di ingresso");
          exit(1); } /* se non riesce a creare il file
               visualizza messaggio di errore ed esce */

while (fscanf(f,"%s%s%d%d\n",e.autore,
               e.titolo, &e.possedute, &e.prestito)>0)
                         V[elementi++] = e;
     fclose(f);
for (i=0; i<elementi; i++)
          printf("Volume %d: %s\t%s\t%d\t%d\n",i,V[i].autore,
               V[i].titolo,V[i].possedute,V[i].prestito);
```

```
/* DOMANDA 2 */
     for (i=0; i<elementi; i++)
               L = cons(V[i].possedute-V[i].prestito,L);
     showlist(L);
/* in che ordine viene stampata la lista??? */




/* DOMANDA 3 */
     for (i=0; i<elementi; i++)
               somma_possedute += V[i].possedute;
     L1=L;
     while (!empty(L1))
               { somma_disponibili += head(L1);
                 L1=tail(L1);
               }
     printf("Rapporto disponibili/possedute = %f\n",
               (float)somma_disponibili/somma_possedute);


/* DOMANDA 3bis */
     i=0; max=-1;
     while (!empty(L)){
          if (head(L)>max) {
               max = head(L);
               pos=i;
          }
          L=tail(L);
          i++;
     }
     printf("Volume con più copie disponibili: %d",
                    elementi-pos-1);
}
```

Esercizio (domanda)
Di che tipo è la variabile **f** restituita dalla funzione fopen (se ne mostri anche la dichiarazione):

```
f = fopen("esame.txt", "r");
```

Soluzione
La variabile **f** è un puntatore a file

```
FILE* f;
f = fopen("esame.txt", "r");
```

Esercizio (domanda)
Si individuino analogie e differenze tra le funzioni **fscanf** e **scanf**.

Soluzione
Entrambe sono funzioni che permettono di leggere dati rispettivamente da un file di testo (passato come parametro alla fscanf) e la seconda da standard input. Entrambe hanno come parametri una stringa di formato e i valori da leggere, ma la fscanf() necessita di un parametro aggiuntivo costituito dal puntatore a FILE su cui svolgere le operazioni di lettura.

Esercizio (domanda)
Siano date due stringhe char s1[]="Pippo", s2[20];
Se si scrive s1=s2; che cosa succede?

A. Tutto il contenuto di s2 viene copiato in s1
B. Si ottiene un errore di compilazione
C. Il primo elemento di s2 viene ricopiato nel primo elemento di s1

Soluzione
B. Si ottiene un errore di compilazione.

Esercizio (domanda)
Supponiamo che sia **int x = 11;**

Che differenza c'è fra

```
fprintf(file,"%d", x);
```

e

```
fwrite(&x, sizeof(int), 1, file);
```

se il codice ASCII del carattere 1 è 00110001?
Soluzione
fprintf emette due byte, ciascuno corrispondente al codice ASCII del carattere 1:
00110001 00110001
fwrite emette (su 16 bit) due byte che rappresentano 11 in notazione binaria:
00000000 00001011