

Esercizio di Progetto :

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;
- **Citta`:** una stringa che riporta la citta` 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, Citta`, 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 stampera` "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) /*ho 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 corso dato*/
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 creafire(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 di Analisi

Dato il seguente programma:

```
#include <stdio.h>
#define DIM 6

int p(int a)
{ if (a%2==0)
  return 0;
  else return a+1;
}

int f(int *a, int b)
{
  if (a[b]!=0)
    return a[b]=5;
  else return p(b+1)+b;
}

main()
{ int A[DIM]={0,0,0,0,0,0};
  int i;
  for(i=0; i<DIM; i+=2)
    A[i]=i;
  printf("%d\n", f(A,0));
  for(i=0; i<DIM; i++)
    printf("%d\t",A[i]);
}
```

Si indichino, nel giusto ordine, i valori stampati dal programma, motivando la risposta data.

Output:

2
0 0 2 0 4 0

... discussione dettagliata di quello che succede...

Esercizio di Progetto di una funzione ricorsiva:

Si consideri la seguente funzione F la cui specifica è data in modo ricorsivo (si supponga N intero):

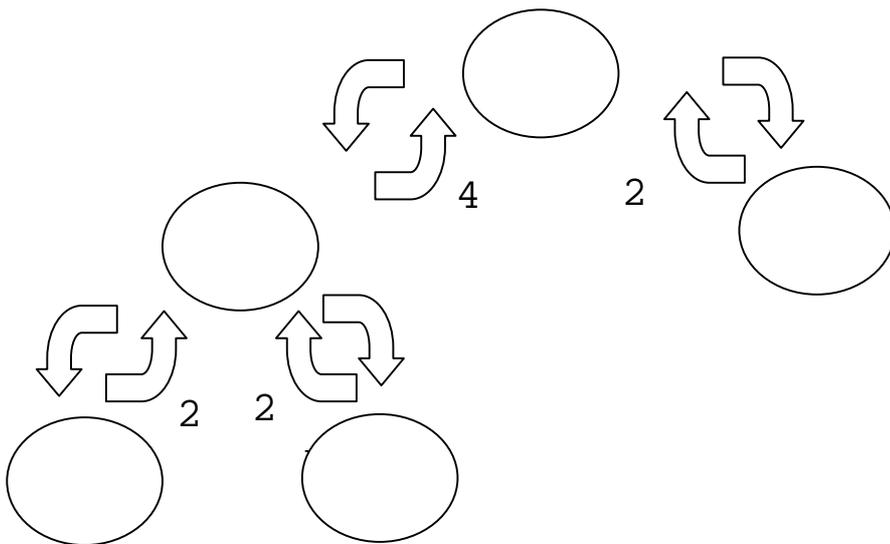
$F(N)$ = restituisce 2 se $N \leq 0$,

$F(N-2) * F(N-3)$, altrimenti

a) Si scriva il risultato della funzione quando chiamata con $N=3$ e si mostrino i valori intermedi assunti da N;

b) Si scriva la funzione C che realizzerebbe tale specifica

Soluzione:



a) Sequenza di attivazioni: **F(3)** restituisce il valore **8**

Valori assunti da N: 3 1 -1 -2 0

b) Codifica in C della Funzione:

```
int F(int N)
{if (N<=0) return 2;
 else return F(N-2)*F(N-3);
}
```