

**PROVA SCRITTA DEL MODULO DI  
ELEMENTI DI INFORMATICA  
CORSO DI LAUREA IN INGEGNERIA BIOMEDICA  
21 febbraio 2019**

**MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI**

**ESERCIZIO 1 (4 punti)**

Indicare il minimo numero di bit necessario per rappresentare il valore intero senza segno 4096.

**ESERCIZIO 2 (4 punti)**

Spiegare in modo chiaro e sintetico in cosa consiste il modulo CPU di un calcolatore elettronico.

**ESERCIZIO 3 (25 punti)**

(7 punti) Un vostro amico ha ideato un'app di sicuro successo: crea una rete di contatti reciprocamente interessati ad una relazione sentimentale. Chi si iscrive alla rete mediante questa app, attivandola, riceverà a fine giornata l'elenco di tutte le persone giunte in prossimità del vostro cellulare durante la giornata che sono interessate ad una relazione sentimentale, ovviamente a loro volta iscritte all'app. Il formato di questo elenco è un file "papabili.txt", per ogni riga del quale è riportato in sequenza il nome della persona, il colore dei capelli ("Biondo"/"Castano"/"Scuro"), l'età, l'altezza e l'indirizzo e-mail.

Tuttavia, dopo un primo periodo di collaudo, si scopre che gli elenchi giornalieri che vengono ricevuti sono lunghissimi (tante persone si sono iscritte!), così che risulta difficilissimo scegliere la persona da contattare. Così il vostro amico pensa che sia il caso di aggiungere una speciale opzione per la quale ognuno possa inserire il colore dei capelli, l'età minima e l'altezza massima della persona alla quale si potrebbe essere potenzialmente interessati. Il sistema farebbe quindi un censimento automatico andando a ridurre il numero di persone a partire dall'elenco presente in "papabili.txt" e scrivendo l'elenco ridotto in un nuovo file "papabilissimi.txt", dove però verrebbe scritto soltanto il nome e l'indirizzo e-mail nel caso venissero soddisfatti tutti e tre i criteri. Tutte le persone non soddisfacenti tutti i criteri verrebbero escluse. Infine, il sistema segnalerebbe all'iscritto il messaggio "Oggi ci sono <X> papabili per te" oppure "Oggi non ci sono papabili per te, mi spiace!", a fine elaborazione, su video, con <X> numero di papabili calcolati in base ai criteri.

Poiché però è a corto di programmatori, chiede a voi di scrivere il codice C della nuovissima funzionalità, in modo che possa integrarla nella sua app innovativa. Accettate e, dopo aver negoziato il compenso, vi mettete a lavoro. Decidete che:

- L'inserimento dei desiderata avvenga in una struttura caratterizzata da una stringa di 10 caratteri per memorizzare il colore dei capelli desiderato, e due interi per memorizzare l'età minima e l'altezza massima. Chiamate tale struttura `Feature`.
- L'inserimento delle persone lette da file avvenga in una lista concatenata di strutture, chiamata appunto `ListaPersone`, il cui elemento di lista è di tipo `Persona`, così caratterizzato: due stringhe di 50 caratteri, una finalizzata alla memorizzazione del nome e dell'indirizzo e-mail della persona, e una variabile di tipo `Feature` dove memorizzare il colore dei capelli, l'età e l'altezza.

Decidete poi di implementare le seguenti funzioni:

- (3 punti) `inserisciDesiderata()`: legge da tastiera il colore dei capelli, l'età minima e l'altezza massima desiderate e restituisce una variabile di tipo `Feature` contenente i valori desiderati.
- (7 punti) `leggiElenco(nomefile)`: legge da file di nome `nomefile`, formattato come indicato sopra, ciascuna riga e la deposita in un puntatore a lista concatenata di tipo `ListaPersone`, restituito in uscita.
- (5 punti) `selezionaPapabile(candidato, desiderata)`: riceve in ingresso due variabili di tipo `Feature`, il primo relativo ai valori di colore dei capelli, età ed altezza letti precedentemente dal file "papabili.txt", il secondo relativo ai valori desiderati, e restituisce 1 se tutti i criteri di colore di capelli, età minima e altezza massima presenti nel parametro `desiderata` corrispondono a quelli presenti nel parametro `candidato`, 0 altrimenti.
- (3 punti) `scriviPapabilissimo(f, candidato)`: scrive su file `f`, già aperto in modalità append, i valori relativi al nome ed all'indirizzo presenti nella variabile `candidato` di tipo `Persona` in un'unica riga.

**Nota:** la gestione della lista concatenata avviene attraverso le funzioni `CONS`, `TAIL`, `HEAD` e `ISEMPTY` che si suppongono già implementate.

### Soluzione dell'esercizio 1.

Poiché  $4096=2^{12}$ , per rappresentare tale valore occorrono almeno 13 bit (12 bit a zero e il 13simo, il più significativo, ad 1).

### Soluzione dell'esercizio 2.

V. dispense del corso.

### Soluzione dell'esercizio 3.

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

/* Tipi fondamentali per la risoluzione dell'esercizio, come da testo */
typedef struct
{
    char coloreDeiCapelli[50];
    int eta, altezza;
} Feature;

typedef struct
{
    char nome[50], indirizzo[50];
    Feature dati;
} Persona;

typedef struct ElementoDiLista
{
    Persona p;
    struct ElementoDiLista *successivo;
} ListaPersone;

/* Funzioni CONS, HEAD, TAIL, ISEMPY */
ListaPersone *CONS(ListaPersone* l, Persona p)
{
    ListaPersone *nl;
    nl=(ListaPersone*)malloc(sizeof(ListaPersone));

    nl->successivo=l;
    nl->p=p;

    return nl;
}

Persona HEAD(ListaPersone *l)
{
    return l->p;
}
```

```

ListaPersone* TAIL(ListaPersone *l)
{
    ListaPersone* nl;
    nl=l->successivo;
    free(l);
    return nl;
}

int ISEMPY(ListaPersone *l)
{
    return l==NULL;
}

/* Funzioni richieste dal compito */
Feature inserisciDesiderata()
{
    Feature f;

    printf("Inserisci il colore dei capelli:\n");
    scanf("%s",f.coloreDeiCapelli);
    printf("Inserisci l\'eta\' minima:\n");
    scanf("%d",&f.eta);
    printf("Inserisci l\'altezza massima:\n");
    scanf("%d",&f.altezza);

    return f;
}

ListaPersone* leggiElenco(char* nomefile)
{
    FILE *f;
    ListaPersone *l=NULL;
    Persona p;
    Feature d;

    f=fopen(nomefile,"r");
    if(!f)
    {
        printf("\nFILE NON TROVATO\n");
        return NULL;
    }

    while (!feof(f))
    {
        fscanf(f,"%s",p.nome);
        fscanf(f,"%s %d %d",d.coloreDeiCapelli, &d.eta, &d.altezza);
        fscanf(f,"%s",p.indirizzo);
        p.dati=d;
        l=CONS(l,p);
    }
    fclose(f);

    return l;
}

```

```

int selezionaPapabile(Feature c, Feature d)
{
    int capelliOK, etaOK, altezzaOK;

    capelliOK=!strcmp(c.coloreDeiCapelli,d.coloreDeiCapelli);
    etaOK=c.eta>=d.eta;
    altezzaOK=c.altezza<=d.altezza;

    return capelliOK && etaOK && altezzaOK;
}

void scriviPapabilissimo(FILE *f, Persona p)
{
    fprintf(f, "%s %s\n",p.nome, p.indirizzo);
}

/* Funzione principale */
int main()
{
    ListaPersone *l;
    Persona p;
    Feature d;
    FILE *f;
    int n, papabile;

    d=inserisciDesiderata();
    l=leggiElenco("papabili.txt");

    n=0;
    while(!ISEMPTY(l))
    {
        p=HEAD(l);
        papabile=selezionaPapabile(p.dati,d);
        if (papabile)
        {
            f=fopen("papabilissimi.txt","a");
            scriviPapabilissimo(f,p);
            fclose(f);
        }
        n=n+papabile;
        l=TAIL(l);
    }

    if (n)
        printf("Oggi ci sono %d papabili per te!\n",n);
    else
        printf("Oggi non ci sono papabili per te, mi spiace!\n");

    free(l);
    return 0;
}

```