

PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSI DI LAUREA IN
INGEGNERIA BIOMEDICA (indirizzo industriale) E INGEGNERIA ELETTRICA
INGEGNERIA CHIMICA E INGEGNERIA MECCANICA
19/10/2016

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

ESERCIZIO 1 (5 punti)

Scrivere la tabella di verità della seguente funzione booleana:

$$F = \bar{A}B + B + A\bar{B}C$$

Dedurre la funzione semplificata dall'osservazione della tabella o con le leggi dell'algebra booleana.

ESERCIZIO 2 (5 punti)

Spiegare in modo chiaro e sintetico la struttura di un moderno sistema operativo.

ESERCIZIO 3 (solo 5 crediti: 5 punti)

(3 punti) Spiegare in modo chiaro e sintetico il modulo di memoria di un calcolatore elettronico.

(2 punti) Si consideri un calcolatore con parole da 32 bit ed un indirizzamento a 16 bit. Esprimere la dimensione della memoria in KB.

ESERCIZIO 3 (solo 6 crediti: 5 punti)

Siano date le istanze di relazione:

Docenti

Nome_D	Cognome_D	Matricola	Corsi_tenuti
Gian Luca	Marcialis	1	2
Giulio	Concas	2	4
Michele	Marchesi	3	5
Giuliano	Armano	4	3
Giorgio	Fumera	5	2
Fabio	Roli	6	5

Allievi

Matricola	Nome	Cognome	Max_voto	Docente
1	Giancarla	Ferrai	30	3
2	Marco	Pistis	30	2
3	Maria	Desogus	18	1
4	Silvia	Ferrai	30	1
5	Gianguido	Pistis	25	2

(2 punti) Spiegare in modo chiaro e sintetico cosa s'intende per schema di relazione.

(3 punti) Scrivere in SQL l'interrogazione: "Si trovi nome e cognome degli studenti che hanno conseguito un voto pari a 25 con docenti che hanno almeno 2 corsi".

ESERCIZIO 4 (18 punti)

(4 punti) Scrivere un programma nel linguaggio scelto tra Python e C che legga da file "input.txt" una sequenza costituita dalle seguenti triple: una stringa indicante il nome di una persona, e due interi indicanti l'altezza in centimetri ed il peso in kg.

Il programma scriva in un file "output.txt" il nome della persona con il peso inferiore, il relativo peso, ed il nome della persona con l'altezza maggiore, e la relativa altezza. Se più persone dovessero presentare lo stesso peso minimo e/o la stessa altezza massima, il programma scriva solo la prima persona identificata nella sequenza.

Esempio:

Input.txt
Pippo 120 50
Pluto 100 80
Paperino 140 35
Paperina 190 90
Topolino 80 50

Output.txt
Paperino 35
Paperina 190

Solo Linguaggio C - Nello scrivere la soluzione, si utilizzi i seguenti tipi:

```
typedef struct
{
    char nome[100];
    int peso, altezza;
} tipo_persona;

typedef tipo_persona lista_persone[N];
```

Solo Linguaggio Python – Si memorizzino in una lista di liste formate ciascuna dai tre valori letti per riga i dati del file. Es. [“Pippo”,120,50],[“Pluto”, 100, 80]...

Si implementino inoltre le seguenti funzioni:

(3 punti) Funzione `leggi_persona`: legge da file una tripla e la memorizza in una variabile di tipo `tipo_persona`, restituendola in uscita.

Parametri in ingresso: identificatore di file (già aperto)

Parametro in uscita: la variabile `tipo_persona` (C), ovvero lista contenente la tripla letta (Python).

(4 punti) Funzione `trova_altezza_massima`: restituisce in uscita l'indice della componente (ovvero, della persona) avente la massima altezza.

Parametri in ingresso: (C) variabile di tipo `lista_persone` chiamata `l` e un intero indicante il numero di persone in `l`; (Python) lista di liste con tutte le terne lette.

Parametro in uscita: indice della componente del parametro in ingresso il cui campo coincidente con l'altezza è massimo.

(4 punti) Funzione `trova_peso_minimo`: restituisce in uscita l'indice della componente (ovvero, della persona) avente il minimo peso.

Parametri in ingresso: come per la precedente.

Parametro in uscita: indice della prima componente del parametro in ingresso coincidente col peso minimo.

(3 punti) Funzione `scrivi_risultato`: scrive nel file “output.txt” il nome della persona e il relativo peso corrispondente alla componente con indice `p` di `l`, e il nome della persona e la relativo altezza corrispondente alla componente con indice `a` di `l`.

Parametri in ingresso: variabile di tipo `lista_persone`, chiamata `l` (C), ovvero lista di liste (Python), un intero `p` ed un intero `a`.

Parametro in uscita: **nessuno**.

Soluzione dell'esercizio 1.

Sulla base della lettura della tabella di verità si deduce che F vale 1 se B=1 oppure se AC=1. Da cui $F=B+AC$. Allo stesso risultato si arriva con le leggi dell'algebra booleana, la cui applicazione è lasciata allo studente.

Soluzione dell'esercizio 3 (5 crediti)

Una memoria con parole di 32 bit, ovvero 4 byte, e 16 bit di indirizzamento è costituita da $4 \cdot 2^{16}$ byte = 256 KB.

Soluzione dell'esercizio 3 (6 crediti)

La query: "Si trovi nome e cognome degli studenti che hanno conseguito un voto massimo strettamente minore di 25 con docenti che hanno almeno 2 corsi" si scrive come segue:

```
SELECT NOME, COGNOME
FROM DOCENTI, ALLIEVI
WHERE DOCENTI.MATRICOLA=DOCENTE
AND MAX_VOTO=25
AND CORSI_TENUTI>=2
```

Soluzione dell'esercizio 4.

VERSIONE C:

```
/*Programma per la lettura di voti e docenti*/
#include <stdio.h>
#define N 100

/*Definizione dei tipi: includere sotto i tipi definiti nel testo*/

/*Prototipi delle funzioni*/
tipo_persona leggi_persona(FILE* fp);
int trova_altezza_massima(lista_personae l, int num_personae);
int trova_peso_minimo(lista_personae l, int num_personae);
void scrivi_risultato(lista_personae l, int p, int a);

int main()
{
    lista_personae l;
    int num_personae, p, a;
    FILE *fp;

    fp=fopen("input.txt","r");
    num_personae=0;
    while((!feof(fp)) && (i<N))
    {
        l[num_personae]=leggi_persona(fp);
        num_personae++;
    }
    fclose(fp);

    if(num_personae>0)
    {
        p=trova_peso_minimo(l,num_personae);
        a=trova_altezza_massima(l,num_personae);
        scrivi_risultato(l,p,a);
    }

    return 0;
}
```

Implementazione delle singole funzioni:

```
tipo_persona leggi_persona(FILE* fp)
{
    tipo_persona persona;

    fscanf(fp,"%s %d %d",&persona.nome[0],&(persona.altezza),&(persona.peso));

    return persona;
}

int trova_altezza_massima(lista_persone l, int num_persone)
{
    int i, a;

    a=0;
    for(i=1; i<num_persone; i++)
        if(l[i].altezza>l[a].altezza)
            a=i;

    return a;
}

int trova_peso_minimo(lista_persone l, int num_persone)
{
    int i, p;

    p=0;
    for(i=1; i<num_persone; i++)
        if(l[i].peso<l[p].peso)
            p=i;

    return p;
}

void scrivi_risultato(lista_persone l, int p, int a)
{
    FILE *fp;

    fp=fopen("output.txt","w");
    fprintf(fp,"%s %d\n%s %d\n",l[p].nome,l[p].peso,l[a].nome,l[a].altezza);
    fclose(fp);
}
```

VERSIONE PYTHON

```
def main():

    fp=open("input.txt","r")

    num_persone=0
    l=[]

    tupla=leggi_persona(fp)
    while (tupla!=[]):
        l=l+[tupla]
        tupla=leggi_persona(fp)
        num_persone=num_persone+1

    fp.close()

    if (num_persone>0):
        p=trova_peso_minimo(l,num_persone)
        a=trova_altezza_massima(l,num_persone)
        scrivi_risultato(l,p,a)
```

Implementazione delle singole funzioni:

```
def leggi_persona(fp):

    tupla=fp.readline()
    if tupla!="":
        tupla=tupla.split()
        tupla[1]=int(tupla[1])
        tupla[2]=int(tupla[2])
        return tupla

    return []

def trova_altezza_massima(l, num_persone):

    a=0
    i=1
    while i<num_persone:
        if l[i][1]>l[a][1]:
            a=i
        i=i+1

    return a

def trova_peso_minimo(l, num_persone):

    p=0
    i=1
    while i<num_persone:
        if (l[i][2]<l[p][2]):
            p=i
        i=i+1

    return p
```

```
def scrivi_risultato(l, p, a):  
    fp=open("output.txt","w")  
  
    stringa=l[p][0]+" "+str(l[p][2])+"\n"+l[a][0]+" "+str(l[a][1])+"\n"  
    fp.write(stringa)  
  
    fp.close()
```