

Informatica - Allievi Ambientali - Prova del 23 gennaio 2023

ATTENZIONE Per superare la prova occorre ottenere almeno 18 in ognuna delle 2 parti. Inoltre:

- Utilizzare solo i fogli messi a disposizione, scrivendo chiaramente sulla metà sinistra del fronte-spazio (**come da figura**): sulla prima riga prima il cognome e poi il nome (entrambi in stampatello), sulla riga seguente numero di matricola (o identificativo di persona) e data dell'appello, sulla riga seguente parte a cui si riferisce il foglio (1 Parte o 2 Parte) e firma.
- Consegnare le due parti su fogli separati.
- Per ogni esercizio indicare il numero dello stesso separando chiaramente le soluzioni con una linea orizzontale.
- Scrivere in modo **CHIARO e COMPRENSIBILE** a penna blu o nera o matita morbida (**max HB**). Non usare penne di altri colori o matite troppo dure, di difficile lettura.
- Consegnare solo i fogli protocollo contenenti le due parti, non il testo o la "brutta".

| | |
|-----------|-------|
| Cognome | Nome |
| Matricola | Data |
| Parte X | Firma |

Esercizio n° Y

.....

Esercizio n° Z

.....

...

Compiti non soddisfacenti queste semplici regole subiranno penalizzazioni.
Durante la prova non è permesso consultare libri o appunti, nè lasciare l'aula.

Parte I - Teoria [Totale punti 30]

Esercizio 1 - Funzioni logiche [5 punti]

Si consideri la funzione logica $f = (A \text{ AND } B) \text{ AND } (\text{NOT } (C \text{ OR } B))$. Si fornisca la tabella di verità, eseguendo tutti i passaggi.

Soluzione

| A | B | C | A AND B | (C OR B) | NOT (C OR B) | f |
|---|---|---|---------|----------|--------------|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Esercizio 2 - Macchina di Von Neumann [5 punti]

Si spieghi brevemente il ruolo del clock di sistema e se ne motivi la presenza in ogni CPU.

Soluzione

Vd libro di testo. Il clock di sistema indica quando è possibile eseguire la prossima istruzione macchina. È necessario perché la transizione di stato da 0 a 1 richiede del tempo e non può essere istantanea.

Esercizio 3 - SQL

Si progetti in SQL lo schema di una base di dati per la gestione di dati relativi a rilevamenti effettuati in centraline per il controllo dell'aria. Per ogni rilevamento si devono memorizzare l'identificativo della centralina, la data, la temperatura, e i valori di PM10 e PM2.5. Per ogni centralina sono memorizzati anche l'indirizzo in cui si trova, le coordinate geografiche latitudine e longitudine, la sigla della provincia relativa, ed il numero di telefono del responsabile. **[5 punti]**

Si progettino due query, la prima per ottenere tutti i dati delle rilevazioni fatte in provincia di Milano nel dicembre 2022 in cui il PM10 ha superato la soglia di $55 \mu\text{g}/\text{m}^3$, la seconda per ritornare l'identificativo delle centraline, la provincia, la data di rilevazione, le coordinate, e il valore di PM2.5 per tutte le centraline di ogni provincia in cui il valore minimo di PM2.5 rilevato nella provincia abbia superato i $40 \mu\text{g}/\text{m}^3$ a gennaio 2022. Si ricorda che esiste una funzione MIN che calcola il minimo dei valori dell'attributo passato come argomento per tutte le tuple del gruppo di riferimento. **[5 punti]**

Soluzione

```
CREATE TABLE Centralina
(Id CHAR(5) NOT NULL,
NomeLuogo CHAR(20),
Latitudine CHAR(8),
Longitudine CHAR (8),
Provincia CHAR(6),
Indirizzo CHAR (20)
Telefono CHAR (13),
PRIMARY KEY (Id)
);
```

```
CREATE UNIQUE INDEX Centralina_key ON Centralina(Id);
```

```
CREATE TABLE Rilevazione
(IdCentralina CHAR(5) NOT NULL,
Data DATE NOT NULL,
Temperatura DECIMAL (3,1),
PM10 DECIMAL (3,1),
PM2.5 DECIMAL (3,1),
PRIMARY KEY (IdCentralina, Data)
);
```

```
CREATE UNIQUE INDEX Rilevazione_key ON Rilevazione(IdCentralina, Data);
```

```
SELECT *
FROM Centralina , Rilevazione
WHERE (Data > 30-11-2022) AND
(Data < 01-01-2023) AND
(PM10 > 55.0) AND
(Provincia = 'MI') AND
(Centralina.Id = Rilevazione.Id);
```

```
SELECT Id, Provincia, Latitudine, Longitudine, Data, PM2.5
FROM Rilevazione JOIN Centralina ON (Rilevazione.Id = Centralina. Id)
WHERE (Data > 31-12-2021) AND
(Data < 01-02-2022)
GROUP BY Provincia
HAVING MIN(PM2.5) > 40.0
);
%
```

Esercizio 4 - Reti di calcolatori

Si descrivano brevemente le caratteristiche di una rete a bus. La rete che ha indirizzi codificati con 4 numeri decimal (es.:192.68.23.1) è una rete con architettura a bus?[5 punti]

Soluzione

Vd libro di testo. La rete a bus consiste in un unico canale a cui è possibile collegare diversi apparati che condividono il canale per scambiarsi pacchetti. Occorre un protocollo di comunicazione che permetta un utilizzo condiviso del canale. Il collegamento può essere fatto in qualunque momento. I problemi principali sono legati al canale stesso: se ci sono malfunzionamenti sul canale, l'intera rete è bloccata; inoltre essendo il canale unico, il traffico può congestionarlo se ci sono molti apparati che lo utilizzano. La rete a cui si fa riferimento è Internet e non ha architettura a bus, ma è gerarchica e mista.

Esercizio 5 - Strumenti di produttività

Progettare un foglio elettronico che permetta di calcolare la formula $dy/dt = a1 * y(t)^2/a2 + a3 * y(t)$ con passo P=0.2 da 0 a 10. [5 punti]

Soluzione

Si dovevano indicare le caselle e i relativi contenuti. Nella prima riga vanno inserite le intestazioni delle colonne. In C0 si può inserire P e in D0 il valore del passo 0.2. Nella colonna A va inserito il tempo da 0 a 10 con passo 0.2 ad esempio scrivendo in A1 0, in A2 A1+D0 e trascinando il valore di questa casella fino ad ottenere il valore 10. Nella casella B1 va inserito il riferimento al valore iniziale di y che qui supponiamo sia memorizzato in F0: =F0. Nella casella B2 va inserito il valore aggiornato di y, cioè =D0*(H0*B1*B1)/L0+O0*B1)-B1. se in H0, L0 e O0 abbiamo rispettivamente a1, a2 e a3.

Parte II - Linguaggio C [Totale punti 30]

Esercizio 1 - Sviluppo

[6 punti] Si definiscano le opportune strutture dati per rappresentare un array di corsi universitari. Per ogni corso è necessario memorizzare il titolo del corso (costituito al più da 60 caratteri), il docente titolare del corso, il numero di CFU, il tipo di corso (emisemestrale, semestrale o annuale) e il numero di studenti iscritti. Per ogni docente è necessario memorizzare il cognome e la qualifica (ricercatore, professore associato, professore ordinario o docente a contratto).

1. [6 punti] Si scriva una funzione **conta** che conti il numero di corsi semestrali con al più 50 iscritti tenuti da un docente che non sia professore associato né professore ordinario. La funzione deve inoltre restituire (ad esempio mediante un parametro passato per indirizzo) la somma dei CFU di tali corsi.
2. [6 punti] Si scriva una funzione **sovrascrivi_docente** che ricevuto il titolo **t** di un corso e un docente **d** come parametri, registri **d** come docente del corso **t** (sovrascrivendo il docente eventualmente già presente). La funzione deve restituire 1 se la sovrascrittura è stata effettuata, 0 altrimenti.

Soluzioni

```
#define MAXCORSI 100
```

```
struct Docente {
    char cognome[30];
    char qualifica[30];
};
```

```
struct Corso {
    char titolo[60];
    struct Docente docente;
    int numero_cfu;
    char tipo_corso[15];
    int numero_studenti;
};
```

```
struct Corso corsi[MAXCORSI];
```

```
int conta(struct Corso corsi[], int num_corsi, int *somma_cfu) {
    int contatore = 0;
    *somma_cfu = 0;
    for (int i = 0; i < num_corsi; i++) {
        if (strcmp(corsi[i].tipo_corso, "semestrale") == 0 && corsi[i].numero_studenti <= 50 &&
            (strcmp(corsi[i].docente.qualifica, "professore associato") != 0 &&
             strcmp(corsi[i].docente.qualifica, "professore ordinario") != 0)) {
            contatore++;
            *somma_cfu += corsi[i].numero_cfu;
        }
    }
    return contatore;
}
```

```
int sovrascrivi_docente(struct Corso corsi[], int num_corsi, char titolo[], struct Docente docente) {
```

```

int sovrascritto = 0;
for (int i = 0; i < num_corsi; i++) {
    if (strcmp(corsi[i].titolo, titolo) == 0) {
        if (strlen(corsi[i].docente.cognome) != 0)
            \\ Nell'ipotesi che se il docente non c'/'e, sia stato inserito un record con cognome vuoto
            sovrascritto = 1;
        corsi[i].docente = docente;
        \\comunque assegnamo il docente al corso (l'ultimo if condiziona solo un'istruzione)
    }
}
return sovrascritto;
}

```

Esercizio 2 - Rilevazione di errori

[6 punti] Sia dato il seguente programma il cui scopo è, data una matrice $N \times M$ di interi inserita dall'utente, determinare l'indice della riga il cui elemento massimo è di valore minimo. Nel caso della seguente matrice:

$$\begin{pmatrix} -3 & 1 & -4 & -10 & -8 \\ 0 & 50 & 32 & -95 & -12 \\ -10 & -30 & 0 & -15 & -3 \end{pmatrix}$$

l'output desiderato è 2 in quanto gli elementi di valore massimo per ciascuna riga sono [1 50 0] e il minimo tra essi è 0 che corrisponde alla terza riga.

Si identifichino gli errori presenti nel programma, motivandone brevemente la scelta, e li si corregga. Sono presenti 6 linee errate o mancanti.

| | |
|---|---|
| 1. #include <stdio.h> | 20. i_maxmin = i; |
| 2. | 21. } |
| 3. #define N 3 | 22. } |
| 4. #define M 5 | 23. return i_maxmin; |
| 5. | 24. } |
| 6. int maxmin (int Mat[][], int n, int m) | 25. |
| 7. { | 26. int main() |
| 8. int i, j, i_maxmin, v_maxmin, max; | 27. { |
| 9. | 28. int i, j; |
| 10. for (i = 0; i <= n; i++) | 29. |
| 11. { | 30. printf("Inserisci gli elementi della matrice: "); |
| 12. max = Mat[i][0]; | 31. for (i = 0; i < N; i++) |
| 13. for (j = 1; j < m; j++) | 32. for (j = 0; j < M; j++) |
| 14. if (Mat[i][j] < min) | 33. scanf("%d", Mat[i][j]); |
| 15. max = Mat[i][j]; | 34. |
| 16. | 35. printf("Indice riga min max: %d\n", |
| 17. if (i == 0 && max < v_maxmin) | 36. maxmin(Mat, N, M)); |
| 18. { | 37. return 0; |
| 19. max = v_maxmin; | 38. } |

Soluzioni

- Linea 28: La matrice Mat non è dichiarata → `int i, j, Mat[N][M];`
- Linea 33: `scanf("%d", Mat[i][j]);` → `scanf("%d", &Mat[i][j]);`
- Linea 6: `int maxmin (int Mat[][], int n, int m)` → `int maxmin (int Mat[N][M], int n, int m)` oppure `int maxmin (int Mat[][M], int n, int m)`
- Linea 10: `for (i = 0; i <= n; i++)` → `for (i = 0; i < n; i++)`
- Linea 17: `if (i == 0 && max > v_maxmin)` → `if (i == 0 || max > v_maxmin)`
- Linea 19: `max = v_maxmin;` → `v_maxmin = max;`

Esercizio 3 - Comprensione del codice

[6 punti] Si dica cosa viene stampato dal seguente programma nei seguenti casi:

1. è inserita la matricola 258963;
2. è inserita la propria matricola.

Si descriva come la funzione `f1` modifica l'array `V` di dimensione `n`, supponendo che `V` contenga solo numeri interi tra 0 e 9 (estremi inclusi).

```
#include <stdio.h>

#define MAX 6

void f1 (int V[], int n)
{
    int X[10] = {0};
    int i, j;

    for (i = 0; i < n; i++)
        X[V[i]]++;

    i = 0;
    for (j = 9; j >= 0; j--)
    {
        while(X[j] > 0)
        {
            V[i] = j;
            i++;
            X[j]--;
        }
    }

    void f2 (int V[], int n, int x)
    {
        if (n == 1)
            *V = x;
        else
        {
            *V = *(V+1);
            f2(V+1, n-1, x);
        }
    }

    void stampa (int V[], int n)
    {
        int i;
        for (i = 0; i < n; i++)
            printf("%d", V[i]);
        printf("\n");
    }

    int main()
    {
        int A[MAX], i;

        printf("Inserisci la tua matricola  
una cifra per volta: ");
        for (i = 0 ; i < MAX; i++)
            scanf("%d", &A[i]);

        stampa(A, i);
        f1(A, i);
        stampa(A, i);
        f2(A, i, A[0]);
        stampa(A, i);

        return 0;
    }
}
```

Soluzioni

Il programma chiede all'utente di inserire una matricola di 6 cifre, una per volta, e quindi memorizza queste cifre in un array chiamato "A". La funzione "f1" ordina le cifre della matricola in ordine decrescente. La funzione utilizza un array "X" di lunghezza 10 per contare quante volte ogni cifra appare nella matricola. Quindi, utilizza un ciclo per scorrere l'array "X" dalla cifra 9 alla cifra 0 e copiare le cifre nell'array "A" in ordine decrescente. La funzione "f2" utilizza la tecnica della ricorsione per spostare tutte le cifre della matricola verso sinistra di una posizione, fatta eccezione per la prima cifra che viene sostituita con la cifra "x" passata come parametro. La funzione "stampa" stampa le cifre della matricola una per riga.

Quindi, quando si inserisce la matricola 258963, il programma stamperà:

258963
986552
695225

E quando si inserisce la matricola 230946, il programma stamperà:

230946
964320
039642

Nota: La funzione `f1` ordina solo in modo decrescente la matricola, non in modo crescente.