

Informatica - Allievi Ambientali - Prova del 19 luglio 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 frontespizio (**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	
<small>Testo dell'esercizio Y</small>	
Esercizio n° Z	
<small>Testo dell'esercizio Z</small>	
...	

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 - Aritmetica binaria [5 punti]

Si codifichino in complemento a 2 i numeri 8 e -13, se ne esegua la somma e la verifica di correttezza della stessa, eseguendo tutti i passaggi.

Soluzione

8|0
4|0
2|0
1|1

$$+8_{10} = 01000$$

13|1
6|0
3|1
1|1

$$+13_{10} = 01101_2$$

Ne facciamo il complemento a 2 invertendo i bit e sommando 1.

$$01101 \rightarrow 10010$$

10010 +
1

10011

$$10011_2 = -13_{10}$$

Facciamo la somma.

10011 +
01000

11011

Verifica, sottraendo 1 e invertendo i bit trovo il numero positivo corrispondente.

```
11011 -  
1  
-----  
11010 -> 00101
```

$$00110_2 = 1 \cdot 2^0 + 1 \cdot 2^2 = 5$$

Quindi il risultato è $11010_2 = -5_{10}$

Esercizio 2 - Macchina di Von Neumann [5 punti]

Si spieghino brevemente i ruoli dell'unità di controllo e del clock. Quante istruzioni macchina al secondo può al massimo eseguire una CPU con un clock a 10 MHz?

Soluzione

L'unità di controllo ha il ruolo di gestire il flusso di dati tra registri e unità aritmetico-logica. Un clock a 10MHz può abilitare l'esecuzione di 10 milioni di istruzioni macchina al secondo.

Esercizio 3 - SQL

Si progetti in SQL lo schema di una base di dati per la gestione di dati relativi a popolazioni di uccelli protetti presenti sul territorio. Per ogni uccello sono memorizzati il nome, l'area di rilevamento, la data del rilevamento, il numero di telefono del responsabile del rilevamento, e la quantità di esemplari rilevati. Si consideri che possono esserci diversi rilevatori nella stessa area e nello stesso giorno. Per ogni area sono memorizzate le coordinate geografiche dei due punti estremi del rettangolo che la include (in alto a sinistra e in basso a destra), il nome e la provincia di appartenenza. [5 punti]

Si progettino due query: la prima per ottenere la lista degli animali rilevati a giugno 2023 in provincia di Como. La seconda per ottenere le aree di rilevamento in cui sono stati rilevati gipeti della provincia in cui non siano stati rilevati più di 15 gipeti in totale. [5 punti]

Soluzione

```
CREATE TABLE Rilevamento  
(NomeUccello CHAR(20) NOT NULL,  
 NomeArea CHAR(20) NOT NULL,  
 NumeroEsemplari INTEGER,  
 Telefono CHAR (13) NOT NULL,  
 Data DATE NOT NULL,  
 PRIMARY KEY (NomeUccello, NomeLuogo, Data, Telefono)  
);  
  
CREATE UNIQUE INDEX Rilevamento_key ON Rilevamento(NomeAnimale, NomeLuogo, Data, Telefono);  
  
CREATE TABLE Area  
(NomeArea CHAR(20) NOT NULL PRIMARY KEY,  
 Latitudine1 CHAR(8),  
 Longitudine1 CHAR (8),  
 Latitudine2 CHAR(8),  
 Longitudine2 CHAR (8),  
 Provincia CHAR(2)  
);  
  
CREATE UNIQUE INDEX Area_key ON Rilevazione(NomeArea);  
  
SELECT NomeUccello  
FROM Rilevamento JOIN Area ON (Area.NomeArea = Rilevamento.NomeArea)  
WHERE (Data > 31-05-2022) AND  
      (Data < 01-07-2023) AND  
      (Provincia = 'CO');
```

```

SELECT NomeLuogo
FROM Rilevamento JOIN Area ON (Area.NomeArea = Rilevamento.NomeArea)
WHERE (NomeUccello = 'Gipeto')
GROUP BY Provincia
HAVING SUM(NumeroEsemplari) < 15
);

```

Esercizio 4 - Sistema Operativo

Si descriva brevemente il ruolo del file system in un sistema operativo e si indichino cinque delle principali funzioni che fornisce all'utente. **[5 punti]**

Soluzione

Vd libro di testo. Il file system gestisce la memoria di massa. Tra le principali funzioni abbiamo creazione, cancellazione, ridenominazione di file e cartelle, spostamento di un file da una cartella a un'altra.

Esercizio 5 - Strumenti di produttività

Progettare un foglio elettronico che permetta di calcolare la formula $dx/dt = k_1 * x(t)^2 + k_2 * x(t) + k_3$ con passo $P=0.1$ da 0 a 5. **[5 punti]**

Soluzione

Si dovevano indicare le caselle e i relativi contenuti. Una possibile organizzazione è la seguente. Nella prima riga, A0 e B0 vanno inserite le intestazioni delle colonne che conterranno gli istanti di tempo (t) e i valori della variabile x, rispettivamente. In D0 si può inserire $P =$ e in E0 il valore del passo 0.1. In D1 si può inserire $x(0) =$ ed in E1 il valore di x al tempo 0. In G0 possiamo scrivere $k_1 =$ e in H0 il valore di k_1 . le altre costanti potrebbero andare sulle righe successive. Nella colonna A va inserito il tempo da 0 a 5 con passo 0.5 ad esempio scrivendo in A1 0, in A2 A1+E0 e trascinando il valore di questa casella fino ad ottenere il valore 5. Nella casella B1 va inserito il riferimento al valore iniziale di x che qui supponiamo sia memorizzato in E1: =E1. Nella casella B2 va inserito il valore aggiornato di x, cioè =E0((H0*B1*B1)+H1*B1) + B1.*

Parte II - Linguaggio C [Totale punti 30]

Esercizio 1 - Sviluppo

[6 punti] Si consideri un'applicazione per la pianificazione delle vacanze estive in diverse destinazioni turistiche. Si definiscano le opportune strutture dati per rappresentare un array di destinazioni turistiche. Per ogni destinazione, è necessario memorizzare il nome della località, il periodo di validità (intervallo di date) dell'offerta, il prezzo dell'offerta e il numero di posti disponibili. **Si supponga che vi sia esattamente un'offerta per ogni località presente nell'array.**

1. [6 punti] Si scriva una funzione `destinazione_con_maggiori_posti` che analizzi l'array delle destinazioni turistiche e restituisca il nome della località con il maggior numero di posti disponibili. La funzione deve restituire una stringa vuota se non ci sono destinazioni presenti o se tutte le destinazioni hanno lo stesso numero di posti disponibili.
2. [6 punti] Si scriva una funzione `prezzo_totale` che riceva come parametri l'array delle destinazioni, un array di località `Locs` e un numero di ospiti `ospiti`. La funzione deve verificare se nell'array di destinazioni vi è un'offerta per ciascuna delle località in `Locs` con un numero di posti disponibili almeno pari a `ospiti`. In caso affermativo, la funzione deve restituire il prezzo totale delle offerte relative alle località in `Locs`. In caso negativo, la funzione deve restituire `-1`.

Soluzioni

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

```
typedef struct
{
    int giorno;
    int mese;
    int anno;
} data_t;
```

```
typedef struct
{
    char localita[100];
    data_t periodo_validita_inizio;
    data_t periodo_validita_fine;
    float prezzo;
    int posti_disponibili;
} destinazione_t;
```

```
void destinazione_con_maggiori_posti(destinazione_t destinazioni[], int num_destinazioni, char* destinazione_out)
{
    if (num_destinazioni == 0)
    {
        strcpy(destinazione_out, "");
        return;
    }
```

```
    int max_posti, min_posti, i;
    int index_max_posti = 0;
```

```
    for (i = 0; i < num_destinazioni; i++)
    {
        if (i == 0 || destinazioni[i].posti_disponibili > max_posti)
        {
            max_posti = destinazioni[i].posti_disponibili;
            index_max_posti = i;
        }
        if (i == 0 || destinazioni[i].posti_disponibili < min_posti)
        {
            min_posti = destinazioni[i].posti_disponibili;
        }
    }
```

```

}

if (max_posti == min_posti)
{
strcpy(destinazione_out, "");
return;
}

strcpy(destinazione_out, destinazioni[index_max_posti].localita);
}

float prezzo_totale(destinazione_t destinazioni[], int num_destinazioni, char* Locs[], int num_localita, int ospiti)
{
float prezzo_tot = 0.0;
int i, j, trovata;

for (i = 0; i < num_localita; i++) {
trovata = 0;

for (j = 0; j < num_destinazioni && trovata == 0; j++)
{
if (strcmp(destinazioni[j].localita, Locs[i]) == 0 &&
destinazioni[j].posti_disponibili >= ospiti) {
prezzo_tot += destinazioni[j].prezzo;
trovata = 1;
}
}

if (trovata == 0)
return -1;
}
return prezzo_tot;
}

```

Parte II - Linguaggio C [Totale punti 30]

Esercizio 1 - Rilevazione di errori

[6 punti] Una **matrice circolante** è una matrice quadrata in cui tutte le righe sono composte dagli stessi elementi e ogni riga è ruotata di un elemento verso destra rispetto al vettore riga precedente. Ad esempio, la seguente è una matrice circolante:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

Sia dato il seguente programma il cui scopo è verificare se una matrice inserita dall'utente è circolante. 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>	13. }
2.	14. return 1 - stop;
3. #define MAX_SIZE 5	15. }
4.	16.
5. int controlla_riga(int A[MAX_SIZE], int B[], int i)	17. int verifica_matrice (int mat[][MAX_SIZE], int i)
6. {	18. {
7. int j, k, stop = 0;	19. if (i == MAX_SIZE)
8. for (j = 0; j < MAX_SIZE && stop == 0; j++)	20. return 1;
9. {	21. else
10. k = j + i	22. {
11. if (A[j] != B[k])	23. res = controlla_riga(mat[0], mat[i]);
12. stop = 1;	24. if (res == 0)

```

25.         return res;
26.
27.     return verifica_matrice(mat, i);
28. }
29. }
30.
31. int main()
32. {
33.     int mat[MAX_SIZE][MAX_SIZE], i, j, res;
34.
35.     printf("Inserisci gli elementi della matrice:\n");
36.     for (i = 0; i < MAX_SIZE; i++)
37.         for (j = 0; j < MAX_SIZE; j++)
38.             scanf("%f", &mat[i][j]);
39.
40.     res = verifica_matrice(mat, 1);
41.     if (res == 1)
42.         printf("La matrice e' circolante.\n");
43.     else
44.         printf("La matrice non e' circolante.\n");
45.
46.     return 0;
47. }

```

Soluzioni

- Linea 10: $k = j + i \rightarrow k = (j + i) \% \text{MAX_SIZE}$; E' stata accettata come corretta anche la sola aggiunta del ;
- Linea 18: Aggiungere int res;
- Linea 23: `res = controlla_riga(mat[0], mat[i]);` → `res = controlla_riga(mat[0], mat[i], i);`
- Linea 27: `return verifica_matrice(mat, i);` → `return verifica_matrice(mat, i + 1);`
- Linea 38: `scanf("%f", &mat[i][j]);` → `scanf("%d", &mat[i][j]);`
- Linea 51: `if (res == 1)` → `if (res == 1)`

Esercizio 3 - Comprensione del codice

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

- è inserita la matricola 123456;
- è inserita la propria matricola.

Si descriva cosa viene salvato nella stringa b quando la funzione f2 è invocata con parametro una stringa generica a. Si fornisca inoltre un esempio di stringa a di lunghezza 10 tale che, con l'invocazione della funzione f2, in b viene salvata la stringa "xyz".

```

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

void f2 (char* b, char *a)
{
    char c = '*';
    int i, k;

    i = 0;
    k = 0;
    while (a[i] != '\0')
    {
        if (a[i] != c)
        {
            b[k] = a[i];
            c = a[i];
            k++;
        }
        i++;
    }
    b[k] = '\0';
}

void f1 (char* b, char *a)
{
    if (*a == '\0')
        *b = *a;
    else
    {
        *b = *a;
        b++;
        *b = *a;
        b++;
        a++;
        f1(b, a);
    }
}

void f0 (char *b, char *a)
{
    strcpy(b, a);
    strcat(b, a);
}

int main()
{

```

```

char matricola[20], s1[40], s2[40], s3[40];

printf("Inserisci la matricola: ");
scanf("%s", matricola);

printf("%s - %d\n", matricola, strlen(matricola));

f0(s1, matricola);
printf("%s - %d\n", s1, strlen(s1));

f1(s2, matricola);
printf("%s - %d\n", s2, strlen(s2));

f2(s3, matricola);
printf("%s - %d\n", s3, strlen(s3));

return 0;
}

```

Soluzioni

1. Inserendo la matricola 123456 viene stampato:

```

123456 - 6
123456123456 - 12
112233445566 - 12
123456 - 6

```

2. Inserendo la matricola 788955 viene stampato:

```

788955 - 6
788955788955 - 12
778888995555 - 12
7895 - 4

```

La funzione `f2` salva in `b` i caratteri della stringa `a` rimuovendo le ripetizioni consecutive dello stesso carattere. Una stringa `a` di lunghezza 10 che fa sì che in `b` sia salvato `"xyz"` è ad esempio `"xxxyyyzzzz"`.