

GESTIONE DELL'INFORMAZIONE AZIENDALE
APPELLO STRAORDINARIO DI PASQUA 12/04/2022

NOME _____

COGNOME _____

SEZIONE 1 - DOMANDE

Domanda 1 (3 punti)

Si considerino la DFunction e la query mostrate di seguito che operano sulla tabella ANAGRAFICA_CLIENTI.

DFUNCTION:

```
X = Nz(DLookup("Cognome", "ANAGRAFICA_CLIENTI", "Regione = 'Lombardia'), "Inesistente")
```

QUERY:

```
SELECT Cognome  
FROM ANAGRAFICA  
WHERE Regione = 'Lombardia'
```

Ci sono differenze tra i due approcci? Se sì, quali?

Domanda 2 (3 punti)

Si considerino le seguenti istruzioni in VBA

```
Public A As Integer
```

```
Public Function foo(ByRef X As Integer) As Integer
```

```
    foo = X
```

```
    X = X + Int(rnd()*100)
```

```
End Function
```

```
Public Sub Use_Foo()
```

```
    A = 2
```

```
    B = foo(A)
```

```
    debug.print(A)
```

```
    debug.print(B)
```

```
End Sub
```

Supponiamo che la subroutine Use_Foo() sia stata chiamata e che, al suo interno, la funzione rnd() abbia restituito il numero 0.54231. Sulla base di tali informazioni, è possibile dire che valori figurerebbero a video? Si spieghi.

Domanda 3 (3 punti)

Si spieghi, in massimo 6 righe di testo, cosa s'intende per evento intermedio di un diagramma BPMN. In che senso un tale evento, può essere assimilato ad una valvola di un circuito idraulico?

Domanda 4 (5 punti)

Si consideri un fatto di vendita a 3 dimensioni: Nazione, Anno e Prodotto. Come si dovrebbe procedere per ottenere le "coordinate" di ciascuna dimensione? Supponendo che si utilizzi un database MOLAP, si disegnino le matrici multidimensionali necessarie a caratterizzare ciascun fatto. Quanti sono in totale i fatti rappresentati?

SEZIONE 2 – ESERCIZI

Esercizio 1 (6 punti)

Si consideri la tabella INVENTARIO di un database relazionale, che riporta tutte le transazioni di prelievo e di deposito effettuate nei magazzini di cui dispone un'azienda. A tal fine la tabella INVENTARIO sarà contraddistinta dai seguenti campi: ID_Magazzino, ID_Prodotto, Data, Quantità. La quantità sarà positiva in caso di deposito, negativa in caso contrario.

Si scriva una query parametrica che restituisce le quantità di ciascun prodotto stoccato, che erano disponibili nel magazzino e nella data indicati in input (parametri della query).

Come andrebbe modificata la query per far sì che vengano visualizzati solo i prodotti la cui disponibilità valorizzata in termini monetari sia maggiore di 10000 euro?

Si assumano a piacere eventuali dati mancanti.

Esercizio 2 (6 punti)

Si scriva una funzione che riceve in input una stringa e restituisce True se la stringa è palindroma, False altrimenti. Una stringa è palindroma se può essere indifferentemente letta da sinistra verso destra o da destra verso sinistra. Si noti che per effettuare la verifica è sufficiente confrontare le prime $n/2$ lettere di sinistra con le rispettive $n/2$ lettere di destre (la prima con l'ultima, la seconda con la penultima, ecc.). Se n è dispari, la lettera centrale non va confrontata con nessun'altra lettera.

Suggerimento: si faccia uso di un ciclo e della funzione `Mid(string, start, length)`

Esercizio 3 (8 punti)

Si consideri un'ipotetica tabella VENDITE composta dei seguenti 5 campi: *ID*, *ID_Nazione*, *ID_Prodotto*, *Data* e *Quantità*. Ogni record, in pratica, rappresenta le quantità di un certo prodotto che sono state vendute in una certa data ed in una certa nazione.

Si crei una classe che permette di eseguire e di aggiornare la somma cumulata delle vendite per uno specifico prodotto in una specifica nazione. A tal fine la classe sarà caratterizzata da:

- Quattro attributi (variabili) pubblici: *ID_Prodotto*, *ID_Nazione*, *Anno_Finale* e *Q_Tot*.
- Due metodi: *Somma_cumulata* e *Update*.

Il metodo *Somma_cumulata* riceve in input 3 argomenti (*ID_Naz*, *ID_Pr*, *Anno_End*), filtra i dati della tabella in base ai valori dei tre argomenti, ed esegue la somma cumulata richiesta. I valori dei tre argomenti e la somma cumulata ottenuta vengono infine registrati nei corrispondenti attributi della classe, di modo da poter essere utilizzati nei futuri aggiornamenti.

Il metodo *Update* riceve un solo input di tipo opzionale (*Anno_End*) e aggiorna la somma cumulata salvata nell'attributo di classe *Q_Tot*, considerando solo gli anni che intercorrono tra *Anno_Finale* ed *Anno_End*. Se *Anno_End* non è passato in input, l'aggiornamento verrà fatta considerando tutti gli anni successivi ad *Anno_Finale*, che sono presenti nella tabella *VENDITE*. Ovviamente, l'aggiornamento deve rispettare la nazione e il prodotto relativamente a cui è stata eseguita la precedente somma cumulata (salvata in *Q_Tot*).

Si risolva il problema utilizzando i recordset e i loro metodi.

In alternativa alla classe è ammissibile risolvere il problema utilizzando una funzione che esegue la somma cumulata. Questo approccio ha però una valutazione inferiore, pari a 6 punti.