Esame del 05 febbraio 2016

Sequenze di tasti per i caratteri in tabella

Car.	Sequenza	Car.	Sequenza
[	AltGR [	]	AltGR]
{	AltGR Shift [	}	AltGR Shift [
~	Alt 126		
>	Alt 062	<	Alt 060

Equivalenze simboli relazionali/logici

Simbolo	VBA	MatLab
$\neq$	<>	~=
`negazione`	`NOT`	~
`congiunzione`	`AND`	&& [
`disgiunzione`	`OR`	||

1. VBA

Il file di testo `Immatricolati2016.txt` contiene i dati relativi al titolo di studio degli immatricolati dei corsi di laurea triennali di Economia dell'anno 2015/2016. Ogni riga contiene i dati di un immatricolato ed è composta dai seguenti dati

(a) matricola dello studente,

(b) voto di maturità,

(c) codice della classe di maturità,

(d) descrizione della classe di maturità,

(e) numero di ore di stage svolte dallo studente.

I dati nella riga sono separati dal carattere '`;`'. Inoltre sia il voto di maturità sia il numero di ore di stage dello studente potrebbero mancare. In particolare le ore di stage dello studente sono previste solo per le scuole professionali (dove possono anche essere pari a 0).

Definire in VBA dei sottoprogrammi per le operazioni:

- `Start` senza parametri. L'operazione deve copiare i dati del file `Immatricolati2016.txt` nel primo foglio di lavoro in modo che la colonna A contenga la matricola dello studente, la colonna B il voto di maturità, la colonna C il codice della classe di maturità e la colonna D la descrizione della classe. Il numero di ore di stage non deve essere copiato. Inoltre le righe del file che non contengono il voto di maturità non devono essere copiate.
- `Trova` con parametri:
  - `val` di tipo `String` e
  - `col` di tipo `Integer`.

  L'operazione `Trova` deve dare come risultato `true` se il valore del parametro `val` è contenuto nella colonna di indice `col` del primo foglio di lavoro. L'operazione deve dare come risultato `false` altrimenti.
- `Frequenza` con due parametri:
  - `val` di tipo `String` e
  - `col` di tipo `Integer`.

L'operazione `Frequenza` deve dare come risultato quante volte il valore del parametro `val` compare nella colonna di indice `col` del primo foglio di lavoro (frequenza assoluta).

- `Copia` con cinque parametri:
  - `val` di tipo `String`,
  - `freqA` di tipo `Integer`,
  - `freqR` di tipo `Double`,
  - `r` di tipo `Integer`, e
  - `c` di tipo `Integer`.

  L'operazione `Copia` deve copiare nel secondo foglio di lavoro:
  - il valore del parametro `val` nella cella di riga `r` e colonna `c`,
  - il valore del parametro `freqA` nella cella di riga `r` e colonna `c+1`,
  - il valore del parametro `freqR` nella cella di riga `r` e colonna `c+2`,
- `Elabora` senza parametri. L'operazione deve esaminare i dati presenti nel primo foglio di lavoro e per ogni classe di maturità deve scrivere consecutivamente nel secondo foglio di lavoro (a partire dalla riga 1) nella colonna A il codice della classe, nella colonna B la sua frequenza assoluta e nella colonna C la sua frequenza relativa.
- `ScriviInFile` senza parametri. L'operazione deve copiare i valori della colonna C del secondo foglio di lavoro nel file di testo `Risultati2016.txt`

Nota Bene:

- Per ogni operazione decidere (ove possibile) se realizzare una `Sub` oppure una `Function`.
- I sottoprogrammi devono avere il medesimo nome delle corrispondenti operazioni che realizzano.
- Se si ritiene necessario aggiungere altri sottoprogrammi.
- I dati nel foglio di lavoro devono essere inseriti a partire dalla riga 1 e senza lasciare righe vuote interposte alle righe occupate dai dati.
- Utilizzare, se è possibile, i sottoprogrammi `Frequenza`, `Trova` e `Copia` per realizzare il sottoprogramma `Elabora`.

Funzioni utili per la soluzione:

Prototipo	Descrizione	Esempio

`Integer Len(String s)`	Restituisce un valore `Integer` corrispondente al numero di caratteri della stringa `s`	`Dim TestS As String`
		`Dim TestL As Integer`
		`TestS = "Hello World"`
		`TestL = Len(TestS)`
		`' TestL contiene 11.`

`String() Split (String s, String d)`	Restituisce un vettore con indice primo elemento zero che contiene il numero di sottostringhe di `s` delimitate da `d`.	`Dim TestS As String`
		`Dim TestA() As String`
		`TestString =`
		`"Questo;esempio;di;test"`
		`'TestArray[0] contiene "Questo"`
		`'TestArray[1] contiene "esempio"`
		`'TestArray[2] contiene "di"`
		`'TestArray[3] contiene "test"`

2. MatLab

L'università Milano-Bicocca ha condotto un'indagine sulla scuola superiore di provenienza degli iscritti ai corsi di laurea di Economia. I risultati di questa indagine sono memorizzati nel file di testo di nome `Risultati2016.txt`.

Si richiede di realizzare in Matlab la funzione `Simula()` che simula il processo di immatricolazione degli studenti all'università.

La funzione `Simula()` ha in ingresso:

- un vettore riga `F` di valori che rappresentano i valori della funzione di probabilità della variabile casuale "Classe di Maturità";
- il valore `N` che indica la durata della simulazione in giorni;
- il valore `T` che indica il numero massimo di studenti che si possono iscrivere giornalmente.

La funzione produce in uscita:

- `Iscritti`, vettore con il numero di studenti iscritti giorno per giorno;
- `Provenienza`, vettore contenente il numero totale di iscritti attribuiti alle classi di maturità di provenienza secondo la distribuzione del vettore `F` al termine della simulazione.

La simulazione ha durata `N` giorni. Per ogni giorno, utilizzando i dati del vettore `F`, la funzione deve calcolare il numero di iscritti attribuiti alle classi di maturità di provenienza e accumularli nel vettore `Provenienza`.

Gli iscritti giornalieri sono generati a partire da una distribuzione di probabilità uniforme fra 1 e `T`.

La funzione controlla:

- il numero dei parametri in ingresso; se vi sono meno di 3 parametri termina con un messaggio di errore; se `T` o `N` non sono numeri positivi termina con un messaggio di errore;
- il numero di parametri in uscita; se i parametri non sono 2 la funzione termina con un messaggio di errore.

Realizzare infine lo script Elabora che

- carica in una variabile di nome `Frequenze` i dati del file `Risultati2016.txt`,
- esegue la funzione `Simula()` con parametri `Frequenze`, 100 e 200 e scrive il risultato nella variabili `I` e `P`.

Funzioni utili per la soluzione: `csvread()`, `error()`, `zeros()`, `randi()`, `cumsum()`, `rand()`, `sum()`.

3. Java (facoltativo)

Costruire un insieme di classi Java per la gestione degli studenti immatricolati all'Università Bicocca, tenendo conto che:

- tutti gli studenti hanno matricola, tipologia di diploma e voto di diploma;
- gli studenti dei corsi professionali hanno un numero di ore di tirocinio;
- gli studenti dei corsi non professionali non hanno ore di tirocinio ma possono avere un numero di ore di ampliamento dell'offerta formativa.

Si deve quindi definire una gerarchia di classi per risolvere il problema proposto che deve definire tre classi `Studente`, `StudenteProfessionale` e `StudenteNonProfessionale`. Nella gerarchia devono essere opportunamente definiti i seguenti attributi e metodi:

- Attributi: `matricola`, `voto` , `tipologia`, `oreTirocinio`, `oreAmpliamento`.
- Costruttori: definire adeguatamente per dare un valore iniziale agli attributi.
- Metodi:
  - `double calcola ()`: il metodo deve permettere di determinare un valore che per i corsi professionali è ottenuto moltiplicando per 0.75 il valore del voto di diploma a cui viene sommato il numero di ore di tirocinio diviso 20. Per gli studenti degli altri corsi il valore si ottiene moltiplicando per 0.75 il valore del voto di diploma a cui deve essere sommato un valore casuale intero uniforme compreso fra 10 e 20 (utilizzare il metodo `Math.random()`).
  - tutti i metodi per reperire il valore degli attributi della classe.

Si deve inoltre realizzare la classe `Immatricolati` con

- un unico attributo di nome `listaStudenti` e tipo `java.util.LinkedList<Studente>`,
- un costruttore che inizializza l'attributo `listaStudenti` ad una lista vuota, e

- il metodo `void addStudente(Studente s)` che permettere di aggiungere un studente all'attributo `listaStudenti`.

Infine, definire il punto di inizio di esecuzione che

- costruisce un oggetto di tipo `Immatricolati` e assegna il suo riferimento ad una variabile di nome `immatricolatiBicocca`;
- legge i dati del file di testo `Immatricolati2016.txt` e per ogni riga del file in cui è presente il voto di diploma aggiunge un oggetto di tipo `StudenteProfessionale` o `StudenteNonProfessionale` all'oggetto riferito da `immatricolatiBicocca` (basandosi sul fatto che la riga contiene o no il numero di ore di stage svolte dallo studente).

Gli attributi di tutte le classi devono essere definiti **privati**.