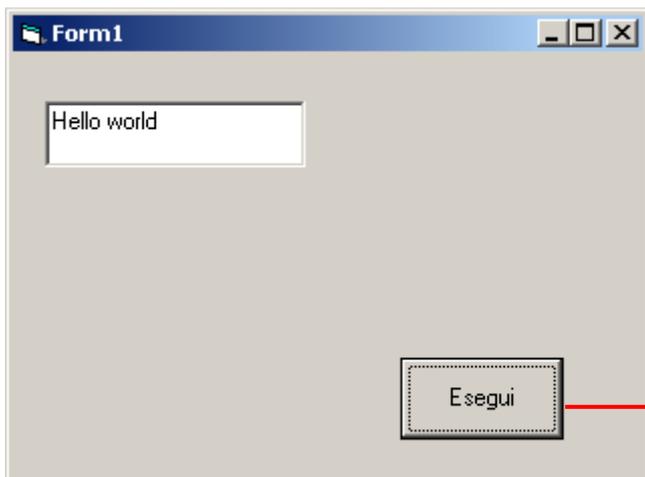
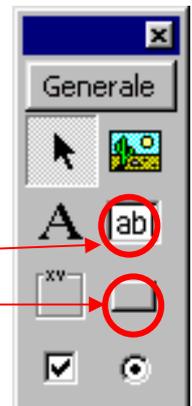


Esercitazione n. 1

1. Creare un **nuovo form** (appare un form vuoto denominato *form1*)
2. Inserire un controllo **TextBox**
3. Inserire un controllo **CommandButton**
4. cliccare 2 volte sul pulsante di comando e scrivere il codice sotto riportato
5. cliccare sul pulsante  per avviare l'esecuzione e verificare il corretto funzionamento
6. Salvare il form con il nome *form1* e il progetto con il nome *esercizio1*

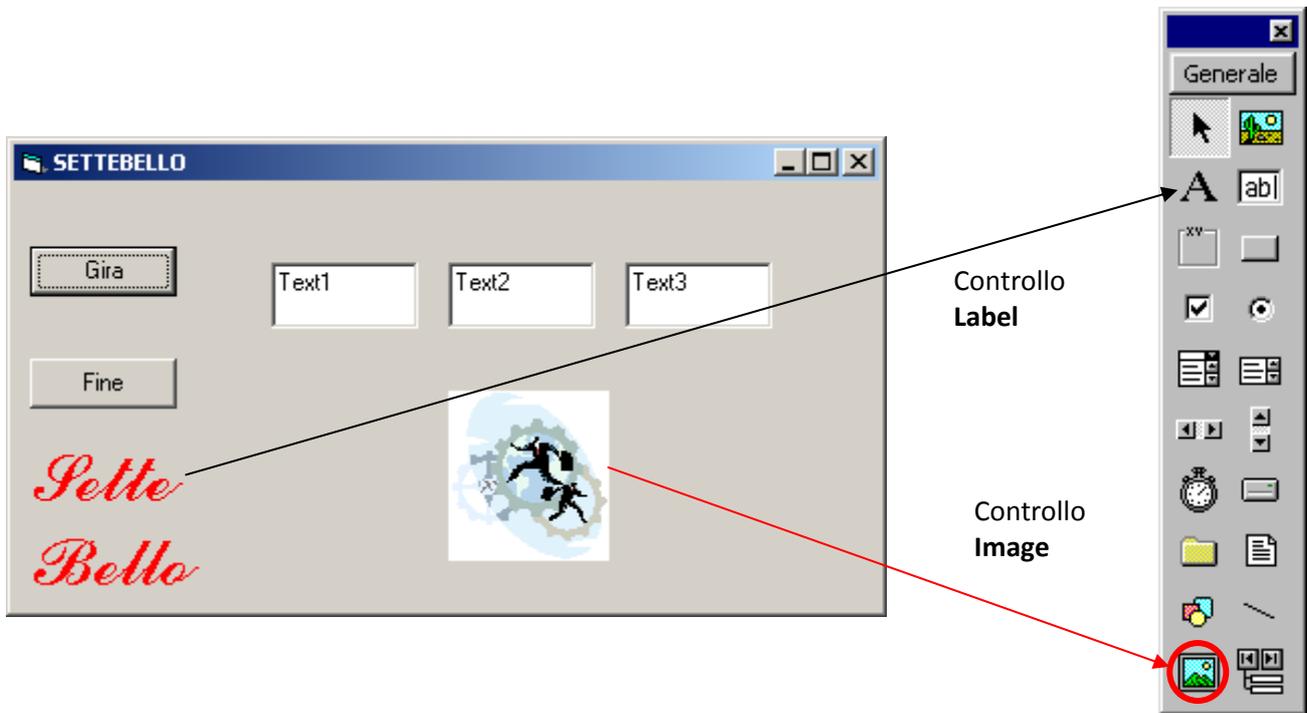


```
Private Sub Esegui_Click()
```

```
'Premendo il pulsante si ottiene alternativamente  
'la scrittura di "Hello world" e di "Hello friends"
```

```
  If Text1 = "Hello world" Then  
    Text1 = "Hello friends"  
  Else  
    Text1 = "Hello world"  
  End If
```

```
End Sub
```



Private Sub Gira_Click()

```
'Se esce un sette visualizza l'immagine
Image1.Visible = False 'Nasconde l'immagine
'Genero dei numeri casuali da 0 a 9
Randomize
Text1 = Int(Rnd * 10)
Text2 = Int(Rnd * 10)
Text3 = Int(Rnd * 10)
If (Text1 = 7) Or (Text2 = 7) Or (Text3 = 7) Then
    Image1.Visible = True 'Visualizza l'immagine
    Beep 'Emette un segnale sonoro
End If
```

End Sub

La funzione **Int()** restituisce la parte intera dell'argomento

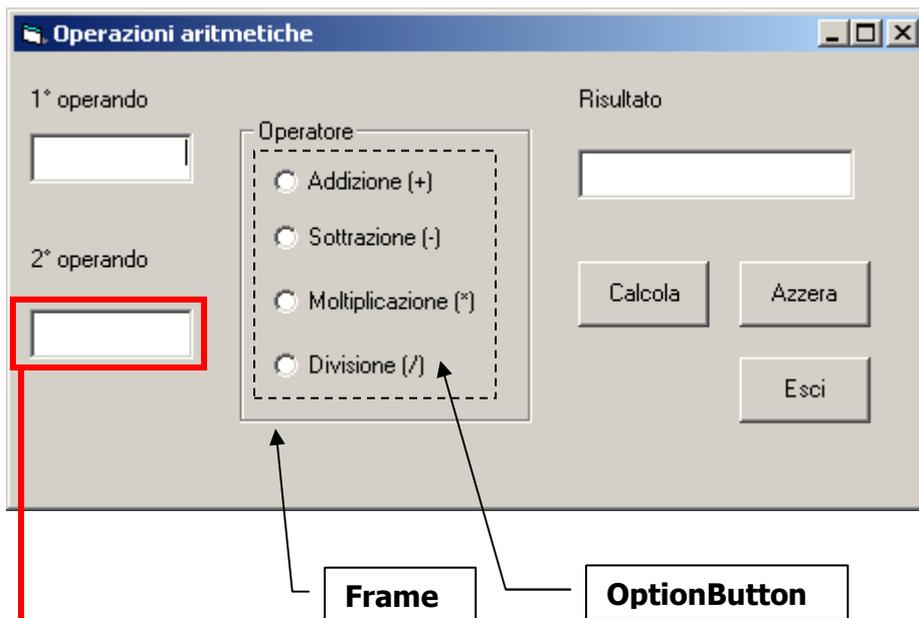
La funzione **Rnd** restituisce un numero casuale decimale compreso tra 0 e 1 ($0 < n < 1$)

Private Sub Fine_Click()

```
End 'termina l'esecuzione del programma
```

End Sub

Esercitazione n. 3



Private Sub txtOperando2_Change()

```
If Not IsNumeric(txtOperando2) Then  
    txtOperando2 = ""  
End IfEnd Sub
```

Private Sub Fine_Click()

```
End 'termina l'esecuzione del programma
```

End Sub

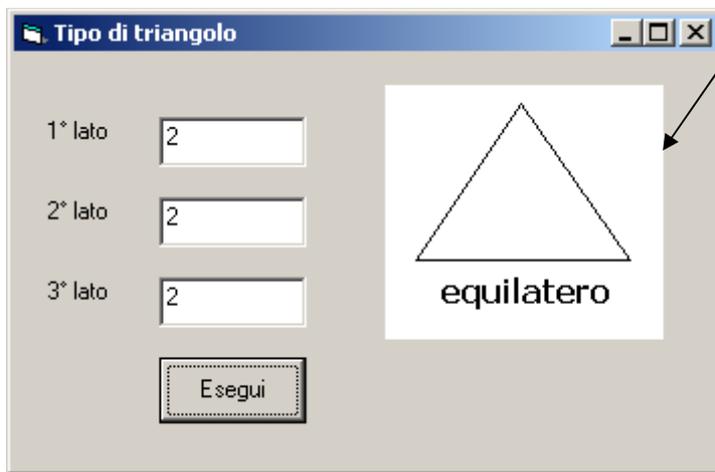


Il controllo **OptionButton** (*pulsante di opzione*) consente di visualizzare un'opzione che può essere attivata o disattivata. In genere, i controlli OptionButton vengono utilizzati in un gruppo di opzioni tra cui è *possibile selezionarne una*.



Il controllo **Frame** (*cornice*) consente di raggruppare controlli in modo visibile e identificabile. Può essere utilizzato per suddividere un form in modo funzionale, ad esempio separando gruppi di controlli OptionButton.

Esercitazione n. 4



Image

Oggetto App

È un oggetto globale a cui si accede mediante la parola chiave App. Consente di specificare il titolo dell'applicazione, il percorso e il nome del file eseguibile e dei file della Guida, nonché di stabilire se è già in esecuzione un'istanza dell'applicazione.

App.path

Restituisce il percorso corrente

Private Sub **esegui_Click()**

```
Dim a, c, b As Single
a = Val(Text1)
b = Val(Text2)
c = Val(Text3)
If a < b + c And b < a + c And c < a + b Then
  If a = b And b = c Then
    Image1.Picture = LoadPicture(App.Path + "\equilatero.gif")
    Image1.Visible = True
  Else
    If a = b Or b = c Or a = c Then
      Image1.Picture = LoadPicture(App.Path + "\isoscele.gif")
      Image1.Visible = True
    Else
      Image1.Picture = LoadPicture(App.Path + "\scaleno.gif")
      Image1.Visible = True
    End If
  End If
Else
  Image1.Visible = False
  MsgBox "Non sono misure di un triangolo"
End If
```

Carica un'immagine grafica in un oggetto Form, un controllo PictureBox o un controllo **Image**.

L'argomento da passare è il nome del file grafico che si desidera caricare, compreso di percorso

End Sub

```
Private Sub Form_Load()
  Image1.Visible = False
End Sub
```

```
Private Sub Text1_GotFocus()
  Image1.Visible = False
End Sub
```

Cercare nell'help l'evento **GotoFocus** e darne di seguito una breve illustrazione:

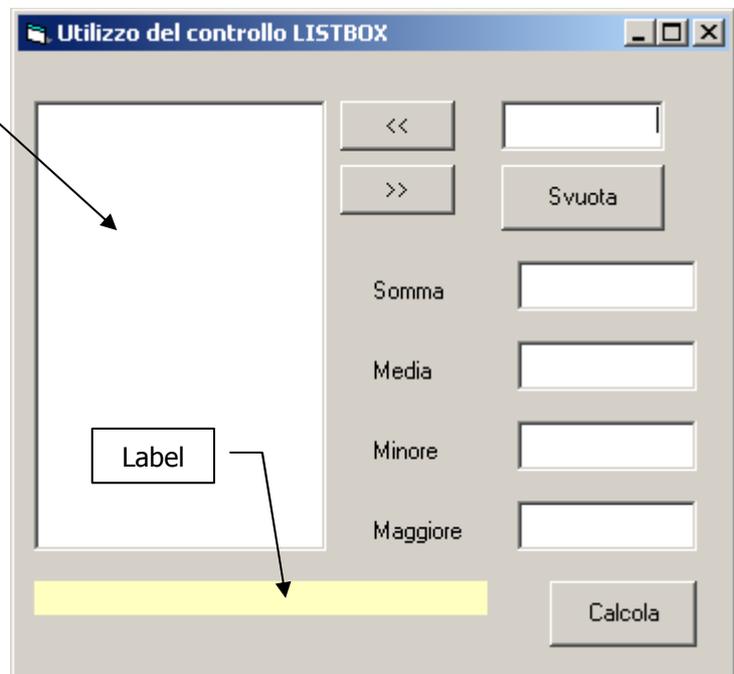
Esercitazione n. 6



Listbox

Il controllo **Listbox** (casella di riepilogo) consente di visualizzare un elenco di elementi tra cui è possibile selezionarne uno o più di uno. Se il numero di elementi dell'elenco è maggiore del numero di elementi visualizzabili, viene automaticamente aggiunta al controllo Listbox una barra di scorrimento.

Se non viene selezionato nessun elemento, il valore della proprietà **ListIndex** è **-1**. Il primo elemento dell'elenco corrisponde a *ListIndex* 0. Il valore della proprietà *ListCount* è sempre superiore di una unità al valore di *ListIndex* maggiore.



Per *aggiungere* o *eliminare* elementi in un controllo Listbox, utilizzare i metodi **AddItem** o **RemoveItem**. Per consentire l'accesso agli elementi di Listbox, impostare le proprietà *List*, **ListCount** e **ListIndex**.

Private Sub **Inserisci_Click()**

```
If IsNumeric(Text1) Then
    List1.AddItem (Text1)
    elementi.Caption = "Nella lista sono contenuti " + Str(List1.ListCount) + " elementi"
Else
    MsgBox "Non hai scritto un numero!"
End If
Text1 = ""
```

End Sub

Se **listIndex** \neq **-1** allora è stato selezionato un elemento

Private Sub **Estrai_Click()**

```
If List1.ListIndex <> -1 Then
    List1.RemoveItem (List1.ListIndex)
Else
    MsgBox "Devi prima selezionare un elemento", vbInformation, "ERRORE"
End If
```

```
elementi.Caption = "Nella lista sono contenuti " + Str(List1.ListCount) + " elementi"
```

End Sub

Private Sub **Svuota_Click()**

```
List1.Clear
elementi.Caption = "Nella lista sono contenuti " + Str(List1.ListCount) + " elementi"
```

End Sub

Private Sub **Calcola_Click()**

```
Dim Somma, min, max As Single
Dim I as Integer
List1.ListIndex = 1
max = List1
```

```

min = List1
For i = 0 To List1.ListCount - 1
  List1.ListIndex = i
  Somma = Somma + Val(List1.Text)

  'Calcolo del minimo
  If min > Val(List1.Text) Then
    min = Val(List1.Text)
  End If

  'Calcolo del massimo
  If max < Val(List1.Text) Then
    max = Val(List1.Text)
  End If
Next i

'Assegnazione dei risultati
txtSomma = Somma
txtMin = min
txtMax = max

If list1.listcount <> 0 Then
  txtMedia = Somma / List1.ListCount
Else
  txtMedia = "Impossibile"
End If

```

End Sub

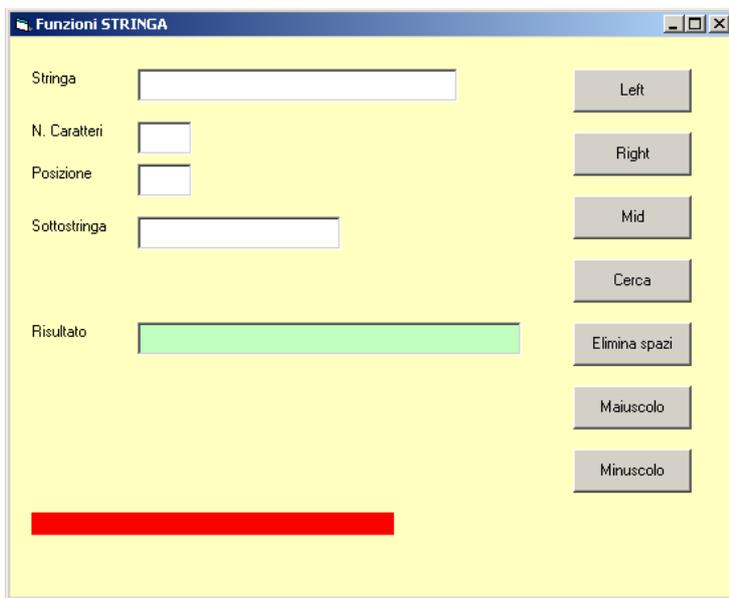
ESERCIZI SULLE STRINGHE

Generazione di un codice personale

Conoscendo cognome e nome, sesso, classe e sezione, Generare un codice secondo le istruzioni riportate di seguito:

Iniziale del cognome e prime due vocali		Primi tre Caratteri in posizione dispari del nome			Sesso: 1=M 0=F	Anno di nascita (formato aa)		Mese di nascita (formato mm)		

Controllare la validità della data di nascita utilizzando la funzione isdate()



```
Private Sub destra_Click()
    txtRisultato = Right(stringa, txtncar)
End Sub
```

```
Private Sub Instr_Click()
    pos = InStr(stringa, sottostringa)
    If pos <> 0 Then
        txtRisultato = "Stringa trovata nella posizione " & pos
    Else
        txtRisultato = "Stringa non trovata"
    End If
End Sub
```

```
Private Sub maiusc_Click()
    txtRisultato = UCase(stringa)
End Sub
```

```
Private Sub mezzo_Click()
    txtRisultato = Mid(stringa, txtpos, txtncar)
End Sub
```

```
Private Sub Minusc_Click()
    txtRisultato = LCase(stringa)
End Sub
```

```
Private Sub sinistra_Click()
    txtRisultato = Left(stringa, txtncar)
End Sub
```

```
Private Sub Spazi_Click()
    txtRisultato = Trim(stringa)
End Sub
```

```
Private Sub stringa_Change()
    caratteri = "La stringa è lunga " & Len(stringa)
End Sub
```

Cifrario di Cesare

Il cifrario di Cesare è il più antico algoritmo crittografico di cui si abbia traccia storica. È un cifrario a sostituzione monoalfabetica in cui ogni lettera del testo in chiaro è sostituita nel testo cifrato dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto. In particolare, Cesare utilizzava uno spostamento di 3 posizioni (la chiave era dunque "3"), secondo il seguente schema:

Testo in chiaro	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	v	z
Testo cifrato	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z	A	B	C

Per cifrare un messaggio, basta prendere ogni lettera del testo in chiaro e sostituirla con la corrispondente lettera della riga "testo cifrato". Per decifrare, viceversa. Ecco un semplice esempio:

Testo in chiaro: attaccare gli irriducibili galli alla ora sesta

Testo crittato: DZZDFFDUH LON NUUNGAFNENON LDOON DOOD RUD VHVZD

Scrivere un programma che fornito un messaggio e la chiave restituisca il messaggio cifrato.

Storia

Il cifrario di Cesare prende il nome da Giulio Cesare, che lo utilizzò con una chiave 3 per proteggere un messaggio d'importanza militare per Cicerone. Al tempo era sicuro, perché i nemici spesso non erano in grado di leggere nemmeno un testo in chiaro, figuriamoci uno cifrato. Dalla scoperta dell'analisi delle frequenze da parte degli Arabi attorno all'anno 1000, tutti i cifrari di questo tipo sono diventati rompibili in modo facile, spesso banale. Nessuno è adatto per comunicazioni sicure, ora, e neanche negli ultimi 1000 anni

Cifrario Atbash

L'atbash è un semplice cifrario a sostituzione monoalfabetica in cui la prima lettera dell'alfabeto con l'ultima, della seconda con la penultima, e così via, "invertendo" l'ordine alfabetico delle lettere.

Nel moderno alfabeto italiano, questo significa:

Testo in chiaro	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	v	z
Testo cifrato	Z	V	U	T	S	R	Q	P	O	N	M	L	I	H	G	F	E	D	C	B	A

il procedimento da utilizzare per decifrare è identico a quello per cifrare.

Scrivere un programma che fornito un messaggio restituisca il messaggio codificato tramite il cifrario Atbash

Scrivere i programmi in Visual Basic per risolvere i seguenti problemi:

1. Trasformare una data nella forma gg/mm/aa nella forma in lettere. Esempio: 10/10/88 --> 10 ottobre 1988. Usare la funzione isdate() per verificare che la stringa immessa sia una data corretta.
2. Data una stringa determinare da quante parole è formata.
3. Ottenere una stringa in ordine inverso a quello in cui è stata scritta.
4. Una "palindrome" è una parola che può essere letta nello stesso modo sia da destra verso sinistra che da sinistra verso destra, ad esempio: otto, Ada. Scrivere una programma che data una parola verifica se è una palindrome oppure no.
5. Eliminare, da una frase, gli spazi ripetuti lasciandone uno solo.
6. Data una stringa contare le lettere "doppie" che vi sono contenute.

PARTITA IVA

La Partita IVA è rappresentata da 11 caratteri numerici di cui l'undicesimo è quello di controllo. L'esatta correttezza del codice si calcola nel seguente modo:

0 0 1 0 3 4 6 0 4 5 7

si esclude il codice di controllo (ultima cifra)

0 0 1 0 3 4 6 0 4 5

si moltiplicano per 2 le cifre di posizione di ordine pari (tenendo presente che nei prodotti superiori a 10 si sommano le cifre per ottenere un numero di una sola cifra, es. 12 diventa 1+2=3)

Dispari	0 1 3 6 4
Pari	0 0 8 0 1

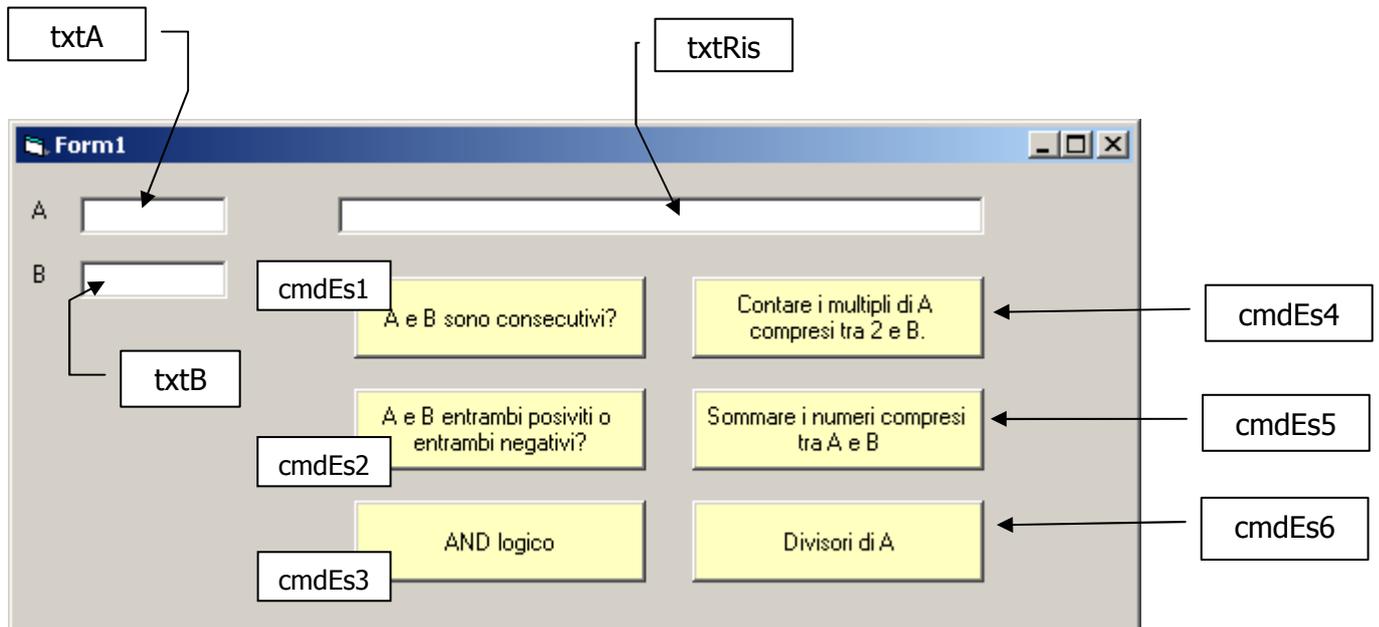
Si sommano le cifre così ottenute

$0+0+1+0+3+8+6+0+4+1=23$

Si prende la cifra dell'unità (3) e se ne fa il complemento a 10

$10 - 3 = 7$ (codice di controllo)

Se il numero ottenuto è uguale alla cifra di controllo della P.I. la P.I. è ESATTA altrimenti è ERRATA



OPERAZIONI RICHIESTE	
Progetto	Creazione del progetto, creazione e gestione degli oggetti, salvataggio
Esercizio 1	Verificare se due numeri A e B sono consecutivi (<i>ad esempio sono consecutivi 5 e 6, 53 e 54, ma non 15 e 18</i>)
Esercizio 2	Verificare se due numeri sono entrambi negativi o entrambi positivi (<i>vedi regola del segno della moltiplicazione</i>)
Esercizio 3	Supponendo di avere in A e B due cifre binarie (bit) simulare l'AND logico
Esercizio 4	Sommare i numeri compresi tra A e B.
Esercizio 5	Contare i multipli di A compresi tra 2 e B
Esercizio 6	Visualizzare in txtRis i divisori del numero A. (<i>per verificare se un numero A è divisibile per un numero B calcolare il resto della divisione – usando l'operatore mod [es. 5 mod 2 = 1] – se il resto è uguale a zero allora A è divisibile per B, altrimenti no</i>)

I risultati devono essere visualizzati nell'oggetto txtRis

Iniziare un nuovo progetto. Aggiungere gli oggetti sotto riportati assegnando il nome specificato. Scrivere il codice associato all'evento *click* dei Command Button.

Svolgere tutti gli esercizi all'interno dello stesso FORM

OPERAZIONI RICHIESTE	
Progetto	Creazione del progetto, creazione e gestione degli oggetti, salvataggio Inserire nel form un controllo ListBox, un textbox un commandButton per l'inserimento di elementi nella lista (accettare soltanto valori compresi tra 0 e 10 - i

	voti di un compito)
Esercizio 1	Estrarre l'elemento selezionato e copiarlo in una label (emettere un messaggio di errore se non è stato selezionato alcun elemento). Verificare se tale voto è sufficiente o insufficiente
Esercizio 2	Calcolare la media dei voti introdotti
Esercizio 3	Contare i voti sufficienti e quelli insufficienti e visualizzare i messaggi: "sono più i voti sufficienti", "sono più i voti insufficienti", "stesso numero"
Esercizio 4	Determinare il numero di voti sotto e sopra la media
Esercizio 5	Visualizzare in txtRis i voti compresi tra 6 e 7 (estremi compresi)

I risultati devono essere visualizzati nell'oggetto txtRis

Svolgere tutti gli esercizi all'interno dello stesso FORM

OPERAZIONI RICHIESTE

Esercizio 1	<i>Data una stringa controllare se il primo carattere è maiuscolo o minuscolo ed emettere un opportuno messaggio.</i>
Esercizio 2	<i>Date due stringhe rappresentanti il cognome e il nome di un utente, ricavarne il codice seguendo le seguenti regole: il terzultimo e il penultimo carattere del cognome seguiti dalla lunghezza del cognome stesso, un trattino, la terza lettera maiuscola del cognome.</i>
Esercizio 3	<i>Contare le parentesi, tonde e quadre, contenute nella stringa data.</i>
Esercizio 4	<i>Data una stringa ricavare una stringa che contenga la stringa data senza alcuno spazio</i>
Esercizio 5	<i>Ricavare una stringa che contenga la stringa data senza alcuna vocale accentata</i>
Esercizio 6	<i>Inserire nel form un controllo ListBox, un textBox e un commandBotton per l'inserimento di elementi nella lista (Non accettare stringhe nulle) Contare gli elementi che iniziano con una stringa data.</i>
Esercizio 7	<i>Invertire i caratteri della stringa (es. "stringa" → "agnirts")</i>
Esercizio 8	<i>Supponendo di avere una lista contenente dati numerici, calcolare la media dei numeri e contare gli elementi che si trovano sopra e sotto tale valore.</i>

I risultati devono essere visualizzati nell'oggetto txtRis

Iniziare un nuovo progetto. Aggiungere gli oggetti sotto riportati assegnando il nome specificato.
Scrivere il codice associato all'evento *click* dei Command Button.

Svolgere tutti gli esercizi all'interno dello stesso FORM

OPERAZIONI RICHIESTE

1	Scrivere una stringa che contenga la stringa data senza cifre numeriche
2	Contare quante volte ricorre una stringa data
3	Sommare le cifre numeriche maggiori di 5 contenute nella stringa data
4	Trasferire in una lista le cifre numeriche e sommarle
5	Scrivere una lista che contenga soltanto uno spazio tra ciascuna parola

I risultati devono essere visualizzati nell'oggetto txtRis

Svolgere tutti gli esercizi all'interno dello stesso FORM

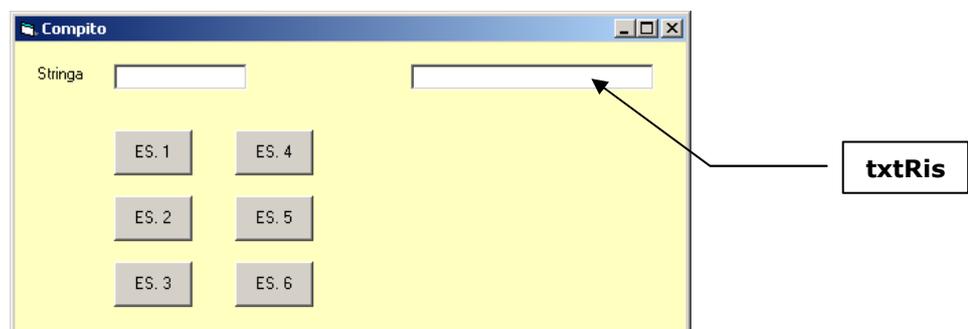
Supponendo di avere un form con i seguenti oggetti:

- txtN
- lst1 lista contenente numeri
- lst2
- txtRis text box per i risultati
- txtA
- txtB

Scrivere il codice relativo alle seguenti richieste:

OPERAZIONI RICHIESTE	
Esercizio 1	<i>Supponendo che ogni elemento della lista sia una cifra binaria, scrivere in txtRis una stringa composta da tutte le cifre scritte alla rovescia.</i>
Esercizio 2	<i>Contare gli elementi non compresi nell'intervallo txtA..txtB</i>
Esercizio 3	<i>Supponendo che ogni elemento della lista sia l'età di un individuo, contare quanti possono votare per la Camera (maggioenni) e quanti possono votare per il Senato (minimo di 25 anni)</i>
Esercizio 4	<i>Trasferire nella seconda lista gli elementi non compresi nell'intervallo [10..20]. Contare gli elementi minori di 10 e quelli maggiori di 20</i>
Esercizio 5	<i>Calcolare la media degli elementi compresi tra 18 e 65</i>
Esercizio 6	<i>Dato un numero (txtN) convertirlo in base 2</i>

I risultati devono essere visualizzati nell'oggetto txtRis



Scrivere il codice relativo alle seguenti richieste:

OPERAZIONI RICHIESTE	
Esercizio 1	<i>Supponendo che la stringa contenga cognome e nome, separati da uno spazio, generare un codice secondo le seguenti regole: secondo carattere, penultimo carattere, posizione dello spazio</i>
Esercizio 2	<i>Verificare che la stringa non contenga segni di punteggiatura</i>
Esercizio 3	<i>Contare gli spazi contenuti nella stringa</i>
Esercizio 4	<i>Scrivere la stringa data priva delle lettere maiuscole</i>
Esercizio 5	<i>Sommare le cifre maggiori di 5 contenute nella stringa data</i>
Esercizio 6	<i>Contare quante volte nella stringa ricorre la sottostringa "la" (Upper o Low Case)</i>

I risultati devono essere visualizzati nell'oggetto txtRis

ESERCIZI SUI GIOCHI

1. *Indovina il numero.* Scopo del gioco è indovinare il numero che in maniera del tutto casuale il computer ha scelto. Una volta cliccato sul pulsante "Nuova partita", il gioco inizia; inserire il numero che si pensa sia stato scelto dal computer, spostare il cursore su di una cella vuota e cliccare sul pulsante "Convalida". Comparirà un messaggio che indicherà se il numero inserito è esatto, è troppo basso o troppo alto, seguire le indicazioni dei diversi messaggi attuando la strategia che più si ritiene idonea, fino a che non verrà indovinato il numero.

L'utente dovrà indovinare il numero, sulla base di risposte del tipo TROPPO ALTO o TROPPO BASSO. Indovinato il numero, o dopo un massimo di 15 tentativi, dovrà apparire la richiesta di ripetizione.

2. *Gioco Mastermind.* Si gioca in due. Il primo giocatore pensa un numero formato da 4 cifre fra loro differenti. Il secondo giocatore deve indovinare il numero. Ad ogni tentativo del secondo giocatore, il primo deve rispondere con una E per ogni cifra esatta e posizionata correttamente e con una A per ogni cifra esatta ma non correttamente posizionata.

Ad esempio: sia da indovinare 1 2 3 4 e il secondo giocatore dica 1 3 4 7. La risposta è E A A (E per la cifra "1" e A A per le cifre "3" e "4").

Scrivere un programma nel quale l'elaboratore simuli il 1° giocatore.

[N = TRUNC(RANDOM*10) in PASCAL è l'istruzione che genera numeri compresi fra 0 e 9 (le cifre del numero); si controlli che le cifre siano fra loro differenti]

3. *Slot-machine.* Si deve simulare il funzionamento di una "slot-machine". La macchina ha 4 simboli a disposizione: \$, !, *, ?

Questi simboli compaiono a gruppi di 3 in modo casuale con possibilità di ripetizione.

Si hanno vincite :

- per tre simboli uguali;
- per 2 \$ e un terzo simbolo qualunque;
- per 2 ! e un terzo simbolo qualunque;

Le vincite sono di valore decrescente nell'ordine che segue: 3\$, 3!, 3*, 3?, 2\$ e x, 2! e x.

Le vincite partono da 10 dollari e sono nell'ordine: 10, 8, 6, 6, 5, 5.

Per ogni gioco non riuscito si perdono 2 dollari. Partendo da un capitale iniziale di 100 dollari eseguire ripetutamente il gioco per un numero massimo di puntate (100 puntate) o fino alla perdita del capitale iniziale. Stampare le singole uscite, la vincita o la perdita ad ogni gioco, il capitale attuale, il numero di giochi eseguiti fino a quel momento.

4. *Gioco degli 11 fiammiferi.* Il gioco comincia con 11 fiammiferi disposti sul tavolo. Due giocatori, alternativamente, possono raccogliere 1, 2 o 3 fiammiferi. Vince il giocatore che costringe l'avversario a raccogliere l'ultimo fiammifero. Determinare la strategia di gioco che permette al giocatore che apre il gioco di vincere sicuramente e realizzare un programma che simuli una partita tra il calcolatore e l'operatore e che permetta al calcolatore di ottenere delle vincite. Si generalizzi quindi il gioco, e quindi il programma, in modo tale che il numero dei fiammiferi sia variabile e possa essere deciso al momento in cui si gioca.
5. *Misuratore di riflessi.* Simulare con un programma un misuratore di riflessi così concepito: generare n segnali acustici ad intervalli irregolari compresi fra un certo tempo minimo e massimo. Dopo ogni segnale acustico l'utente ha 0.3 secondi di tempo per battere un tasto. Alla fine dare la percentuale di errori commessi.
6. *Misuratore di riflessi 2.* Realizzare un misuratore di riflessi così concepito: sullo schermo appaiono di continuo lettere ad intervalli casuali che bisogna ribattere sulla tastiera entro 0.5 secondi pena il conteggio di un errore. In qualche angolo dello schermo si deve vedere continuamente il numero degli errori e la percentuale degli errori.
7. *Timer.* Un segnale acustico avverte che è iniziato un conteggio temporale interno; quando si riterranno passati 30 secondi, dovrà essere premuto un tasto che causerà la cessazione del conteggio. Dovranno venir visualizzati sia il tempo effettivo che il tempo presunto, e le differenze tra i tempi con l'indicazione se in eccesso o in difetto.
8. *Filetto.* Preparare un programma che consenta a due giocatori di utilizzare lo schermo per una partita a tris (filetto).

	X	O
	X	O
X	O	

Nel gioco del filetto vince chi completa per primo una riga, colonna o una diagonale principale di una matrice 3 x 3 con il proprio simbolo ('O' o 'X').

9. *Gioco dell'impiccato.* Il gioco dell'impiccato consiste nell'indovinare una parola segreta pensata da un altro giocatore lettera dopo lettera. Se la lettera annunciata compare nella parola, va evidenziato il posto che occupa, altrimenti si inizia a disegnare la figura di un impiccato, composta da faccia, braccia, gambe e piedi. Se l'impiccato viene completamente disegnato, chi doveva disegnare ha perso. Automatizzare questo popolare giochetto. (Anche non grafico)
10. *Tombola.* Simulare a video il tabellone del gioco della tombola. Ad ogni estrazione i numeri (da 1 a 90), dovranno essere visualizzati nella giusta posizione. Controllare che un numero non sia già uscito.
11. *Battaglia navale.* Realizzare un programma che consenta di condurre il gioco della battaglia navale tra la macchina stessa e l'utente. Stabilire casualmente chi inizia il gioco.