

## Elenco Esercitazioni per Laboratorio di Informatica

TUTTI GLI ESERCIZI, SE NON ALTRIMENTI INDICATO, RICHIEDONO LA CREAZIONE DI UN PROGETTO DI TIPO "WINDOWS FORM APPLICATION", IN AMBIENTE "VISUAL STUDIO 2022 COMMUNITY" E "LINGUAGGIO C#", CHE RISOLVA LA PROBLEMATICA INDICATA NEL TESTO.

### Primo Approccio all'Ambiente Visual Studio

1. Avviare Visual Studio; Creare un Nuovo Progetto per realizzare una App in Linguaggio C# per il S.O. Windows su Personal Computer ("App Windows Forms (.NET Framework)"); Salvare il Progetto sul Desktop del Computer; Visualizzare in modo permanente le finestre "Proprietà" sulla destra e "Casella degli Strumenti" sulla sinistra; Disporre sulla Form: (a) un Pulsante di nome "plsInf" con sfondo di colore Rosso e scritta "INFORMATICA" in Bianco; (b) una Casella di Testo di nome "txtSis" con sfondo Giallo e scritta "SISTEMI E RETI" in Verde; (c) una Etichetta di nome "lblTps" con la scritta "TPSIT" di colore Bianco con sfondo di colore Blu. Eseguire l'App così costruita e verificare come i tre Controlli (Pulsante, Casella di Testo ed Etichetta) reagiscono alle azioni con il Mouse e la Tastiera; infine, dopo aver salvato il progetto, consegnarlo via Mail secondo la procedura vista in laboratorio (eliminare cartelle "bin", "obj", comprimere tutto il progetto in un unico file zip e spedire all'indirizzo mail del prof.).

### Primo Approccio al Codice in Ambiente Visual Studio

2. Prevedere 3 pulsanti che visualizzino in una piccola finestra i messaggi "OGGI C'E' IL SOLE" il primo pulsante; "OGGI E' NUVOLOSO" il secondo, "OGGI STA PIOVENDO" il terzo
3. Prevedere 5 pulsanti denominati Rosso, Verde, Blu, Giallo, Nero. Al clic su uno di essi, lo sfondo di un'unica apposita Label (inizialmente vuota, cioè priva di testo) deve assumere il colore del pulsante cliccato. Inoltre, nella Label stessa, deve apparire una scritta visibile con il nome del colore attivato. Nota Bene: per poter ridimensionare liberamente la Label, disattivare la sua proprietà AutoSize
4. Al clic su un pulsante, sulla Form deve apparire un "arcobaleno". Per realizzarlo utilizzare 7 Label, colorare lo sfondo di ciascuna con uno dei 7 colori dell'arcobaleno e impostare il loro testo con il nome del relativo colore. Nota Bene: per poter ridimensionare liberamente la Label, disattivare la sua proprietà AutoSize

### Variabili, Istruzione di Assegnazione e Input/Output

5. Realizzare un'applicazione Windows che, digitata in INPUT la misura del Lato di un quadrato, calcoli e visualizzi in OUTPUT la misura del Perimetro del quadrato stesso
6. Realizzare un'applicazione Windows che, digitate in INPUT le misure dei due lati L1 ed L2 di un rettangolo, calcoli e visualizzi in OUTPUT le misure dell'Area e del Perimetro del rettangolo stesso
7. Realizzare un'applicazione Windows che, fornito in INPUT un numero intero N, calcoli e visualizzi in OUTPUT: (a) il valore del triplo di N; (b) il valore della potenza  $N^3$

### Sintassi, Espressioni, Operatori, Funzioni

8. Realizzare un'applicazione Windows che, fornita la misura del Lato di un quadrato, calcoli e visualizzi la misura della Diagonale del quadrato stesso
9. Realizzare un'applicazione Windows che, dati due numeri interi N1 ed N2, calcoli e visualizzi il risultato INTERO della divisione fra N1 e N2 (cioè il risultato della divisione troncato all'unità) e il Resto della divisione stessa
10. Realizzare un'applicazione Windows che, fornite le misure dei cateti C1 e C2 di un triangolo rettangolo, calcoli e visualizzi la misura del Perimetro e dell'Area del triangolo stesso
11. Realizzare un'applicazione Windows che, digitato il valore di un numero reale X, calcoli e visualizzi: (a) l'intero che precede il numero reale X; (b) l'intero che viene dopo il numero reale X; (c) l'intero che approssima il numero reale X all'unità più vicina
12. Realizzare un'applicazione Windows che, fornita la misura del Raggio di un cerchio, calcoli e visualizzi la misura dell'Area e della Lunghezza della Circonferenza del cerchio stesso

### Uso degli Oggetti – DateTime e TimeSpan

13. Fornita in input una data (giorno, mese e anno), memorizzarla in un oggetto di classe DateTime e visualizzarla in una Label in formato "esteso", ossia con giorno della settimana e mese espresso a parole
14. Fornito in input un numero N di giorni, calcolare e visualizzare, in formato "esteso", la data che si ottiene quando sono trascorsi N giorni a partire da oggi
15. Fornite in input DUE date, calcolare il numero di giorni che intercorrono fra le due date
16. Calcolare quanti giorni intercorrono fra oggi e il giorno del proprio compleanno, nell'anno in corso
17. Forniti in input un numero N di ore e una data D, calcolare e visualizzare: (a) il numero G di giorni completi inclusi nelle N ore; (b) il giorno e il mese della data che si ottiene G giorni dopo la data D

### Selezione e istruzione if

18. Data l'età N di un individuo, visualizzare di conseguenza il giusto messaggio fra "maggiorenne" e "minorenne".
19. Dato un numero intero N, visualizzare un messaggio che indichi se il numero N è "pari" o è "dispari" (suggerimento: se N è pari, allora il RESTO della divisione  $N / 2$  è zero).
20. Dati due numeri reali A e B, visualizzarli ordinati in senso crescente.

### Selezioni Annidate e in Sequenza

21. Data una temperatura T, visualizzare qual è lo stato fisico dell'acqua alla temperatura T (solido, liquido o gassoso) utilizzando delle istruzioni if annidate e solo condizioni semplici.
22. Dati due numeri reali A e B, convertirli in due numeri positivi e calcolarne la somma. Utilizzare delle istruzioni if poste in sequenza e solo condizioni semplici.
23. Dati i coefficienti a, b, c di una Equazione di Secondo Grado del tipo  $ax^2 + bx + c = 0$  risolverla in tutte e tre le possibili situazioni: (a) Delta < 0: visualizzare il messaggio "nessuna soluzione reale"; (b) Delta = 0: visualizzare il messaggio "due soluzioni reali e coincidenti" e calcolare l'unica soluzione  $x_1=x_2$ ; (c) visualizzare il messaggio "soluzione reali e distinte" e calcolare le due soluzioni reali  $x_1$  e  $x_2$ .

### Selezione Multipla e istruzione switch

24. Chiesto in input il "numero" N di una Carta da Gioco (numero intero da 1 a 10), visualizzare nuovamente il numero della Carta, ma espresso "a parole" (1=Asso, 2=Due, 3=Tre, ..., 8=Donna, 9=Cavallo, 10=Re).
25. Si chiedano in input due numeri reali A e B. Chiesto un terzo numero intero P, calcolare: (A+B) se P=1; (A-B) se P=2; (A\*B) se P=3; (A/B) se P=4 realizzando così una semplice calcolatrice. Se P non è fra i valori previsti, non effettuare nessun calcolo e visualizzare un messaggio di errore.

### Condizioni Composte

26. Una scuola calcio accetta le iscrizioni di ragazzi purché di età compresa fra 6 e 18 anni. Data l'età di un ragazzo, visualizzare il messaggio corretto, fra i messaggi "ISCRIVIBILE" o "NON ISCRIVIBILE". Utilizzare un unico if con una condizione composta.
27. Una partita di calcio termina con il punteggio di A a B (esempio: 3 - 2 ossia A=3 e B=2). Una scommessa prevede la vincita nel caso in cui vinca la squadra di casa (quella che ha fatto A gol) oppure se il numero totale di gol supera 4. Dati i valori di A e B e usando un unico if con condizione composta, verificare se la scommessa è vincente o perdente.

### Variabili Booleane

28. Una Scuola di Pallavolo accetta ragazzi di età compresa fra 14 e 18 anni e solo se di statura superiore a 1,65 cm. Dati i valori Eta e Altezza di un ragazzo, visualizzare se può essere iscritto oppure no. Utilizzare le variabili booleane "EtaAccettabile", "AbbastanzaAlto" e "Iscrivibile" calcolandone il valore logico.
29. Un individuo è considerato "nella media" se la sua Altezza è fra 1,55 e 1,95 cm e il suo Peso è fra 60 e 100 Kg. Dato il Peso e l'Altezza di un individuo, verificare se egli è "nella media". Utilizzare le opportune variabili booleane.

### Istruzione while – Contatori e Accumulatori

30. Dato un numero intero N, visualizzare i numeri da 1 a N, usando dei MessageBox in successione. Se, in particolare, si ha  $N \leq 0$ , allora non visualizzare alcun messaggio.
31. Dati due numeri interi A e B, calcolare la somma S di tutti i numeri compresi fra A e B, inclusi A e B stessi (esempio: se A=3 e B=6 allora  $S = 3+4+5+6 = 18$ ). In particolare, se  $A > B$ , allora la somma è da considerarsi pari a 0.
32. Dato un numero intero N, visualizzare, con dei MessageBox in successione, tutti i numeri pari da 2 fino al numero  $2*N$  (esempio: se N=4, visualizzare 2, 4, 6, 8). In particolare, se  $N \leq 0$ , allora non visualizzare alcun messaggio.
33. Calcolare l'elevazione a potenza intera  $A^B$  con A numero reale e B numero intero. Si ricordi che l'elevazione a potenza intera è così definita:  $A^B = A * A * A * \dots$  per B volte. In particolare, se  $B = 0$ , allora  $A^B = A^0 = 1$  (suggerimento: inizializzare a 1 la variabile accumulatore). Facoltativo: implementare anche il caso particolare  $B < 0$ : in tal caso si ha  $A^B = (1 / A)^{-B}$ . Ad esempio  $2^{(-2)} = (1/2)^2 = 1/4 = 0,25$

### Istruzione do-while

34. Dato un numero intero N (con  $N > 0$ ), calcolare la somma S dei primi N numeri (esempio: se N=4, allora il risultato è  $S=10$ , ossia  $S = 1 + 2 + 3 + 4 = 10$ ). Usare necessariamente una iterazione do-while.
35. Dati due numeri interi A e B, visualizzare i numeri da A a B (A e B compresi), usando dei MessageBox in successione (Esempio: se A=3 e B=6 allora visualizzare 3, 4, 5, 6). Usare necessariamente una iterazione do-while. Se, in particolare, si ha  $A > B$ , allora non procedere nella visualizzazione dei numeri e visualizzare, invece, un messaggio di errore del tipo "DATI NON VALIDI".

36. Calcolare la somma dei primi M multipli di un numero intero positivo N (esempio: se  $N=3$  e  $M=7$  multipli, allora  $S=6+9+12+15+18+21+24$ ). Usare necessariamente una iterazione do-while. Nei casi particolari  $M \leq 0$  ed  $N \leq 0$ , non procedere nel calcolo e visualizzare, invece, un messaggio di errore del tipo "DATI NON VALIDI".
37. Dati due numeri interi A e B (con  $A \leq B$ ), visualizzare, con dei MessageBox in successione, i numeri che si ottengono a partire da B, sottraendo, ad ogni iterazione, un dato passo P. La visualizzazione deve terminare quando il numero ottenuto diventa minore o uguale ad A (esempio: se  $A=10$ ,  $B=20$  e  $P=3$  allora visualizzare: 20, 17, 14, 11, 8). Usare necessariamente una iterazione do-while. Se, in particolare, si ha che  $A > B$ , allora, prima di procedere, SCAMBIARE fra loro i valori di A e B in modo che, anche in questo caso, A risulti minore di B.

#### Istruzione for

38. Dato un numero intero positivo N, contare da 1 a N, usando dei MessageBox in successione e l'istruzione FOR..
39. Dato un numero intero positivo N compreso fra 1 e 10, visualizzarne la "tabellina" (ossia i valori  $N \times 1$ ,  $N \times 2$ ,  $N \times 3$ , ...  $N \times 10$ ) con dei MsgBox in successione (es.: Se N è 3 visualizzare dei MsgBox con 3, 6, 9, 12, 15, 18, 21, 24, 27, 30).
40. Dati due numeri interi positivi A e B (con  $A < B$ ), calcolare la somma di tutti i numeri PARI dal numero  $2 \times A$  al numero  $2 \times B$  (es.: se  $A=5$  e  $B=9$ , calcolare  $10+12+14+16+18 = 70$ ).
41. Dato un numero intero positivo N, visualizzare, con dei MsgBox in successione, tutti i numeri da N a 1, ad eccezione di quelli divisibili per un dato numero P. Al termine della visualizzazione, mostrare anche la somma dei numeri elencati (Es.: se  $N=10$  e  $P=3$  allora visualizzare 10, 8, 7, 5, 4, 2, 1 e, come somma 37)

#### Iterazioni senza Conteggio

42. Dati due numeri interi positivi A e B, calcolarne il Minimo Comune Multiplo (mcm), utilizzando il metodo delle somme successive. Il metodo è analogo a quello del "MCD con sottrazioni" solo che anziché sottrarre, si deve accumulare. Suggerimento: si devono usare altre due variabili SommaA e SommaB inizializzandole rispettivamente con i valori di A e di B. Se  $A > B$  allora si aggiunge B a SommaB, altrimenti si aggiunge A a SommaA. L'elaborazione termina quando SommaA diventa uguale a SommaB. Al termine, il mcm è in SommaA (o in SommaB, tanto sono uguali).
43. Dato un numero intero N, visualizzare (con MsgBox in successione) i numeri che si ottengono a partire da N e raddoppiando ogni volta N stesso. Fermare la visualizzazione quando il valore ottenuto supera un dato valore limite MAX (Es.: se  $N=3$  e  $MAX=50$ , visualizzare 3, 6, 12, 24, 48, 96).
44. Dati due numeri interi positivi N e D, calcolare quante volte N è divisibile per D (esempio: se  $N=48$ ,  $D=2$ , visto che  $48/2=24$ ;  $24/2=12$ ;  $12/2=6$ ;  $6/2=3$ , allora 2 divide il 48 per 4 volte).
45. La celebre Serie di Fibonacci inizia con "1, 1" e in essa, ogni numero è la somma dei due numeri che lo precedono (1,1,2,3,5,8,13,21,34,55,...). Generare la Serie di Fibonacci fino al primo valore della serie che superi un dato numero N e visualizzarla con dei MessagBox successivi.

#### Liste di Dati e funzione InputBox

50. Data una sequenza di numeri interi positivi, terminata dal numero 0 (zero), calcolarne la somma. Si effettui l'input ripetendo in un ciclo la funzione InputBox.
51. Data una sequenza di numeri interi positivi, terminata dal numero 0 (zero), contare quanti numeri maggiori di 10 sono presenti nella sequenza. Es.: se la sequenza è: 6 - 9 - 24 - 12 - 7 - 0 allora il risultato è: 2. Si effettui l'input ripetendo in un ciclo la funzione InputBox.
52. Data una sequenza di numeri reali positivi, terminata dal numero 0 (zero), calcolarne la media aritmetica (0 non è un dato, fa solo da "fine lista"). Si effettui l'input ripetendo in un ciclo la funzione InputBox. Si ricordi che la Media Aritmetica si ottiene sommando tutti i dati e dividendo tale somma per il numero dei dati stessi.
53. Data una sequenza di numeri interi (positivi o negativi), terminata dal numero 0 (zero), calcolarne la somma SP dei soli numeri positivi della lista e la somma SN dei soli numeri negativi della lista. Si effettui l'input ripetendo in un ciclo la funzione InputBox.

#### Tipo String - Concatenamento e Lunghezza

54. Dati il Nominativo di un calciatore e un numero intero M (che indica un minuto in una partita di calcio), costruire e visualizzare una stringa per avvisare che, al minuto dato, il calciatore ha segnato (ad esempio: se Nominativo = "Messi" e  $M = 56$ , costruire e visualizzare la stringa "al 56° minuto ha segnato Messi").
55. Data una parola P, costruire e visualizzare una stringa del tipo "La parola <P> è lunga <N> caratteri" (ad esempio: se la parola P = "casa", allora visualizzare la stringa "La parola casa è lunga 4 caratteri").
56. Date due parole P1 e P2, visualizzare, in una Label, la parola più lunga fra P1 e P2. Se le due parole hanno stessa lunghezza, visualizzare, nella Label, il messaggio "Le due parole hanno stessa lunghezza pari a <N> caratteri", dove al posto di <N> deve apparire la lunghezza comune delle due stringhe.

### Costruzione di Stringhe

57. Dato un numero intero positivo N, costruire (e visualizzare in una Label) una stringa composta da una sequenza di N caratteri "X" (es.: se N = 7, allora visualizzare la stringa "XXXXXXX").
58. Dato un numero intero positivo N, costruire (e visualizzare in una Label) una stringa contenente tutti i suoi divisori interi, separati da virgola (es.: se N = 24, allora costruire e visualizzare la stringa "1, 2, 3, 4, 6, 8, 12, 24").
59. Dato un numero intero N positivo o nullo, costruire e visualizzare una stringa contenente la sua rappresentazione nel sistema di numerazione binario (es.: se N = 24, allora costruire e visualizzare la stringa "11000").

### Confronto fra Stringhe

60. Date due parole P1 e P2 (composte solo da lettere maiuscole), visualizzare, in una Label, quella "minore" (ossia quella che "viene prima") secondo ordine alfabetico. Se le due parole risultano uguali, visualizzare "SONO UGUALI".
61. Data una parola di riferimento P0, chiedere una seconda parola P in una apposita TextBox e, tramite un apposito pulsante, determinare se essa è "maggiore" di P0: se lo è, aggiungerla al termine del contenuto di una Label presente sulla Form. In tal modo, digitando nella TextBox altre parole e agendo più volte sul pulsante, è possibile processare più parole, aggiungendo alla Label solo quelle maggiori di P0. Separare le parole aggiunte alla Label, con il carattere ":"

### Accesso al Singolo Carattere di una Stringa

62. Dati un Nome e Cognome, visualizzarne le iniziali puntate. Es. "Mario" / "Rossi" deve produrre "M.R."
63. Data una Parola e un numero intero K, visualizzare il carattere che, nella parola, si trova in posizione K (si intende che 0 è la posizione del primo carattere, 1 quella del secondo, ecc.).

### Scansione di una Stringa

64. Data una Parola visualizzarne, con dei MessageBox successivi, tutti i caratteri, uno per volta, dall'ultimo al primo.
65. Data una frase e dato un carattere, contare quante volte il carattere dato è presente nella frase.
66. Data una stringa contenente un Numero Binario (ossia solo caratteri "0" o "1"), contare quanti "1" e quanti "0" sono presenti in esso.
67. Data una frase e dato un carattere, verificare se nella frase esiste o no il carattere dato e, se sì, visualizzare la posizione della sua prima occorrenza. Es.: se Frase = "Ciao mamma!" e Carattere = "a" il programma deve fornire: "SI, esiste in posizione 2" (nota bene: è preferibile usare un while, anziché un for ... perché?)

### Tipo char e Codice ASCII

68. Consentire all'utente di calcolare: (a) dato un Carattere, il corrispondente Codice ASCII; (b) dato un numero intero indicante un Codice ASCII, il corrispondente Carattere. Prevedere, sulla form, due aree distinte per le due operazioni (quindi, due pulsanti diversi, textbox distinte per input/output, ecc.).
69. Dati 2 numeri interi A e B (con A<B), visualizzare, con dei MessageBox successivi, tutti i caratteri aventi codice ASCII da A a B (A e B inclusi).
70. Dato un Nome (scritto interamente con lettere minuscole), costruire una stringa contenente il nome stesso, ma scritto interamente con le corrispondenti lettere maiuscole (non è consentito l'uso del metodo toUpper).
71. Data una parola P e un numero intero positivo N (con N <= 10), costruire e visualizzare una stringa ST contenente la parola stessa ma "crittografata" secondo la seguente regola: a ciascun carattere viene sostituito il carattere che, rispetto al primo, ha codice ASCII aumentato di N (esempio: se N = 2 e P = "CASE" allora ST = "ECUG")

### Metodi del Tipo string

72. Data una frase, verificare se in essa "è presente" o "non è presente" la sequenza di caratteri "ciao" (minuscolo).
73. Data una parola composta da un numero pari di caratteri, visualizzarne separatamente la prima metà e l'ultima metà, entrambe convertite in lettere maiuscole (esempio: se la parole è "Scudetto" allora visualizzare "SCUD" e "ETTO").
74. Data una frase F e una parola P (presente una sola volta nella frase F), costruire una nuova frase che sia come F ma priva della parola P (esempio: se F = "la mamma stira i pantaloni" e P = "stira", visualizzare "la mamma i pantaloni").
75. Data una frase F e una parola P, contare quante volte P è presente in F e generare una nuova frase, simile ad F, ma convertita in lettere maiuscole e in cui, al posto di ogni occorrenza della parola P, sia presente una sequenza di cinque "x" (esempio: se F = "capitano, mio capitano!" e P = "capitano", allora P, in F, è presente 2 volte e la nuova frase dovrà essere "xxxxx, mio xxxxx!").

### Esercitazioni di Riepilogo sul Tipo String

76. Data una parola P, composta solo da lettere maiuscole, costruire e visualizzare una stringa che sia come P, ma priva di tutte le lettere "A" (es.: se Parola = "AMARENA", allora visualizzare "MREN").

77. Data una parola P, composta solo da lettere maiuscole, costruire e visualizzare una stringa PV contenente tutte le vocali di P e una stringa PC contenente tutte le consonanti di P (se Parola = "STIPENDIO", allora PV = "IEIO" e PC = "STPND").
78. Dato un numero intero positivo N, costruire e visualizzare una stringa contenente la sua rappresentazione nel sistema di numerazione in base 16 (sistema esadecimale). Si ricordi che: (a) la conversione da decimale ad altra base B si ottiene dividendo N per B fino all'annullamento di N e considerando i resti in senso inverso; (b) l'esadecimale prevede 16 cifre: 0, 1, ..., 9, A, B, C, D, E, F.
79. Date due parole P1 e P2, determinare quale delle due è "minore" (ossia quella che "viene prima") secondo ordine alfabetico. Individuare la parola minore utilizzando sia il confronto ordinale che quello non ordinale (due pulsanti distinti). Individuare almeno un caso in cui i due tipi di confronto generano un risultato diverso.
80. Usando un'unica TextBox e un singolo Pulsante è possibile inserire, una dopo l'altra, una successione di N parole. Al termine della digitazione di tutte le N parole, visualizzare, in una label, un messaggio che indichi se E' VERO o se NON E' VERO che le parole inserite sono ordinate alfabeticamente.

### Esercitazioni di Riepilogo

81. Dato l'elenco dei voti di fine anno (10 materie) di un singolo alunno, determinare se è bocciato (4 insufficienze o più), se il giudizio è sospeso (3 insufficienze o meno) o se è promosso (nessuna insufficienza). Effettuare l'input e il calcolo in un unico evento click e utilizzare la funzione InputBox per l'input dei 10 voti.
82. Data una lista di temperature (numeri interi) rilevate durante il trascorrere di un singolo giorno, verificare quante volte accade che la temperatura è aumentata fra una rilevazione e la precedente. Verificare, inoltre se la temperatura è scesa almeno una volta sotto lo zero.
83. Dato un numero N, verificare se esso è o non è un Numero Primo. Si consiglia verificare se N è divisibile per 2, per 3, ..., per N-1: se non è divisibile per nessuno di essi, allora è un Numero Primo.
84. Dato un numero N, scomporlo in fattori primi (esempio: se N=60 allora visualizzare, con MessageBox successivi, i numeri 2, 2, 3, 5 perché  $60/2 = 30/2 = 15/3 = 5/5 = 1$ ). Si consiglia di provare a dividere N per tutti i numeri da 2 in poi.
85. Senza utilizzare le funzioni della libreria Math e, quindi, senza usare la funzione Math.Sqrt(<argomento>), determinare la radice quadrata di un numero X (compreso fra 0 e 100) con un errore inferiore a un dato errore ERR introdotto in input (es. 0,0001). Utilizzare una tecnica basata su approssimazioni successive. Si ricordi che se  $a^2 < X < b^2$  allora la radice quadrata di X è sicuramente compresa fra a e b, quindi, si può provare a ridurre l'intervallo (a,b) dimezzandolo.

### Vettori - 1

86. Si desidera gestire una lista di Numeri Interi (massimo 10 numeri) memorizzandola in un Vettore. Con degli appositi pulsanti, si consenta di: (a) digitare un Nuovo Numero e aggiungerlo alla fine della lista presente nel Vettore; (b) copiare tutti i dati del Vettore (separati da uno spazio) in una Stringa e visualizzarla, mostrando così il contenuto del Vettore; (c) calcolare la Somma dei dati presenti nel Vettore; (d) svuotare il Vettore. Dopo ogni aggiunta, deve essere aggiornata una Label che visualizzi il Numero di Dati presenti nel vettore.
87. L'utente inserisce i Nominativi presi da un Elenco (max 20 nominativi) indicando, per ogni nominativo digitato, se si tratta di un Maschio o una Femmina. I nominativi dei Maschi devono essere memorizzati in un vettore VetM, mentre i nominativi delle Femmine in un vettore VetF. Solo al termine dell'input, un click su un apposito pulsante, consente di visualizzare, in due Stringhe distinte, i due elenchi presenti in VetM e VetF, indicando con due apposite Label, anche il Numero di Maschi e il Numero di Femmine.
88. Una fornitura di materiale prevede la consegna di una serie di Casse chiuse. Per ogni cassa consegnata, ne viene digitato il peso in Kg (numero intero). Più casse possono avere lo stesso peso. In un vettore "Pesi" è necessario man mano memorizzare i pesi delle casse, in modo che ogni peso sia memorizzato solo una volta. In un secondo vettore "NumeroCasse" è necessario aggiornare il conteggio di quante casse hanno peso pari a quello indicato corrispondentemente nel vettore "Pesi" (esempio: se i pesi di 6 casse sono 4, 5, 5, 4, 8, 5 allora il vettore Pesi è 4, 5, 8 e il vettore NumeroCasse deve contenere corrispondentemente i conteggi 2, 3, 1 cioè 2 casse da 4 Kg, 3 casse da 5 Kg e 1 cassa da 8 Kg). Al termine dell'input visualizzare, in due Stringhe, i due vettori "Pesi" e "NumeroCasse" e si calcoli il peso complessivo della fornitura.

### Vettori - 2

89. Dato l'elenco dei voti di fine anno (10 voti) di un singolo alunno, memorizzarlo in un vettore VetVoti. Consentire, in qualsiasi momento, di riportare il contenuto del vettore in una stringa per visualizzarlo in una Label. Infine, determinare se l'alunno è "bocciato" (4 insufficienze o più), se il giudizio è "sospeso" (3, 2 o 1 insufficienza) o se è "promosso" (0 insufficienze) visualizzando l'esito in una apposita Label. Impedire l'inserimento di voti non compresi fra 1 e 10.

90. Data una lista di temperature (numeri interi) rilevate durante il trascorrere di un singolo giorno, memorizzarla in un vettore VetTemp. Consentire, in qualsiasi momento, di riportare il contenuto del vettore in una stringa per visualizzarlo in una Label. Verificare quante volte accade che la temperatura aumenta fra una rilevazione e la precedente. Verificare, inoltre se la temperatura è scesa almeno una volta sotto lo zero.
91. Data una lista di Date (da gestire come oggetti di classe DateTime), memorizzarla in un vettore VetDate (i cui elementi sono oggetti di classe DateTime). Consentire, in qualsiasi momento, di riportare le date del vettore, in una stringa, per visualizzarle in una Label. Generare un secondo vettore VetCont che, in posizione 1, riporta quante date hanno per mese Gennaio; in posizione 2, quante date hanno mese Febbraio; ... in posizione 12, quante date hanno mese Dicembre. Visualizzare VetCont in una stringa nella forma: < mese>=< conteggio>; < ecc.) (Es.: "1=5; 2=7; ... 12=6")

#### Sottoprogrammi

92. Specificato il giorno G e il mese M di una data dell'anno, visualizzare nuovamente giorno e mese, ma con il mese espresso in lettere (es. se G = 21 e M = 5, visualizzare "21 Maggio"). Per risolvere il problema si introduca un sottoprogramma che, ricevendo un numero intero N compreso fra 1 e 12 (indicante il numero del mese), restituisca una stringa contenente il mese corrispondente espresso in lettere (es. se N = 1 restituire "Gennaio", se N = 2 restituire "Febbraio", ecc.). Creare due versioni del programma: la prima deve usare un sottoprogramma di tipo FUNZIONE, la seconda uno equivalente di tipo PROCEDURA.
93. Dati due numeri A e B si visualizzino, con dei MessageBox in successione, tutti i numeri da A a B in senso crescente. Nel caso in cui A sia maggiore di B, prima di procedere alla visualizzazione, si richiami un opportuno sottoprogramma che, agendo su A e B, ne scambii i valori (es. se prima A = 12 e B = 7 allora, dopo la chiamata, si ha A = 7 e B = 12). Si presti attenzione al tipo di parametrizzazione da utilizzare nel sottoprogramma (out, ref, ecc.)
94. Uno stesso prodotto viene offerto sia nel negozio online di Paperino sia nel negozio online di Topolino: Paperino lo offre a un prezzo P1 con uno sconto percentuale di S1 (numero intero, es. 20 significa sconto 20%), mentre Topolino a P2 con sconto S2. Dati in input P1, S1 e P2, S2 determinare i reali prezzi di vendita e indicare quale negozio è più conveniente. Introdurre un opportuno sottoprogramma che riceva un prezzo P e uno sconto S, restituendo il reale prezzo di vendita.
95. Dato un numero intero Num, visualizzare l'elenco di tutti i numeri primi presenti fra 1 ed Num. Per risolvere il problema, si introduca un sottoprogramma che, ricevendo un intero N, restituisca, come risultato, se esso è o non è un numero primo (suggerimento: per far questo, il sottoprogramma deve verificare tutte le divisioni  $N/2, N/3, \dots, N/(N-1)$  e, se sono tutte con resto non nullo, allora N è un numero primo).

#### Algoritmi di Base - 1: Scansione, Somma, Ricerca, Minimo

96. Dato l'Elenco dei Voti conseguiti da uno Studente a fine anno (un voto per materia, max 10 materie), memorizzarlo in un vettore Vet e offrire le seguenti opzioni: (a) comporre l'intero contenuto del vettore in una stringa e visualizzarlo in una Label; (b) determinarne la Media Aritmetica dei voti allo scopo di individuare in quale "fascia di credito" lo studente si colloca: le possibili fasce sono: fascia A (< 6); fascia B (>=6 e <8); fascia C (>=8). (c) determinare il Voto Massimo conseguito dallo studente; (d) verificare se "è vero" o "non è vero" che fra i voti dello studente è presente almeno un 3.
97. Una Lista di Importi in euro (max 20 importi) documenta i versamenti fatti su un conto corrente bancario. Effettuare l'input degli importi, memorizzarli in un vettore V e offrire le seguenti opzioni: (a) visualizzare il contenuto del vettore in una Label; (b) calcolare la Somma degli importi versati; (c) dato un importo X, verificare se Esiste nel vettore e determinarne la Posizione; (d) determinare Posizione e Valore dell'Importo Massimo versato.
98. Un Gruppo di Atleti (max 15) gareggia in una competizione di Lancio del Disco durante la quale, ciascun atleta effettua un singolo lancio che viene misurato in metri (es. 50,65 m). Chiesti in input i Nominativi degli Atleti e le Misure del loro Lancio memorizzarle in due vettori VetNomi e VetMisure e offrire le seguenti opzioni: (a) visualizzare, in un'unica Label, il contenuto dei due vettori, nella forma: < nome>:< misura> - < nome>:< misura> - ecc. (b) calcolare la Media Aritmetica delle misure; (c) determinare il Nome e la Misura dell'atleta che ha effettuato il lancio più lungo; (d) dato il Nome di un Atleta, visualizzare, in un'altra Label, l'elenco dei soli atleti che hanno ottenuto misure superiori alla sua.

#### Algoritmi di Base - 2: Caratteristica, Tutti con Caratteristica, Ins./Elim. su Vettore

99. Dato un Elenco di Nominativi (max 20), inizialmente vuoto, memorizzarlo in un vettore Vet offrendo le seguenti opzioni: (a) Aggiungere un Nuovo Nominativo alla fine dell'elenco; (b) Inserire un Nuovo Nominativo in una data posizione P (con P compreso fra 0 e N-1); (c) Eliminare il Nominativo in una data posizione P; (d) Azzerare l'elenco; (e) Visualizzare l'intero Elenco in un'unica Label, separando i nomi con una virgola; (f) Data una lettera (da 'A' a 'Z'), visualizzare, in una apposita Label, solo i nominativi che iniziano per tale lettera; (g) Dato un numero intero X, verificare se è vero o no che tutti i nominativi sono più lunghi di X caratteri.
100. Un giocatore riceve inizialmente 10 carte (si tratta di Carte da Poker o "Francesi", da un mazzo di 52 carte). L'input di una carta avviene digitando, in due TextBox distinte, il "Numero" (A, 2, 3, ..., 9, 0, J, Q, K) e il "Palo" (C, Q, F, P) della

carta. Le 10 carte vanno memorizzate in due vettori di stringhe VetCarta e VetPalo: il primo memorizza il "Numero" della carta e il secondo ne memorizza il Palo corrispondente (Es.: se le carte sono "3 di Fiori", "5 di Picche", "Asso di Cuori", "Re di Quadri", allora VetCarta conterrà ["3", "5", "A", "K"] e VetPalo conterrà ["F", "P", "C", "Q"]). Quando l'utente digita in input una Nuova Carta, ne specifica anche la posizione di inserimento fra 0 e N (0 significa prima della prima carta, N significa dopo l'ultima carta). Una volta completato l'input delle 10 carte, si dia la possibilità di "Scartare" una carta, eliminandola dal vettore e, al tempo stesso, di inserirne una Nuova, indicandone sempre la posizione di inserimento. Dopo ogni inserimento/scarto, una Label deve essere riempita con l'intero contenuto del vettore, allo scopo di visualizzare tutte le carte. Tramite appositi pulsanti permettere, inoltre, di: (a) verificare se è vero o no che tutte le carte appartengono ad uno stesso Palo; (b) dato un Numero di Carta (es. "A", "5", ecc.), verificare se fra le carte c'è un gruppo di almeno 3 carte con quel numero. Facoltativo: impedire di inserire una carta, se essa è già presente nel vettore.

101. Durante una Competizione Automobilistica, i piloti percorrono, in momenti diversi, uno stesso tracciato, realizzando ciascuno un tempo di percorrenza espresso in secondi (numero intero). Ogni volta che viene completata una prestazione, si chiedi in input il nominativo del Pilota e il Tempo che ha ottenuto: tali dati vanno memorizzati rispettivamente in due vettori VetPiloti e VetTempi, inserendo man mano le prestazioni in modo da mantenere sempre un ORDINAMENTO secondo i tempi (all'inizio il tempo minore), costituendo così la Classifica della competizione. Può anche accadere che, dopo un certo tempo dal completamento di una prestazione, un pilota venga squalificato, per cui dovrà essere possibile rimuoverlo dalla classifica. Dopo ogni inserimento (o squalifica), visualizzare in una Label l'elenco aggiornato di tutte le prestazioni, nella forma "<Posizione> - <Pilota>: <Tempo>". Si consenta di specificare un Tempo Limite, al di sotto del quale un pilota è "qualificato". Si verifichi se è vero o no che tutti i piloti sono qualificati. Infine, in una apposita Label, si riportino i nominativi dei soli piloti "qualificati".

#### Approfondimento sui Controlli (Parte 1)

102. Si disponga sulla Form un controllo di classe Label, lo si renda liberamente dimensionabile e, tramite appositi pulsanti si consenta di attuare su di esso le seguenti azioni: Nascondere/Visualizzare (con 1 solo pulsante); Ingrandire/Rimpicciolire (con 4 pulsanti: ogni click 10 pixels); Spostare (con 4 pulsanti: ogni click 10 pixels). Si imposti correttamente l'Ordine di Messa a Fuoco degli oggetti sulla Form.
103. Si disponga sulla Form un controllo di classe TextBox e, tramite appositi pulsanti si consenta di attuare su di esso le seguenti azioni: Abilitare/Disabilitare (con 1 solo pulsante); Spostare la TextBox nell'angolo in alto a sinistra della Form; Spostare la TextBox nell'angolo in basso a destra della Form; Spostare la TextBox al centro della Form. Si imposti correttamente l'Ordine di Messa a Fuoco degli oggetti sulla Form.
104. Si disponga sulla Form un controllo di classe Label, lo si renda liberamente dimensionabile e, tramite appositi pulsanti si consenta di attuare su di esso le seguenti azioni: Cambiare il Testo (usare una TextBox per l'input del nuovo testo); Cambiare l'Allineamento (con 1 solo pulsante a rotazione: Sinistra, Centro, Destra); Cambiare colore (con 1 solo pulsante a rotazione: Nero su Bianco, Rosso su Blu, Giallo su Verde). Dopo aver fatto click su uno dei pulsanti, il programma deve posizionare automaticamente il "Fuoco" sul pulsante dell'opzione successiva (dopo l'ultima, la prima).

#### Approfondimento sui Controlli (Parte 2)

105. Attribuire alla Form il nome frmAvvio e disporre su di essa una TextBox e un Button. Utilizzando l'evento Load della Form stessa, impostare la TextBox in modo che in essa appaia, già all'avvio, la parola "INFORMATICA". Il pulsante, quando premuto, deve mettere a fuoco la TextBox e selezionare automaticamente tutto il testo in essa presente. Ogni volta che l'utente "abbandona" la TextBox (ossia la TextBox perde il "fuoco") è necessario verificare se il contenuto della TextBox è ancora "INFORMATICA" e, solo nel caso in cui il contenuto sia stato modificato, visualizzare il messaggio "La TextBox non contiene più la parola INFORMATICA".
106. Si disponga sulla Form un controllo di classe Label, lo si renda liberamente dimensionabile e con colore di fondo Rosso. Sfruttando l'evento KeyDown della Form riconoscere la pressione dei tasti con le 4 frecce direzionali e, ad ogni freccia premuta, si sposti la Label di un pixel nella relativa direzione. Se la Label raggiunge un bordo della Form, essa si deve bloccare, ossia si devono impedire ulteriori spostamenti nella direzione del bordo stesso.
107. Si disponga sulla Form un controllo di classe TextBox, lo si renda adatto a digitare un testo lungo (su più righe, senza ritorno a capo automatico e con la presenza di 2 scrollbar). Quindi, si consenta, in essa, solo la digitazione di: Lettere Maiuscole ("A", ... "Z"), lo Spazio, il tasto di correzione BackSpace (codice Ascii = 8) e il tasto Invio (codice Ascii = 13). Per filtrare i caratteri si usi l'evento KeyPress.

#### Liste Tipizzate e classe List<T>

108. Si gestisca una Lista di Nomi memorizzandola in un oggetto di classe List<T>. Si consenta di: (a) aggiungere un nome alla lista; (b) data una posizione (la prima posizione è 0), eliminare il nome in tale posizione; (c) visualizzare in

- un'unica Label, tutti i nomi della lista separati dal carattere virgola; (d) azzerare la lista. Dopo ogni modifica della lista, deve essere visualizzato automaticamente, in una apposita Label, quanti Nomi sono presenti nella lista.
109. Dato un Insieme di Numeri Reali memorizzarli in un oggetto di classe List<T> e consentire di: (a) calcolarne la Somma; (b) calcolarne la Media Aritmetica; (c) visualizzare in un'unica Label, l'elenco di tutti i numeri, separati da una barra ( / ); (c) dato un numero X, visualizzare in un'unica Label, l'elenco dei soli numeri maggiori di X, separati da una barra ( / ).
110. Si gestisca una Lista di Numeri Interi tramite un oggetto di classe List<T> consentendo di: (a) aggiungere un nuovo numero, ma solo se esso non è già presente nella lista; (b) dato un numero X, eliminarlo dalla lista, se esso è presente; (c) data una posizione P, sostituire il numero presente in posizione P, con un altro numero dato; (d) visualizzare in un'unica Label, tutti i numeri della lista, separati da trattino, ma in ordine inverso a quello di inserimento; (e) svuotare la lista. E' consentito utilizzare solo i membri Add, Count, Clear e RemoveAt e l'accesso agli elementi della classe List<T>.

### ListBox e ComboBox

111. Si gestisca una Lista di Nomi, inizialmente vuota, memorizzandola e visualizzandola in una ListBox. Si utilizzi una ComboBox che, opportunamente inizializzata, elenchi le seguenti scelte: Aggiungere, Eliminare, Posizione, Azzerare. Un unico pulsante, in base alla selezione effettuata con la ComboBox, realizzi l'operazione scelta: "Aggiungere": aggiungere un nome alla fine della lista; "Eliminare": selezionato un nome nella ListBox, eliminarlo; "Posizione": chiesto un numero intero P, evidenziare il nome in posizione P (0 significa prima posizione); "Azzerare": Svuotare la lista. Dopo ogni modifica della lista, deve essere visualizzato automaticamente, in una Label, quanti Nomi sono presenti nella lista.
112. Si consenta all'utente di inserire in una ListBox, l'elenco delle Squadre di Calcio della Serie A Italiana. Si chieda in input una partita (ossia, la Squadra1, la Squadra2, Gol1, Gol2) e si determini il Segno del Totocalcio (1, X o 2) della partita. Per facilitare l'utente, l'input delle due squadre deve avvenire utilizzando due ComboBox: nel loro "elenco a discesa", esse devono riportare automaticamente le Squadre precedentemente inserite nella ListBox.
113. Si gestisca, in tempo reale, la Classifica di una gara di Formula 1. Inizialmente vengono inseriti, in una ListBox, i Nomi dei Piloti, nell'ordine di partenza. Si consenta di: (a) selezionato un Pilota, farlo avanzare (o arretrare) di una posizione; (b) selezionato un Pilota, notificarne il ritiro, rimuovendolo dalla classifica e spostandolo in un'altra ListBox, dedicata ad elencare i ritirati; (c) scelto, tramite una ComboBox, uno dei Piloti, lo si evidenzi nella ListBox della classifica e si visualizzi il Punteggio che, in quel momento, maturerebbe in base alla classifica corrente (i punteggi in Formula1 sono, nell'ordine: 25, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0 ...). A gara terminata, nella ListBox della classifica, dovranno comparire, accanto a ciascun Nome di Pilota, anche la sua Posizione e il suo Punteggio finali.

### Ordinamento dei Dati e algoritmo Selection Sort

114. Dato un Elenco di Alunni (max 30) memorizzarlo in un Vettore e, dopo ogni input, visualizzare il contenuto del Vettore in una ListBox. Ad input completato, effettuare l'Ordinamento Crescente del Vettore con l'algoritmo di ordinamento Selection-Sort e visualizzare il Vettore così ordinato in una seconda ListBox. Introdurre opportuni sottoprogrammi per l'Ordinamento e per la Visualizzazione.
115. Effettuare l'input della classifica di un Campionato di Calcio a 20 squadre, memorizzandola in due vettori VetSquadre e VetPunteggi. Utilizzando l'algoritmo di ordinamento Selection-Sort, consentire di ordinare la classifica o "per Squadre" o "per Punteggio", visualizzando il risultato in una ListBox (ogni riga del tipo "Squadra - Punteggio"). Introdurre opportuni sottoprogrammi per l'Ordinamento e per la Visualizzazione.

### Ricerca Binaria

116. Estendere l'esercizio N. 114 (primo esercizio su "Selection Sort") come segue: solo dopo aver effettuato l'ordinamento, chiedere in input un Nominativo e determinarne la posizione nel Vettore ordinato, applicando la tecnica della Ricerca Binaria. Segnalare, con un appropriato messaggio, il caso in cui il Nominativo dato, non è presente nel Vettore.
117. Estendere l'esercizio N. 115 (secondo esercizio su "Selection Sort") come segue: solo se l'ordinamento corrente è "per Punteggio", chiedere in input un possibile punteggio P e, applicando la Ricerca Binaria, determinare e visualizzare in quale posizione si collocherebbe in classifica una squadra che avesse ottenuto il punteggio P.

### Bubble Sort e Inserimento in Ordine

118. Dato un Elenco di Numeri Interi (max 20) memorizzarlo in un Vettore. Durante l'input, visualizzare costantemente il Numero di Dati inseriti fino a quel momento e il Numero Massimo di dati inseribili separati da una barra (esempio: 5 / 20). Ad input terminato, attivare degli appositi pulsanti per consentire: (a) Visualizzare il contenuto corrente del Vettore in una ListBox, per consentire la visione aggiornata dell'elenco; (b) Effettuare l'Ordinamento DECRESCENTE del Vettore con il Metodo Bubble-Sort (dal più grande al più piccolo); (c) Solo ad ordinamento effettuato, attivare la



possibilità di aggiungere un Nuovo Numero Intero nel vettore, inserendolo in posizione tale da mantenere ordinato il vettore stesso (Inserimento in Ordine).

119. Effettuare l'input della classifica del Campionato Piloti di Formula 1 (max 22 piloti), memorizzandola in due vettori VetPiloti e VetPunteggi. Ogni Nuovo Pilota specificato, deve essere inserito, nei vettori, in posizione tale da mantenere la classifica ordinata per Punteggio. Dopo ogni inserimento, si visualizzi la classifica in una ListBox. Al termine dell'input, si consenta di scegliere se si desidera riordinare la classifica per "Punteggio" o per "Pilota", si effettui l'ordinamento richiesto usando il Metodo di Ordinamento chiamato Bubble-Sort e si visualizzi il risultato aggiornato nella ListBox.
120. Come in una "mano" del gioco del Poker, si chiedano in input 5 carte da gioco (mazzo da 52 carte). Per esprimere la singola carta si usino due ComboBox, una per scegliere il Numero della Carta (1, 2, 3, ..., 9, 10, 11, 12, 13), l'altra per specificare il Palo della Carta (C, Q, F, P). Si memorizzino le carte in due vettori distinti VetNumCarte (numeri interi) e VetPaloCarte (caratteri) e si visualizzino costantemente le carte in una ListBox. Ogni nuova carta deve essere "inserita" nei vettori, in posizione tale da mantenere sempre le carte ordinate per Numero di Carta: si utilizzi la tecnica dell'Inserimento in Ordine. Completato l'inserimento delle 5 carte, all'utente deve essere consentito "selezionare" una carta sulla ListBox e di "cambiarla" con un'altra, specificata in input. Dopo ogni cambio si riordinino le carte con il Metodo di Ordinamento chiamato Bubble-Sort e si verifichi se si è ottenuto il punto "SCALA" (ossia 5 carte "consecutive" ... es: 5, 6, 7, 8, 9).

### Matrici

121. Usando le tecniche viste in classe, si chieda in input una tabella di NR x NC numeri reali, memorizzandoli in una matrice Mat (massimo 10x10). Completato l'input, si visualizzino i dati contenuti nella matrice Mat, utilizzando una ListBox, come visto in aula. Infine si calcoli la Somma di tutti i numeri presenti nella matrice stessa.
122. Data una matrice quadrata N x N (max 10 x 10) di Numeri Interi, effettuarne l'input e visualizzarla in una ListBox. Dato un numero intero R, verificare se è vero o no che tutti i numeri presenti nella riga R sono uguali.
123. Data una matrice quadrata Mat di N x N (massimo 10x10) numeri interi, costruire un vettore V di N elementi, ciascuno dei quali contenente la somma della corrispondente riga nella matrice. Si determini infine quale riga della matrice ha la somma dei propri numeri, più grande, rispetto a tutte le altre.

### Matrici 2

124. Data una matrice Mat di NR x NC (massimo 10x10) numeri interi, effettuarne l'input, visualizzarla in una ListBox e calcolare quanti 0 (Zero) sono presenti nella matrice stessa.
125. Data una matrice Mat di NR x NC (massimo 10x10) numeri interi, effettuarne l'input, visualizzarla in una ListBox. Per ogni riga della matrice Mat, si determini quanti 0 (Zero) essa contiene e si visualizzi, in una ListBox, il vettore di NR numeri interi così ottenuto. Infine, verificare se è vero o no, che esiste almeno una riga priva di zeri.
126. In modo simile ad un cruciverba, una matrice Mat di NR x NC (massimo 10x10) contiene lettere maiuscole: ogni elemento della matrice contiene UNA SOLA lettera, ossia un solo carattere. Effettuare l'input della matrice, visualizzarla in una ListBox e, data una parola P, determinare su quale riga o su quale colonna è eventualmente presente la parola P.

### Matrici 3

127. Data una matrice Mat di NR x NC (massimo 10x10) numeri interi, effettuarne l'input, visualizzarla in una ListBox e consentire di calcolare la somma degli elementi di una data riga o di una data colonna. Si usi una ComboBox per scegliere se sommare una riga o una colonna.
128. Date due matrici A e B di numeri reali, entrambe quadrate di dimensioni N x N (max 10 x 10), effettuarne l'input, visualizzarle in due ListBox e calcolare la matrice somma  $S = A+B$ , in cui ogni elemento di S è dato dalla somma degli elementi corrispondenti di A e di B (ossia:  $S(r,c) = A(r,c) + B(r,c)$ ). Visualizzare la matrice S così calcolata, usando una terza ListBox.
129. NR studenti (max 30) maturano a fine anno NC voti (max 10) nelle varie materie, costituendo così una matrice di NR x NC voti. Dopo aver chiesto in input i nomi degli NR studenti (vettore Studenti) e gli NRxNC voti (matrice Voti) si determini lo studente con la migliore media e l'elenco degli studenti "promossi" (ossia quelli che hanno tutte le materie con voto  $\geq 6$ ).

### DataGridView

130. Utilizzando un controllo DataGridView (inizialmente privo di righe e colonne), effettuare l'input di una matrice di Numeri Interi. Durante l'input consentire, con due appositi pulsanti, di Aggiungere una Colonna o di Rimuovere l'Ultima Colonna. Lasciare attiva la funzionalità per l'inserimento di Nuove Righe e, naturalmente, la possibilità di digitare i dati direttamente nelle celle del DataGridView. Impedire di aggiungere più di 10 Colonne. Ad input completato, calcolare la

Somma di tutti i numeri memorizzati nel DataGridView, indicando anche il Numero di Righe e il Numero di Colonne risultanti dall'input.

131. Dato un numero intero N (max 10): dimensionare un controllo DataGridView a N righe e N colonne; impostarlo in modo che non consenta input e sia privo di intestazioni; inizializzarne automaticamente tutte le celle al valore Stringa Vuota. Cliccando su una cella del DataGridView, in essa deve apparire una "X" o, se la "X" è già presente, la cella deve tornare vuota. Tramite una ComboBox, si deve poter scegliere un colore fra Bianco, Giallo, Arancio, Rosso e, con un pulsante, applicarlo alla cella selezionata come Colore di Fondo. Infine, si dia la possibilità di calcolare il numero di "X" presenti nel DataGridView.
132. Si realizzi una versione semplificata del celebre gioco "Prato Fiorito" (o "CampoMinato"): in una matrice MAT quadrata N x N (max 10) si dispongano casualmente N valori pari a 1, lasciando a 0 tutti gli altri elementi. Tramite un DataGridView (anch'esso N x N), inizialmente con le celle tutte vuote, si consenta di cliccare le varie posizioni fino a toccare una cella che nella matrice corrisponde a un elemento pari a 1 (bomba) che determina la fine della partita. Se viene cliccata una cella corrispondente a 0, si modifichi il colore di fondo della cella in Verde.

### Numeri Casuali

133. Dati in input i nomi di due squadre di calcio, simulare l'esito di una loro partita, generando casualmente il numero di gol segnati da ciascuna squadra (generare quindi 2 numeri casuali da 0 a 5). Visualizzare il risultato della partita ed evidenziare in Rosso il nome della squadra vincente. In caso di pareggio visualizzare in Verde entrambe le squadre.
134. Simulare la corsa di due "segnalini" Rosso e Verde, su un percorso lungo 30 caselle. Per visualizzare il percorso si usi un DataGridView dimensionato con 2 righe (una per segnalino) e 30 colonne. Inizialmente i segnalini si trovano sulla prima casella alla sinistra del percorso (colonna 0). Ad ogni clic su un pulsante si simuli il lancio di 2 dadi: il primo fa avanzare il segnalino Rosso e il secondo quello Verde. Evidenziare quale segnalino vince, superando per primo il traguardo della trentesima casella.
135. Consentire, tramite un pulsante, di simulare il lancio di una moneta (testa o croce). Ad ogni simulazione, aggiungere il risultato del lancio ad una ListBox ricalcolando ogni volta i rapporti "Numero di teste / Numero di lanci" e "Numero di croci / Numero di lanci" (ad esempio se i lanci sono "Testa, Croce, Croce, Testa, Testa" allora i rapporti sono  $3/5=0,6$  e  $2/5=0,4$ ).