

Il teorema di Böhm-Jacopini

Esso è un teorema di informatica teorica, enunciato nel 1966 dagli informatici italiani Corrado Böhm e Giuseppe Jacopini. Esso afferma che **“qualunque algoritmo può essere implementato in fase di programmazione** (in diagramma di flusso, pseudocodice o codice sorgente) **utilizzando tre sole strutture di controllo: sequenza, selezione e ciclo** (iterazione)

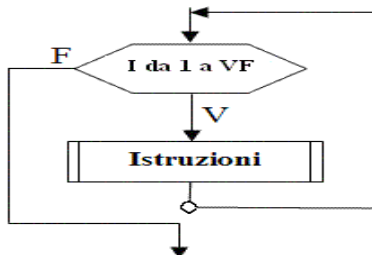
Le strutture di ripetizione o iterative o ciclo

Esse, in generale, consentono di ripetere l'esecuzione di una porzione di algoritmo più volte, finché è verificata una certa condizione. Esse possono essere di tre tipi:

- a iterazione prestabilita o ciclo a contatore
- a iterazione a controllo di testa o ciclo a condizione iniziale.
- a iterazione a controllo di coda o ciclo a condizione finale.

La struttura a iterazione prestabilita (FOR)

Essa, nel metodo del diagramma a blocchi, si rappresenta così:



Essa utilizza:

- 1) un nuovo blocco, l'esagono, che al suo interno contiene: una variabile conteggio come ad es. i, j, k
- 2) un valore finale, VF, scelto dal programmatore, che stabilisce a priori il numero massimo di iterazioni, ad es. 10, 100
- 3) blocco istruzioni che rappresenta la porzione di algoritmo da iterare, cioè il corpo dell'algoritmo
- 4) un piccolo rombo che sottintende ad ogni iterazione l'incremento della variabile conteggio ad es. $i \leftarrow i + 1$; $j \leftarrow j + 1$

La ripetizione del blocco istruzioni avviene finché la variabile conteggio assume un valore \leq al valore finale VF prestabilito, mentre si arresta quando la variabile conteggio assume un valore $>$ del valore finale VF prestabilito.

Quando si usa tale struttura a iterazione prestabilita?

Essa si usa quando nell'algoritmo ci sono una o più istruzioni che si ripetono un numero di volte prestabilito.

PROBLEMA1: Trovare l'algoritmo che letti 10 numeri interi N1,...N100, calcoli la loro somma e la stampi a video.

Prima di applicare il metodo dei 5 passi, cerchiamo di capire meglio il problema:

Ricordiamo che l'algoritmo che risolveva il problema della somma di due numeri x e y, richiedeva 3 variabili: di cui 2 per i numeri N1 e N2 e una per la variabile som.

Qui, invece, il nostro problema è di fare la somma di 100 numeri, quindi dovremmo usare 101 variabili: di cui 100 per i numeri N1, ..., N100 e una per la variabile somma SOM. Inoltre, dovremmo leggere la prima variabile N1 e poi sommarla alla variabile somma SOM, leggere la seconda variabile N2 e poi sommarla alla variabile somma SOM,.....leggere la 100° variabile

N100 e poi sommarla alla variabile somma SOM. Tale soluzione fa prevedere un algoritmo lunghissimo. Come possiamo ovviare a questo problema? ¹

Osservazione: In tale soluzione si ripetono 100 volte la coppia di istruzioni: lettura della variabile Ni e somma del valore della variabile Ni al valore della variabile somma SOM.

Allora per quanto suddetto, possiamo usare una struttura a iterazione prestabilita per la coppia di istruzioni suddette,

1) Tabelle delle variabili di input e output						
Nome	Descrizione	Tipo	Input	Output	Lavoro	
N	Numero generico da sommare	Intero	X			
SOM	Somma totale dei 10 numeri	Intero		X		
I	Indice del ciclo for	Intero				X

<p>2) Algoritmo per passi</p> <pre> graph TD Inizio([inizio]) --> SOM0[SOM ← 0] SOM0 --> LoopStart{I da 1 a 5} LoopStart -- V --> Leggi[/leggi (N)/] Leggi --> SOMAdd[SOM ← SOM+N] SOMAdd --> LoopStart LoopStart -- F --> Scrivi[/SCRIVI (SOM)/] Scrivi --> Fine([fine]) </pre>	<p>3) Programma in C++</p> <pre> #include <iostream> using namespace std; int main() { int i, n, som; cout<<"Calcolo somma di 5 numeri"<<endl; som=0; for(i=1;i<=5;i++) { cout<<i<<") Numero = ";cin>>n; som+=n; } cout<<"somma = "<<som; return 0; } </pre>
--	---

Test sul programma

```

Somma di 5 num
1> Numero = 1
2> Numero = 1
3> Numero = 1
4> Numero = 1
5> Numero = 1
somma = 5
    
```

PROBLEMA2: Trovare l’algoritmo che letti M numeri interi, calcoli la loro somma.

PROBLEMA3: Trovare l’algoritmo che permette di calcolare il maggiore tra 5 numeri interi letti in input e stamparlo in output

1) Tabelle delle variabili di input e output						
Nome	Descrizione	Tipo	Input	Output	Lavoro	
N	E’ il numero letto	Intero	X			
MAX	E’ il maggiore dei 5 numeri	Intero		X		
I	E’ l’indice del ciclo	Intero				X

¹ Nota.: per far rendere conto gli allievi che effettivamente tale soluzione è lunghissima conviene disegnare alla lavagna a sx l’algoritmo senza ciclo (fermandosi alla lettura e alla somma parziale delle prime due variabili n1 e n2, scrivendo i puntini, e poi scrivendo la lettura e la somma parziale della variabile n100) e a dx l’algoritmo con il ciclo

<pre> graph TD Start([inizio]) --> ReadN[/Leggi (N)/] ReadN --> AssignMax[MAX ← N] AssignMax --> LoopStart{ I da 2 a 5 } LoopStart -- V --> ReadN2[/Leggi (N)/] ReadN2 --> Decision{ N > MAX } Decision -- V --> AssignMax2[MAX ← N] Decision -- F --> WriteMax[/Scrivi(MAX)/] AssignMax2 --> WriteMax WriteMax --> End([fine]) </pre>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Test sul programma </div> <pre> Max di 5 numeri 1) numero 2 2) Numero 3 3) Numero -4 4) Numero -10 5) Numero 20 Max= 20 </pre>
<pre> int main() { int i, n, max; cout<<"Max di 5 numeri"<<endl; cout<<"1) numero ";cin>>n; max=n; for(i=2;i<=5;i++) { cout<<i<<" Numero ";cin>>n; if (n>max) max=n; } cout<<"Max= "<<max; return 0; } </pre>	

PROBLEMA4: Trovare l’algoritmo che letti M numeri interi, calcoli il maggiore e il minore degli M numeri e lo stampi a video.

PROBLEMA5: Trovare l’algoritmo che letti i primi 5 numeri interi negativi e positivi, calcoli e stampi a video il messaggio “è positivo” se il numero letto è positivo e il messaggio “è negativo” se il numero letto è negativo.

1) Tabelle delle variabili di input e output

Nome	Descrizione	Tipo	Input	Output	Lavoro
N	E’ il numero letto	Intero	X		
I	E’ l’indice del ciclo	Intero			X

2) Algoritmo per passi

1. inizio
2. per I che va da 1 a 5
3. leggi il I) numero, N
4. se il numero N > 0 allora
 stampa “è positivo” altrimenti
 stampa “è negativo”,
 fine se
5. incrementa I
6. fine

```

Neg / Pos di 5 numeri
1) Numero 2
2) numero positivo
2) Numero -2
-2 numero negativo
3) Numero 4
4) numero positivo
4) Numero -10
-10 numero negativo
5) Numero 1
1 numero positivo
    
```

```

int main() {
  int i, n;
  cout<<"Neg / Pos di 5 numeri"<<endl;
  for(i=1;i<=5;i++)
  {
    cout<<i<<" Numero "; cin>>n;
    if (n>0)
      cout<<n<<" numero positivo"<<endl;
    else
      cout<<n<<" numero negativo"<<endl;
  }
  return 0; }
    
```

PROBLEMA6: Trovare l’algoritmo che letti i primi M numeri interi positivi e negativi, calcoli e stampi a video il messaggio “è positivo” se il numero letto è positivo e il messaggio “è negativo” se il numero letto è negativo.

PROBLEMA7: Trovare l’algoritmo che letti 5 numeri interi, calcoli e stampi a video il messaggio “è pari” se il numero letto è pari e il messaggio “è dispari” se il numero letto è dispari. (Suggerimento: usare la funzione % per calcolare il resto)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N	E' il numero letto	Intero	X		
I	E' l'indice del ciclo	Intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. per I che va da 1 a 5 3. leggi il I) numero N 4. se $N \% 2 = 0$ allora stampa "è pari", altrimenti stampa "è dispari" fine se 5. Incrementa I 6. fine 	<pre>Pari / Dispari di 5 numeri 1) Numero 1 1 numero dispari 2) Numero 2 2 numero pari 3) Numero 4 4 numero pari 4) Numero 10 10 numero pari 5) Numero 11 11 numero dispari</pre>	<pre>int main() { int i, n; cout<<"Pari / Dispari di 5 numeri"<<endl; for(i=1;i<=5;i++) { cout<<i<<" Numero ";cin>>n; if (n % 2 == 0) cout<<n<<" numero pari"<<endl; else cout<<n<<" numero dispari"<<endl; } return 0; }</pre>
--	---	--


PROBLEMA7: Trovare l'algoritmo che letti N numeri interi con N dispari, calcoli la media tra il 1° elemento, l'ultimo elemento e l'elemento centrale e la stampi.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N	E' la dimensione dei numeri	Intero	X		
Num	E' il numero letto	Intero	X		
I	E' l'indice del ciclo	Intero			X
Som	Somma dei 3 elementi	Intero			X
Media	E' la media dei 3 elementi	Decimale		X	
Meta	Posizione dell'elemento centrale	Intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. leggi la dimensione dispari N 3. calcola la meta, $N/2+1$ 4. som=0 3. Per I che va da 1 a N 4. Leggi I) Numero, Num 5. Se $I=1$ allora Som, $Som+Num$ 6. Se $I=meta$ allora Som, $Som+Num$ 7. Se $I=N$ allora Som, $Som+Num$ 8. Incrementa I 9. $Media=Som / 3$ 10. Scrivi Media 11. Fine 	<pre>Media di 3, su N letti N= 5 meta 3 1) Numero 1 2) Numero 2 3) Numero 3 4) Numero 4 5) Numero 5 Media = 3</pre>	<pre>int main() { int i, n, som, meta, num; float media; cout<<"Media di 3, su N letti"<<endl; cout<<"N= ";cin>>n; meta=n/2+1; cout<<"meta "<<meta<<endl; som=0; for(i=1;i<=n;i++) { cout<<i<<" Numero ";cin>>num; if(i==1) som=som+num; if (i==meta) som=som+num; if (i==n) som=som+num; } media=som/3; cout<<"Media = "<<media; return 0; }</pre>
---	---	--

PROBLEMA8: Trovare l'algoritmo che letti N numeri naturali, calcoli la media tra gli elementi di indici pari e la stampi a video.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N	E' la dimensione dei numeri	Numero intero	X		
Num	E' il numero letto	Numero intero	X		
I	E' l'indice del ciclo	Numero intero			X
Som	Somma degli elementi di indici I pari	Numero Decimale			X
Media	E' la media degli elementi di indici pari	Numero Decimale		X	
ContIp	Contatore di elementi di indici pari	Numero Intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. Inizio 2. Leggi N 3. Assegna a ContIp, N / 2 4. Per I che va da 1 a N 5. Leggi Numero, Num 6. Se I%2=0 allora assegna a Som, Som+Num 9. Next I, di 1 e vai al passo 4 10. Media, Som / ContIp 11. Scrivi Media 12. Fine 		<pre>int main() { int i, n, num, contip; float media, som=0; cout<<"Media di indici pari"<<endl; cout<<"N= ";cin>>n; contip=n/2; for(i=1;i<=n;i++) { cout<<i<<" Numero ";cin>>num; if(i%2 == 0) som=som+num; } media=som/contip; cout<<"Media = "<<media; return 0; }</pre>
--	---	--

PROBLEMA 9: Trovare l’algoritmo che letti N numeri interi con N pari, calcoli la media tra il 1° elemento, l’ultimo elemento e i due elementi centrali e la stampi.

PROBLEMA 10: Trovare l’algoritmo che letti N numeri interi, calcoli la media tra i contatori degli interi pari e dispari letti e la stampi.

PROBLEMA 11: Trovare l’algoritmo che letti N interi, calcoli la media tra il max e il min e stampi.

PROBLEMA 12: Trovare l’algoritmo che letti N numeri interi, calcoli la media tra il 2) elemento e il penultimo elemento inserito e il contatore dei numeri 0 trovati e la stampi.

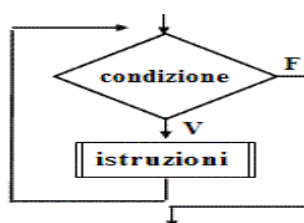
PROBLEMA 13: Trovare l’algoritmo che letti N numeri interi, calcoli la media dei quadrati degli elementi di posto dispari e la stampi.

PROBLEMA 14: Trovare l’algoritmo che letti N numeri interi, conti quanti sono: i numeri positivi, i numeri negativi, i numeri pari e dispari. Calcolare e stampare la media dei tre contatori e la stampi.

PROBLEMA 15: Trovare l’algoritmo che letti n voti reali Vi e n pesi associati Pi, calcoli la media aritmetica ponderata e la stampi. (Suggerimento: $m_p = \frac{p_1 \cdot v_1 + p_2 \cdot v_2 + \dots + p_n \cdot v_n}{p_1 + p_2 + \dots + p_n}$)

La struttura a iterazione a controllo di testa (WHILE)

Essa, nel metodo del diagramma a blocchi, si rappresenta così:



Essa permette di iterare una porzione di algoritmo finché è vera una certa condizione **stabilita non a priori**. Essa si dice a controllo di testa poiché la condizione è posta all’inizio ovvero prima del blocco di istruzioni dell’algoritmo da iterare.

Quando si usa tale struttura a iterazione a controllo di testa?

Essa si usa quando non si sa a priori quante volte si deve iterare il corpo dell’ algoritmo e la porzione di algoritmo da iterare potrebbe essere eseguita anche nessuna volta.

PROBLEMA16: Leggere una serie di numeri (num) interi positivi, finché’ l’utente non inserisce un numero negativo, che farà uscire dal ciclo (ad es. 1,2,3,4,-5). Prima di uscire dal ciclo stampare il messaggio “Lettura terminata”.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
NUM	E’ il numero da leggere.	Numero intero	X		

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. leggi il 1), NUM 3. finché NUM è > 0 esegui 4. leggi un numero 5. vai al passo 3 6. stampa “lettura terminata” 7. fine 	<pre> Leggi una serie di numeri NUM= 1 NUM= 2 NUM= 0 NUM= 0 NUM= 4 NUM= -5 Lettura terminata </pre>	<p>3) Programma in C++</p> <pre> int main() { int num; cout<<"Leggi una serie di numeri >0"<<endl; cout<<"NUM= ";cin>>num; while (num>0) { cout<<"NUM= "; cin>>num; } cout<<"Lettura terminata "; return 0; } </pre>
---	---	---

PROBLEMA17: Leggere una serie di numeri (num) interi positivi e nulli, finché l’utente non inserisca un numero negativo (ad. es.1,2,0,0,3,5,6,0,0,-1). Prima di uscire dal ciclo, il programma deve stampare quanti sono (cont0) i numeri 0 inseriti.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
NUM	E’ il numero da leggere.	Numero intero	X		
CONT0	E’ il contatore dei numeri = 0	Numero intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. leggi il 1), NUM 3. finché NUM >= 0 esegui 4. leggi un numero 5. se num==0, Cont0++ 5. vai al passo 3 6. stampa “lettura terminata” 7. fine 	<pre> Leggi una serie di numeri >=0 NUM= 1 NUM= 2 NUM= 0 NUM= 0 NUM= 3 NUM= -5 Lettura terminata Cont0 =2 </pre>	<p>3) Programma</p> <pre> #include <iostream> using namespace std; int main() { int num, cont0=0; cout<<"Leggi una serie di n.ri >=0"<<endl; cout<<"NUM= ";cin>>num; while (num>=0) { cout<<"NUM= "; cin>>num; if(num==0) cont0++; } cout<<"Lettura terminata Cont0 ="<<cont0; return 0; } </pre>
---	---	--

ESERCIZI PER CASA

PROBLEMA 19. Leggere una serie di numeri negativi e positivi (ad es. -4,-3,-2,-1,1,2,3,4,0), finchè non è inserito il numero 0 come simbolo di chiusura della serie. Contare i numeri positivi contP e negativi contN inseriti e stamparli a video

PROBLEMA 20. Leggere una serie di numeri non negativi (ad es. 1,2,0,0,2,4,5,0,0,3,-55) finchè non è inserito il numero -55 come simbolo di chiusura della serie. Sommare e contare i numeri positivi inseriti, stampando a video la media dei positivi inseriti.

IL TIPO STRINGA IN C++

Una stringa è una sequenza variabile di caratteri terminante col carattere spazio ' \0 '. Per usare una stringa in C++, è necessario includere la direttiva: **# include <string> e fare la dichiarazione seguente string: s1=""**;

La stringa è una sequenza di caratteri terminante col carattere spazio ' \0 ' ed è memorizzata in una variabile multidimensionale detta vettore fatta così

C	A	S	A	\0
---	---	---	---	----

Se s1 è una variabile stringa, essa ha lunghezza = 4. Ciò vuol dire che in memoria sono allocate, 4+1 celle: 4 celle per le singole lettere della stringa e la 5) lettera per la posizione del carattere \0 della stringa racchiusa tra parentesi quadre.

Dopo aver dichiarata in C++ una stringa così "string: s1;" essa come si può leggere?

1) Tramite il comando `cin>>s1`; Tale istruzione permette di leggere una **sola parola** di testo e selezionare **i singoli caratteri della parola scrivendo s1 [i]** (vedi algoritmo parola palindroma). **Con tale comando, non è possibile, invece, terminare l'inserimento di parole di testo con il tasto enter, poiché il comando cin non riconosce il tasto della tastiera enter** e quindi esso non si può usare per uscire dal ciclo. In tal caso per far terminare l'inserimento di una parola è necessario stabilire una parola di uscita ad es. "stop". Più precisamente con il comando `cin`, durante l'esecuzione del programma, se non si inserisce niente e si da enter, il `cin` attende ancora l'input e questo fa bloccare l'esecuzione del programma.

2) Tramite il comando `getline(cin, s1)`; Tale istruzione **permette di terminare la lettura di un testo con il tasto enter**, ma si deve leggere una **riga di testo e non solo una parola e non è possibile selezionare i singoli caratteri della riga di testo** cosa che è possibile con `cin>>s1`, ma solo calcolarne la lunghezza (Per leggere una frase di testo (ovvero più righe di testo consecutive) basterà usare il comando `getline(cin, s1)`; più volte.

3) **per fare operazioni sulle righe di testo (ad. es contare quante sono le vocali in una riga di testo), bisogna salvare la riga di testo in un array di caratteri ovvero leggere un array di caratteri**

Esempi di istruzioni con le stringhe.

- dichiarazione `string s1=""`; (crea una stringa vuota inizializzata a stringa nulla)
- se `string: s1,s2`; l'istruzione `s2 = s1`; (copia la stringa s1, nella stringa s2)

Le funzioni `s1.length()` e `getline(cin, s1)`

- l'istruzione `s1.length()` (calcola la lunghezza di una stringa s1). Se la stringa contiene al suo interno il carattere spazio, esso viene contato. Ad es. se `s1="la mia casa"`, `s1.length=11` caratteri.
- l'istruzione `getline(cin, s1)` permette di inserire una riga di testo terminata da invio

Il confronto tra stringhe

Due stringhe s1 e s2 sono uguali se hanno la stessa lunghezza e contengono gli stessi caratteri. Due stringhe si confrontano, confrontando i singoli caratteri (operazione "case-sensitive", cioè stessi caratteri, maiuscolo e minuscolo saranno diversi) e utilizzando gli operatori == e !=

Esempio siano: string s1 = "Hello"; string s2="Hello World"; string s3 = "Hi";
 Confrontando le due stringhe: **s1 != s2 → true; s2 > s1 → true; s3 > s2 → true**

PROBLEMA 20 - Leggere una serie di parole (par) finchè l'utente non inserisce la parola "stop". Calcolare la somma delle lunghezze pari delle parole inserite e stamparla. (Suggerimenti: utilizzare la funzione par.length() per determinare la lunghezza di par e la funzione % per calcolare il resto, infine ricordare che il programma termina quando la par inserita è "stop")

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
PAR	Parola da leggere.	Stringa	X		
SOMP	Somma delle lunghezze pari parole	Numero intero		X	
L	Lunghezza di una parola	Numero intero			X

2) Algoritmo per passi	3) Programma
<ol style="list-style-type: none"> 1. inizio 2. leggi parola, par 3. finché la parola, par è != "stop" esegui 4. assegna ad L, la lunghezza di par 5. se (1 % 2 == 0) allora 6. assegna a somp, somp + L 7. leggi un' altra parola, par 8. torna al passo 3 9. stampa somp 10. fine 	<pre>#include <iostream> #include <string> using namespace std; int main() { int l=0, somp=0; string par=""; cout<<"Leggi una serie di parole"<<endl; cout<<"Parola = ";cin>>par; while (par!="stop") { l=par.length(); /* calcola lunghezza par*/ if (1 % 2 == 0) somp=somp+l; cout<<"Parola = "; cin>>par; } cout<<"Somma lunghezze e' = "<<somp; return 0; }</pre>

```

Leggi una serie di parole
Parola = casa
Parola = dolce
Parola = casa
Parola = stop
La somma delle lunghezze e' = 13
    
```

PROBLEMA 21:

Leggere due stringhe (s1 e s2), verificare se sono uguali (cioè se hanno gli stessi caratteri e hanno la stessa lunghezza), altrimenti calcolare la lunghezza massima (lmax) e minima (lmin).

PROBLEMA 22 - Leggere una serie di righe di testo (riga) finchè l'utente non inserisce una riga vuota (premendo il tasto enter). Calcolare il totale dei caratteri inseriti nelle diverse righe di testo. (Suggerimenti: utilizzare la funzione riga.length() per determinare la lunghezza di ogni riga e la funzione getline(cin, riga) per inserire le varie righe)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
RIGA	righe da leggere.	Stringa	X		
SOMC	Somma dei caratteri inseriti	Numero intero		X	
I	Contatore delle righe inserite	Numero intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. Inizializza somc=0, i=1, L=0, riga="" 3 leggi 1) riga 4. finché la riga è != "" esegui 5. L=riga.length 6. somma i caratteri somc=somc+L 7. Incrementa i e leggi i) riga 8. e torna al passo 3 9. stampa somc 10. fine 	<p>3) Programma</p> <pre>#include <iostream> #include <string> using namespace std; int main() { int L=0, somc=0, i=1; string riga=""; cout<<"Serie di righe (getline e length)"<<endl; cout<<i<<" Riga = "; getline(cin, riga); while (riga!="") { L=riga.length(); somc=somc+L; i++; cout<<i<<" Riga = "; getline(cin, riga); } cout<<"Il n.ro di caratteri e' = "<<somc; return 0; }</pre>
---	--

```
Serie di righe (getline e length)
1) Riga = paolo e luca
2) Riga = amano molto il mare
3) Riga = e cosi fanno il bagnetto
4) Riga = per ore ed ore
5) Riga =
Il n.ro di caratteri e' = 69
```

PROBLEMA 23:

Leggere una serie di cognomi (Cogn) finché l'utente non inserisce la parola Fine. Contare tutti i cognomi dei tuoi compagni di classe che terminano per A, (contA) e tutti quelli che hanno il cognome che inizia per S (contS). Stampare a video i due contatori.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Cogn	Cognome da leggere	Stringa	X		
contA	Contatore che terminano per a	Numero intero		X	
contS	Contatore che iniziano per s	Numero intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. inizializza variabili l, contA, contS, cogn 3. leggi cognome cogn 4. finché il cogn!= "fine" esegui 5. assegna ad L, la lunghezza di cogn 6. se (cogn[0] == 's') allora contS++ 7. altrimenti se (cogn[l-1] == 'a') contA++ 8. leggi un' altro cognome, cogn 9. torna al passo 3 10. stampa contA e contS 11. fine 	<p>3) Programma</p> <pre>#include <iostream> #include <string> using namespace std; int main() { int l=0, contA=0, contS=0; string cogn=""; cout<<"Leggi una serie di cognomi"<<endl; cout<<"Cognome = ";cin>>cogn; while (cogn!="fine") { l=cogn.length(); /* calcola lunghezza cogn*/ if (cogn[0] == 's') contS++; else if (cogn[l-1] == 'a') contA++; /* leggi altro cognome */ cout<<"Cognome = ";cin>>cogn; } cout<<"ContA = "<<contA; cout<<" ContS = "<<contS; return 0; }</pre>
--	--

```
Leggi una serie di cognomi
Cognome = simone
Cognome = scirocco
Cognome = salzillo
Cognome = zampella
Cognome = vanella
Cognome = fine
ContA = 2 ContS = 3
```

PROBLEMA 24.

Leggere una serie di nomi di marche automobilistiche MAuto finchè l'utente non inserisca la parola "good". Determini quante marche di auto (ad es. fOrd, dOdge, ecc) hanno come secondo carattere la vocale 'o' (contO) e quante marche di auto (ad es. daciA, lanciA, ecc) terminano con la vocale 'a' (contA). Stampare a video la somma dei 2 contatori (SomOA)

PROBLEMA 25.

Leggere una serie di nomi di giocatori di calcio Ngio, finchè l'utente non inserisca "seriea", determini quanti nomi (ad es. Marko Rog, Marek, ecc.) hanno come 1 carattere la lettera 'm', (contM) e quanti (ad es. rOg, albiOl, callejOn, ecc.) ha come pen'ultimo carattere la vocale 'o'.(contO). Stampare a video il prodotto dei 2 contatori prodMO.

Una parola si dice che è palindroma se è leggibile nello stesso modo da destra a sinistra e viceversa come ad es. anna, osso, salas, ecc

PROBLEMA 26: Trovare un algoritmo che letta una parola PAR, controlli se essa sia palindroma. (Per controllare ciò basta confrontare i caratteri simmetrici della parola, finché siano uguali. Appena si trovano due caratteri simmetrici diversi, si esce dal ciclo del controllo, per stampare il messaggio "la parola non è palindroma". Serve per tal fine, la funzione par.length().

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
PAR	Parola da leggere.	Stringa	X		
I, J	Indici dei caratteri di PAR	Intero		X	
L	Lunghezza di una parola	Intero			X
BIT	Variabile binaria 0 oppure 1	Intera			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> inizio Inizializza le variabili L=0, j=0, i=0, bit=0; par=""; leggi parola, PAR; L=par.length() Finché (i<L/2 e bit= 0), esegui J=L-1-i se PAR(I) <> PAR (J) allora BIT = 1 incrementa l'indice I, di 1 e vai al passo 5 se BIT = 0 allora stampa: "la parola è palindroma" altrimenti stampa "la parola non è palindroma" fine 	<p>3)Programma</p> <pre>#include <iostream> #include <string> using namespace std; int main() { int L=0, j=0, i=0, bit=0; string par=""; cout<<"Controlla parola, se palindroma"<<endl; cout<<"Parola=";<<cin>>par; L=par.length(); while (i<L/2 && bit==0) { j=L-1-i; if (par[i] != par[j]) bit=1; /* stampa seguente non necessaria */ cout<<"par["<<i<<"]="<<par[i]<<endl; cout<<"par["<<j<<"]="<<par[j]<<endl; i++; } if(bit == 0) cout<<"La Parola e' palindroma"; else cout<<"Parola non e' palindroma"; return 0; }</pre>
--	--

<pre>Controlla parola, se palindroma Parola=osso par[0]=o par[3]=o par[1]=s par[2]=s La Parola e' palindroma</pre>	<pre>Controlla parola, se palindroma Parola=anni par[0]=a par[3]=i La Parola non e' palindroma</pre>
--	--

Definizione di cicli annidati

Si parla di **cicli annidati** (o *nested* in inglese) quando un ciclo si trova all'interno di un altro ciclo

PROBLEMA 27:

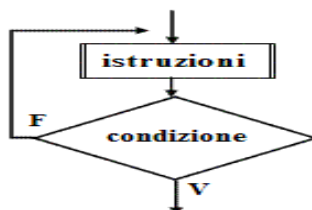
Leggere una serie di parole (PAR), finché l'utente non inserisce, parola FINE, contare quante parole palindrome sono state inserite (CPal) e stamparla a video. (Suggerimento: usare due cicli annidati)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
PAR	Parola da leggere.	Stringa	X		
I, J	Indici dei caratteri di PAR	Intero		X	
L	Lunghezza di una parola	Intero			X
BIT	Variabile binaria 0 oppure 1	Intera			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. Inizializza le variabili L=0, j=0, i=0, bit=0; par="" 3. leggi parola, PAR; 4. Finche' (par <> "stop") esegui 5. bit=0; i=0; L=par.length () 6. Finché (i<L/2 e bit= 0), esegui 7. J=L-1-i 8. se par(I) <> par(J) allora BIT = 1 9. incrementa l'indice I, di 1 e vai al passo 6 10. se BIT = 0 allora stampa: "la parola è palindroma" altrimenti stampa "la parola non è palindroma" 11. leggi altra parola par e torna al passo 4 12. fine 	<p>3)Programma</p> <pre>#include <iostream> #include <string> using namespace std; int main() { int L=0, j=0, i=0, bit=0; string par=""; cout<<"Serie di parole, palindrome? "<<endl; cout<<"Parola=";<<cin>>par; while (par !="stop") { bit=0; i=0; L=par.length(); while (i<L/2 && bit==0) { j=L-1-i; if (par[i] != par[j]) bit=1; /* stampa seguente non necessaria */ cout<<"par["<<i<<"]="<<par[i]<<endl; cout<<"par["<<j<<"]="<<par[j]<<endl; i++; } if(bit == 0) cout<<"La Parola e' palindroma"<<endl; else cout<<"Parola non e' palindroma"; <<endl; cout<<"Parola=";<<cin>>par; } return 0; }</pre>
<pre>Serie di parole, palindrome? Parola=anna par[0]=a par[3]=a par[1]=n par[2]=n La Parola e' palindroma Parola=stop</pre>	<pre>Serie di parole, palindrome? Parola=anni par[0]=a par[3]=i Parola non e' palindroma Parola=stop</pre>

La struttura a iterazione a controllo di coda (DO - LOOP - UNTIL)

Essa, nel metodo del diagramma a blocchi, si rappresenta così:



Essa permette di iterare una porzione di algoritmo finché è vera (come il ciclo while) una certa condizione stabilita non a priori. Essa **si dice a controllo di coda** poiché la condizione è posta alla fine ovvero dopo il blocco di istruzioni dell' algoritmo da iterare e per tal motivo **il corpo di istruzioni da iterare verrà eseguita almeno una volta.**

Quando si usa tale struttura a iterazione a controllo di coda?

Essa si usa quando non si sa a priori quante volte si deve iterare il corpo dell' algoritmo e la porzione di algoritmo da iterare potrebbe essere eseguita almeno una volta, ed esempio è molto utile nel controllo dei dati di ingresso di qualsiasi algoritmo.

Notare che ogni ciclo do-while può venire convertito in un ciclo while e viceversa

Ad es. si consideri il ciclo do-while: corpo do corpo while(condizione);	Esso si può riscrivere nel ciclo while: corpo while(condizione) corpo /* chiusura while */
--	--

PROBLEMA 28: Trovare un algoritmo che legga una serie di voti (da 1 a 10) di alunni sul C++ finchè l'utente non inserisca un voto <1. Contare tutti i debiti e tutte le eccellenze e stamparle.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Voto	E' il voto da leggere (da 1 a 10)	Numero intero	X		
Debiti	E' il debito, se voto <=4	Numero intero		X	
Cont10	E' il contatore delle eccellenze	Numero intero		X	

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> Inizio Inizializza variabili voto, debito, cont10 Leggi voto Fai se voto=10 allora cont10++ altrimenti se voto <=4, debito++ Leggi altro voto Finchè voto letto >0, torna al passo 4. stampa debito e cont10. fine <pre> Serie di voti <1-10> in C++ voto = 10 Voto = 10 Voto = 3 Voto = 3 Voto = 4 Voto = 1 Voto = 0 Debiti 4 Eccellenze = 2 </pre>	<p>3) Programma</p> <pre> #include <iostream> using namespace std; int main() { int voto=0, debito=0, cont10=0; cout<<"Serie di voti (1-10) in C++"<<endl; cout<<"voto = ";cin>>voto; do { if(voto==10) cont10++; else if (voto<=4) debito++; cout<<"Voto = ";cin>>voto; } while(voto>0); /* ripete finchè vera*/ cout<<"Debiti "<<debito; cout<<" Eccellenze = "<<cont10; return 0; } </pre>
--	--

ESERCIZI PER CASA:

- Ripetere gli algoritmi associati ai problemi 19 e 20 svolti con il ciclo for, stavolta con il ciclo do-while
- Ripetere gli algoritmi associati ai problemi 20-27 svolti con il ciclo while, stavolta con il ciclo do-while

PROBLEMA 29: Usando il ciclo do-while, scrivere un algoritmo che inserito un numero intero Num da tastiera stampi a video se tale numero è primo o no.

1) Tabelle delle variabili di input e output

Nome	Descrizione	Tipo	Input	Output	Lavoro
Num	E' il numero da leggere	Numero intero	X		
Div	E' il divisore del Num	Numero intero			X
R	E' il resto della divisione Num/Div.	Numero intero			X

Analisi del problema:

Per definizione un numero è primo se è divisibile solo per 1 e per se stesso. I numeri interi 2,3,5,7,11,13,17,19, ecc. sono es. di numeri primi. Infatti, ad es. 5 è primo poiché $5:1 = 5$ da resto di 0 (cioè 5 è divisibile per 1) e $5:5 = 1$ con resto di 0 (cioè 5 è divisibile per 5). Invece 5 : un qualsiasi altro numero < di 5 non da mai resto 0. Ora dando per scontato che tutti i numeri siano divisibili per 1, per verificare che il numero num scelto sia primo basta verificare che num sia divisibile solo per se stesso ovvero che il resto della divisione di num con se stesso dia come risultato 0.

Nell'algoritmo, non sapendo quale numero l'utente inserirà, dobbiamo dividere il num per i tutti i numeri che lo precedono a partire dal numero 2. Ad es. se il numero scelto è 5, dobbiamo dividere 5 con tutti i numeri da 2 sino a 5; se il numero scelto è 8 dobbiamo dividere 8 per tutti i numeri da 2 a 8; ecc. La variabile divisore div verrà inizializzata a 1, fuori dal ciclo e nel ciclo do-while incrementata subito di 1, cioè posta a 2. Nel corpo dell'algoritmo si dividerà il numero num per div e si controllerà il resto posto nella variabile r. Si uscirà dal ciclo do-while solo se div è un divisore di num (cioè solo il $num \% div = 0$). Successivamente se num è un numero primo, risulterà essere $div=num$ ovvero che la divisione $num \% div = 0$ solo nel caso $div=num$ (solo nel caso in cui il divisore è uguale al numero).

Se invece, $div < num$ (essendo sempre $num \% div = 0$) vorrà dire che il numero dato non è primo.

2) Pseudocodifica		3) Programma in C++
1. Inizio 2. Inizializzazione $div=1$. 3. Inserisci il numero num 4. Fai /* inizio ciclo do-while */ 5. incrementa di 1 div 6. calcola il resto $r=num \% div$ 7. finchè il resto $r != 0$ torna al passo 4 8. se $div=num$ allora num è primo altrimenti num non è primo, fine se 9. fine		<pre>#include <iostream> using namespace std; int main() { int div=1,r,num; cout<<"Controlla numero se primo"<<endl; cout<< "Num: ";cin >> num; do{ div++; r=num%div; }while(r!=0); if(div==num) cout<<num<<" e' primo"; else cout<<num<<" non e' primo"; return 0; }</pre>
<pre>Controlla numero se primo Num: 7 7 e' primo</pre>	<pre>Controlla numero se primo Num: 12 12 non e' primo</pre>	

PROBLEMA 30: Usando il ciclo do-while, calcola il fattoriale di un numero num e stampalo.

Analisi del problema:

Per definizione il fattoriale di $n = n*(n-1)*(n-2)*...* 3*2*1$ con $n>1$. Per $n=0$ oppure 1, il fattoriale è uguale a 1 per convenzione. Ad esempio il fattoriale di $5 = 5*4*3*2*1=1*2*3*4*5$. Per calcolare per $5!$, la strategia è: moltiplicare $1*2$ e conservarlo in una variabile prodotto P, poi moltiplicare $P*3$ e conservarlo sempre in P, poi moltiplicare $P*4$, poi $P*5$ e fine ovvero calcolare $P=1*2$; $P=P*3$; $P=P*4$; $P=P*5$. Quindi la formula è $P=P*I$ con $I=1$ a num.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Num	E' il numero di cui conoscere il fatt.	Numero intero	X		

P	E' il prodotto parziale	Numero intero			X
I	E' il fattore corrente da moltiplicare	Numero intero			X
Fat	E' il fattoriale del numero letto	Numero intero		X	

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> 1. Inizio 2. Inizializza variabili i=1, p=1 3. Inserisci numero num 4. se (num>1) allora <ol style="list-style-type: none"> 4.1 Fai /* inizio ciclo do-while */ 4.2 incrementa di 1, i 4.3 calcola il prodotto parziale p=p*i 4.4 Finchè (i<num) vera vai al 4.1 4.5. Assegna fat=p 5. altrimenti assegna fat=1 6. Stampa il fattoriale è fat 7. fine <pre style="font-family: monospace; font-size: small;"> Fattoriale di n=0,1,2,3,... Numero = 4 Fattoriale di 4 e' 24 Fattoriale di n=0,1,2,3,... Numero = 0 Fattoriale di 0 e' 1 </pre>	<p>3) Programma in C++</p> <pre style="font-family: monospace; font-size: small;"> #include<iostream> using namespace std; int main() { int i=1, p=1, num, fat; cout<<"Fattoriale numero = "; cin>>num; /* con n =0, 1 → fat = 1 per convenzione */ if (num>1) { do{ i++; p=p*i; }while(i<num); fat=p; } else fat=1; cout<<"Fattoriale di "<<num<<" e' "<<fat; return 0; } </pre>
--	--

PROBLEMA 31: In occasione di una gita scolastica a Latina, creare un gruppo whatsapp di nome LatinaDay selezionando dalla rubrica del telefono una serie di contatti, fino all'utente "zona". Stampare quanti hanno nel contatto come ultimo carattere la consonante "n" (Ad es. se leggo i contatti: BigoN, MarangoN, BuffoN, ZoniN,.....Zona, devo stampare: contN = 4)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Contatto	E' il numero di cui conoscere il fatt.	Stringa	X		
ContN	E' il prodotto parziale	Numero intero		X	
L	E' la lunghezza del contatto	Numero Intero			X

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> 1. Inizio 2. Inizializza l=0, contN=0; contatto="" ; 3. Scrivi "serie di contatti di whatsapp" 4. Inserisci contatto 5. Fai /* inizio ciclo do-while */ 6. calcola lunghezza l=contatto.lenght() 7. se contatto[l-1] = 'n' allora conN++ 8. Leggi altro contatto 9. Finchè contatto != "zona" torna al passo 5 10. Stampa contN 11. Fine <pre style="font-family: monospace; font-size: small;"> Serie di contatti di whatsapp Contatto = abbatiello Contatto = buffon Contatto = carfora Contatto = zoniin Contatto = zona ContN = 2 </pre>	<p>3) Programma</p> <pre style="font-family: monospace; font-size: small;"> #include <iostream> #include <string> using namespace std; int main() { int l=0, contN=0; string contatto=""; cout<<"Serie di contatti di whatsapp"<<endl; cout<<"Contatto = ";cin>>contatto; do { l=contatto.length(); /* lunghezza contatto*/ if (contatto[l-1] == 'n') contN++; cout<<"Contatto = ";cin>>contatto; } while (contatto!="zona"); cout<<"ContN = "<<contN; return 0; } </pre>
--	--

PROBLEMA 32: Per una indagine Statistica, leggere una serie di nomi di Supermercati, finchè l'utente non inserisce la parola "STOP". Stampare quanti hanno il nome lungo al massimo 5 caratteri (Ad es. se leggo i nomi: Todis, Sidis, Conad, Deco, Famila, ecc. devo stampare: cont5 = 3)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Marca	E' la marca del supermercato	Stringa	X		
ContL5	Contatore delle marche lunghe 5	Numero intero		X	
L	E' la lunghezza della marca	Numero Intero			X

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> Inizio Inizializza l=0, contL5=0; marca="" Scrivi "Serie di marche del supermercato" Inserisci marca del supermercato Fai /* inizio ciclo do-while */ calcola lunghezza l=marca.lenght() se l = 5 allora contL5++ Leggi altra marca del supermercato Finchè marca != "stop" torna al passo 5 Stampa contL5 Fine <pre> Serie di marche del supermercato Marca = todis Marca = sidis Marca = conad Marca = famila Marca = deco Marca = stop Marche di 5 caratteri sono: 3 </pre>	<p>3) Programma in C++</p> <pre> #include <iostream> #include <string> using namespace std; int main() { int l=0, contL5=0; string marca=""; cout<<"Serie marche del supermerc"<<endl; cout<<"Marca = ";cin>>marca; do { l=marca.length(); /* lunghezza marca */ if (l==5) contL5++; /* leggi altra marca */ cout<<"Marca = ";cin>>marca; } while (marca!="stop"); cout<<"Marche di 5 char sono: "<<contL5; return 0; } </pre>
--	---

ESEMPI DI PROBLEMI CON CONTROLLO DEI DATI DI INPUT CON DO-WHILE

PROBLEMA 33. Leggere N voti di informatica e calcolare il voto medio, vmedio. Assicurarsi che l'utente inserisca una dimensione dei voti N>0 e che voti siano compresi tra 2 e 10 inclusi voti decimali.

Analisi del Problema

Useremo un primo ciclo do-while per leggere la dimensione N dei voti e per controllare che sia >0. Poi useremo un ciclo for per leggere gli N voti e calcolare il voto medio, con al suo interno un ciclo do-while che controllerà i voti inseriti, nel senso che se l'utente dovesse inserire, nonostante il messaggio di avviso, voti non appartenenti all'intervallo limitato [2, 10], il programma dovrà richiedere il reinserimento dell'ultimo voto inserito.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N	E' la dimensione dei voti da inserire	Numero intero	X		
Voto	E' il singolo voto da inserire	Numero decimale	X		
Vmedio	E' il voto medio dei singoli voti	Numero decimale		X	
Som	E' la somma dei singoli voti	Numero decimale			X
I	E' l'indice del ciclo For	Numero intero			X

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> 1. Inizio 2. Inizializza som=0 3. Fai /* Inizio ciclo do-while */ 4. Scrivi "Inserisci la dimensione N" 5. Inserisci N 6. Finchè N<=0 torna al passo 3 7. Per I=1 a N /* inizio ciclo for */ 8. Fai /* ciclo do-while controllo voti*/ 9. Scrivi I ") voto (2-10)" 10. Inserisci voto 11. Finchè voto <2 oppure voto > 10 torna a 8. 12. Calcola som=som+voto 13. Incrementa I e torna al passo 7. 14. Calcola vmedio=som/N 15. Scrivi vmedio 16. fine. <p>Test sul Programma</p> <pre>Numero dei voti < > 0 > = -2 Numero dei voti < > 0 > = 4 1) Voto < 2..10 > = 4 2) Voto < 2..10 > = 1 2) Voto < 2..10 > = 4 3) Voto < 2..10 > = 12 3) Voto < 2..10 > = 10 4) Voto < 2..10 > = 6 Il Voto Medio e' 5.75</pre>	<p>3) Programma in C++</p> <pre>#include <iostream> #include <string> using namespace std; int main() { int i, N; float som=0, voto, vmedio; do { /* controllo sulla N inserita */ cout<<"Numero dei voti (> 0) = ";cin>>N; } while (N<=0); for (i=1;i<=N;i++) { do { /* controllo sul voto inserito */ cout<<i<<") Voto (2..10) = "; cin>>voto; } while (voto<2 voto>10); /* basta che una condizione sia vera */ som=som+voto; } vmedio=som/N; cout<<"Il Voto Medio e' "<<vmedio; return 0; }</pre>
---	---

PROBLEMA 34: Leggere 10 caratteri minuscoli (charm) dell’alfabeto inglese (26 caratteri) a cui corrispondono nel codice ASCII i numeri da 97-122. Controllare che l’utente non inserisca altri simboli oltre a quelle consentite, utilizzando la funzione int sui caratteri. Contare quanti caratteri sono nel 1 gruppo (a-m) (primi 13 caratteri) cont1g e quanti sono nel 2 gruppo (n-z) cont2g. Oltre alle variabili charm, cont1g, cont2g, usare la variabile asci=int(charm).

<p>2) Pseudocodifica</p> <ol style="list-style-type: none"> 1. Inizio 2. Inizializzo cont1g=0 e cont2g=0 3. Per I=1, to 10 /* apertura ciclo for */ 4. Fai /* ciclo do-while controllo caratteri */ 5. Inserisci carattere minuscolo, charm 6. assegna a codasci, la codifica asci di charm 7. Finchè codasci <97 o >122 torna al p.to 4 8. Se codasci >=97 e codasci <=122, allora incrementa cont1g, altrimenti cont2g++, finese 9. Incrementa I e torna al punto 3 10. Stampa cont1g e cont2g 11. Fine <pre>Inserisci 10 caratteri minuscoli 1) Carattere minuscolo <a..z (97-122)>= a 2) Carattere minuscolo <a..z (97-122)>= b 3) Carattere minuscolo <a..z (97-122)>= c 4) Carattere minuscolo <a..z (97-122)>= d 5) Carattere minuscolo <a..z (97-122)>= e 6) Carattere minuscolo <a..z (97-122)>= f 7) Carattere minuscolo <a..z (97-122)>= z 8) Carattere minuscolo <a..z (97-122)>= x 9) Carattere minuscolo <a..z (97-122)>= w 10) Carattere minuscolo <a..z (97-122)>= q Caratteri del 1) gruppo sono: 6 Caratteri del 2) gruppo sono: 4</pre>	<p>3) Programma in C++</p> <pre>#include <iostream> #include <string> using namespace std; int main() { char charm; /* carattere minuscolo */ int i, cont1g=0, cont2g=0, codasci; cout<<"10 caratteri minuscoli"<<endl; for(i=1;i<=10;i++) { do { /* controllo sul carattere inserito */ cout<<i<<") carattere =";cin>>charm; codasci=int(charm); } while (codasci<97 codasci>122); if (codasci>=97 && codasci<=109) cont1g++; else cont2g++; } cout<<cont1g; cout<<cont2g; return 0; }</pre>
--	---