

Istruzioni per il trasferimento del controllo: **break e continue**

Istruzione break:

L'istruzione **break** provoca l'uscita immediata dal ciclo (o da un'istruzione **switch**) in cui è racchiusa.

Esempio: fattoriale

```
for (F=1, i=1; ; i++)/* manca il controllo */
    if (i>N) break;
    else F=F*i;
```

Istruzione continue:

L'istruzione **continue** provoca l'inizio della successiva iterazione del ciclo in cui è racchiusa (non si applica a **switch**).

Esempio: somma dei valori pari compresi tra 1 e N

```
for (sum=0, i =1; i <= N; i++)
    if (i%2) continue;
    else sum+=i;
```

Vettori in C

```
int V[25]; /*def. di un vettore di 25 elementi interi*/
```

Indici:

- ad ogni elemento è associato univocamente un **indice**.
- è possibile riferire una singola componente specificando l'indice i corrispondente, utilizzando la notazione $V[i]$.
- L'indice deve essere di tipo enumerabile (ad esempio: `int`, `char`).
- se N è la **dimensione** del vettore (numero degli elementi), il dominio degli indici è l'intervallo

$[0, N-1]$

Esempio:

```
float A[3], i; /* A è un vettore di 3 reali*/  
A[0]=22.05; /*ass. di un valore al primo elemento */  
A[2]=17.9; /*ass. di un valore al terzo elemento */  
A[1]=0; /*ass. di un valore al secondo elemento */
```

	A
0	22.05
1	0.0
2	17.09

Gestione degli elementi di un vettore

Le singole componenti di un vettore possono essere elaborate con gli operatori del tipo ad esse associato.

Esempio: agli elementi di un vettore di interi è possibile applicare tutti gli operatori definiti per gli interi:

```
int A[10], n=7 i=2;
A[i] =n%i;          /* A[2]=1 */
A[9]=A[i++]+7;     /* A[9]=1+7=8, i=3 */
scanf("%d",&A[i]); /*ass. ad A[3] il valore letto */
```

Esempio

(somma di due vettori)

Problema:

Si realizzi un programma che, dati da standard input gli elementi di due vettori A e B, entrambi di 10 interi, calcoli e stampi gli elementi del vettore C (ancora di 10 interi), ottenuto come somma di A e B.

- Dichiariamo il nuovo tipo di dato `vettint`:

```
typedef int vettint[10];
```

- Definiamo i due vettori dati A e B, e il vettore C risultante dalla somma :

```
vettint A,B,C;
```

```

#include <stdio.h>
typedef int vettint[10];

main()
{
    vettint A, B, C ;
    int i;
    for(i=0; i<10; i++) /* lettura del vettore A */
    {
        printf("valore di A[%d] ? ", i);
        scanf("%d", &A[i]);
    }
    for(i=0; i<10; i++) /* lettura del vettore B */
    {
        printf("valore di B[%d] ? ", i);
        scanf("%d", &B[i]);
    }

    for(i=0; i<10; i++) /* calcolo del risultato */
        C[i]=A[i]+B[i];

    for(i=0; i<10; i++) /* stampa del risultato C */
        printf("C[%d]=%d\n",i, C[i]);
}

```

Esempio

(calcolo del numero medio di iscritti per classe)

Si consideri una scuola media costituita da 4 sezioni (da 'A' a 'D'), ognuna costituita da tre classi ("livello" 1, 2, 3).

Dati gli iscritti ad ogni classe di ogni sezione, si vuole calcolare il numero medio di studenti per classe relativo ad ogni sezione.

Definiamo un nuovo tipo che rappresenta il **livello**:

```
/* un elemento per ogni sezione */
```

```
typedef int livello[4];  
livello prime, seconde, terze;
```

NB: L'indice di un vettore deve essere di tipo *enumerabile* → può essere un **char**:

```
typedef char sezione;  
sezione s; // s = 'A', 'B', ...  
/*indice per accedere ai vettori delle classi */
```

s='A', s-'A' = 0

n0

s='B', s-'A' = 1

n1

s='C', s-'A' = 2

n2

s='D', s-'A' = 3

n3

indici

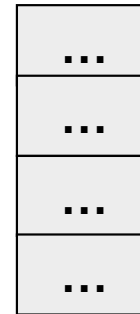
elementi

```

#include <stdio.h>
typedef int livello[4]; /*un elemento per ogni sezione */
typedef char sezione;
main()
{
    livello prime, seconde, terze;
    sezione s;
    float media=0;
    for(s='A'; s<='D'; s++)
    {
        printf("Iscritti alla prima %c: ", s);
        scanf("%d", &prime[s-'A']);
    }
    for(s='A'; s<='D'; s++)
    {
        printf("Iscritti alla seconda %c: ", s);
        scanf("%d", &seconde[s-'A']);
    }
    for(s='A'; s<='D'; s++)
    {
        printf("Iscritti alla terza %c: ", s);
        scanf("%d", &terze[s-'A']);
    }
    for(s='A'; s<='D'; s++, media=0)
    {
        media = ( prime[s-'A']+seconde[s-'A']+terze[s-'A'] )/3 ;
        printf("\nLa media degli iscritti per classe
                nella sezione %c è: %f\n", s, media );
    }
}

```

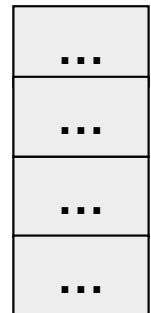
prime



seconde



terze



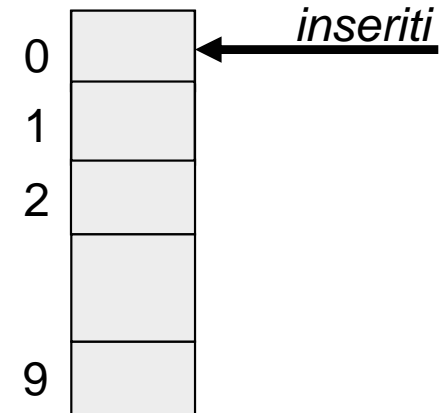
Esempio

(stampa caratteri senza ripetizioni)

Si leggano da input alcuni caratteri alfabetici maiuscoli (hp: al massimo, 10) e si riscrivano in uscita evitando di ripetere caratteri già stampati.

Soluzione:

```
while <ci sono caratteri da leggere>
    {
        <leggi carattere>;
        if <non già memorizzato>
            <memorizzalo in una struttura dati>;
    };
while <ci sono elementi della struttura dati>
    <stampa elemento>;
```



→ Occorre una variabile dove memorizzare (senza ripetizioni) gli elementi letti in ingresso: il **vettore A**

```
char A[10];
```

→ il vettore A verrà riempito con al più 10 valori: definiamo la variabile **inseriti**, che rappresenterà la “**dimensione logica**” di A (cioè, il numero di elementi significativi):

```
int inseriti=0;
```

→ la **dimensione fisica** del vettore è 10, mentre la **dimensione logica** è data dal numero effettivo di caratteri diversi tra loro

Soluzione

```
#include <stdio.h>
main()
{ char A[10], c;
  int i, j, inseriti=0, trovato;
  printf( "\n Dammi 10 caratteri: " );
  for ( i=0; ( i< 10 ); i++ )
  {   scanf("%c", &c);
      /* verifica unicità: */
      trovato=0;
      for( j=0 ; (j<inseriti)&&!trovato ; j++ )
          if (c==A[j])
              trovato=1;
      /* se c non è ancora presente in A, lo inserisco: */
      if ( !trovato )
      {   A[inseriti]=c;
          inseriti++;
      }
  }
  printf( "Inseriti %d caratteri \n", inseriti );
  for ( i=0 ; i<inseriti ; i++ )
      printf( "%c\n", A[i] );
}
```

Esempio

(lettura e stampa di matrici)

```
#include <stdio.h>
#define R 10
#define C 25

typedef float matrice[R][C];
main()
{
    matrice M;
    int i, j;
    /* lettura: */
    for(i=0; i<R; i++)
        for(j=0; j<C; j++)
            {
                printf("M[%d][%d]? ", i, j);
                scanf("%f", &M[i][j]);
            }

    /*stampa: */
    for(i=0; i<R; i++)
    {
        for(j=0; j<C; j++)
            printf("%f\t", M[i][j]);
        printf("\n");
    }
}
```

Soluzione

```
#include <stdio.h>
#define N 2
#define M 3
#define L 4
main()
{
    float A[N][L];
    float B[L][M];
    float C[N][M];
    int i, j, k;
    /* inizializzazione di A e B */
    printf("Elementi di A?\n");
    for (i=0; i<N; i++)
    {
        printf("\nRiga %d (%d elementi): ", i, L);
        for (j=0; j<L; j++)
            scanf("%f",&A[i][j]);
    }

    printf("Elementi di B?\n");
    for (i=0; i<L; i++)
    {
        printf("\nRiga %d (%d elementi): ", i, M);
        for (j=0; j<M; j++)
            scanf("%f",&B[i][j]);
    } /* continua.. */
}
```

```

/* ...prodotto matriciale: */
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
    {
        C[i][j]=0;
        for (k=0; k<L; k++)
            C[i][j]+= A[i][k]*B[k][j];
    }

/* stampa del risultato: */
for (i=0; i<N; i++)
{
    printf("\nC[%d]:\t", i);
    for (j=0; j<M; j++)
        printf("%f\t",C[i][j]);
}
}

```

```

#include <stdio.h>
#define dim 10
main()
{ int V[dim], i,j, max, tmp, quanti;
  /* lettura dei dati */
  for ( i=0; i<dim; i++ )
  { printf("valore n. %d: ",i);
    scanf("%d", &V[i]);
  }
  /*ordinamento */
  for( i=dim-1; i>1; i-- )
  { max=i;
    for( j=0; j<i; j++ )
      if ( V[j]>V[max] )
        max=j;
    if ( max!=i ) /*scambio */
    { tmp=V[i];
      V[i]=V[max];
      V[max]=tmp;
    }
  }
  /* stampa del vettore ordinato per esercizio */

```

Soluzione

```
/* ..stampa il vettore ordinato */  
  for (i=0; i<dim; i++)  
      printf("\n%d", v[i]);  
}
```

Esempio: concatenamento di stringhe

Programma che concatena due stringhe s1 e s2 date: il risultato viene inserito in s1.

```
#include <stdio.h>
/* concatenamento di due stringhe */
main()
{
    char          s1[81], s2[81];
    int  j,i;
printf("\nPrima stringa: \t");
scanf("%s",&s1);
printf("\nSeconda stringa: \t");
scanf("%s",&s2);
for ( j=0 ; s1[j]!='\0' ; j++ );
for ( i=0 ; s2[i]!='\0' && i+j<79 ; i++ )
    s1[i+j]=s2[i];
s1[i+j]='\0'; /* fine stringa */
printf("\nStringa:\t%s\n",s1);
}
```

NB: il concatenamento di due stringhe si può anche ottenere utilizzando la funzione standard di libreria `strcat()` (previa inclusione di `string.h`)