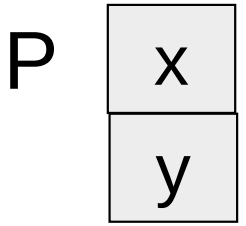
Esercizio sui record

Realizzare un programma che, lette da input le coordinate di un punto P del piano, sia in grado di applicare a P alcune trasformazioni geometriche (traslazione, e proiezioni sui due assi).

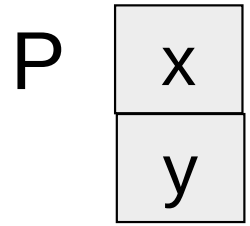
➔ Rappresentiamo il punto del piano cartesiano mediante una **`struct`** di due campi (float), ciascuno associato a una particolare coordinata:

```
typedef struct{   float x;
                  float y;
              } punto;

punto P; /* P è una variabile di tipo punto */
```



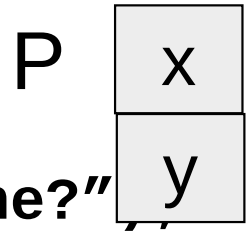
Esercizio sui record

```
#include <stdio.h>

main()
{  typedef struct{float x,y;} punto;
   punto P;
   unsigned int op;
   float  Dx, Dy;
/* lettura delle coordinate da input in P: */
   printf("ascissa? ");
   scanf("%f",&P.x);
   printf("ordinata? ");
   scanf("%f",&P.y);

/* lettura dell'operazione richiesta assumendo le
   convenzioni:
   0: termina
   1: proietta sull'asse x
   2: proietta sull'asse y
   3: trasla di Dx, Dy */
   printf("%s\n","operazione(0,1,2,3)?");
   scanf("%d",&op);
```



Esercizio sui record

```
switch (op)
  {   case 1:     P.y= 0;break;
      case 2:     P.x= 0; break;
      case 3:     printf("%s","Traslazione?”);
                  scanf("%f%f",&Dx,&Dy);
                  P.x=P.x+Dx;
                  P.y=P.y+Dy;
                  break;
      default: printf("errore!");
  }
printf( "Nuove coordinate:
         %f\t%f\n", P.x, P.y );
}
```



Esercizio

Scrivere un programma che acquisisca i dati relativi agli studenti di una classe:

- nome
- cognome
- voti: rappresenta i voti dello studente in 3 materie (italiano, matematica, inglese);

Il programma deve successivamente **calcolare** e **stampare**, per ogni studente, la **media dei voti ottenuti nelle 3 materie**.

Introduciamo un tipo di dato per rappresentare il generico studente:

```
typedef struct {    char nome[30];
                    char cognome[30];
                    int voto[3];
               } studente;
```

La classe è rappresentata da un vettore di studenti:

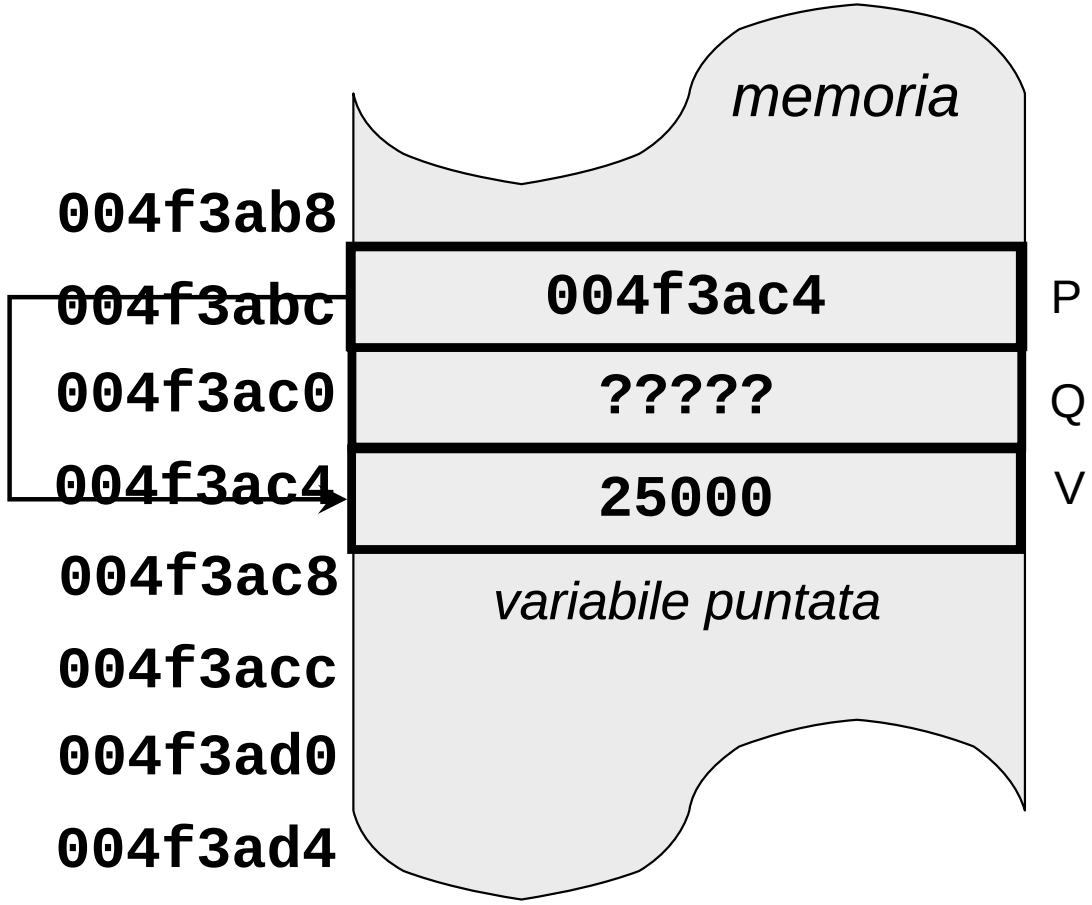
```
studente classe[20];
```

```
#include <stdio.h>
#define ita 0
#define mat 1
#define ing 2
#define N 20
typedef struct{ char nome[30],
                    cognome[30];
                int  voto[3];
      } studente;

main()
{     studente classe[N];
      float m;
      int i; int j;
      /* lettura dati */
      for(i=0;i<N; i++)
      {     fflush(stdin);
            gets(  classe[i].nome   );
            gets( classe[i].cognome );
            for(j=ita; j<=ing; j++)
                  scanf( "%d", &classe[i].voto[j] );
      }
```

```
        /* continua.. stampa delle medie */
        for( i=0; i<N; i++ ) // N studenti
        {  for( m=0,j=ita; j<=ing; j++ ) // ita=0; ing=2
               m += classe[i].voto[j]; // m accumulatore
           printf( "media di %s %s: %f\n",
               classe[i].nome, classe[i].cognome, m/3 );
        }
}
```

Esempio

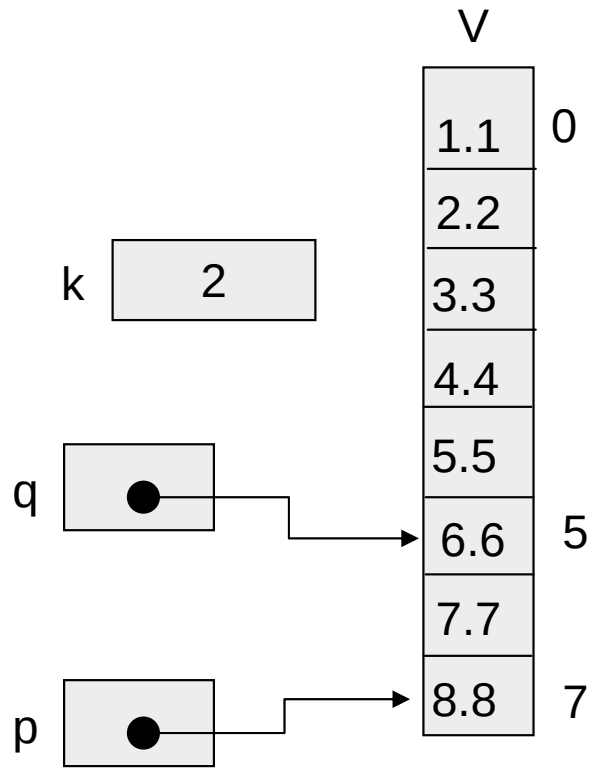
```
int *P,*Q,
    V=25000;
P=&V;
```



Operatori *aritmetici* su puntatori a vettori

Esempio:

```
#include <stdio.h>

main()
{ float V[8]={1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8};
   int k;
   float *p, *q;
   p=V+7;
   q=p-2;
   k=p-q;
   printf("%f,\t%f,\t%d\n",*p, *q, k);
}
```



➔ **Stampa: `8.8, 6.6, 2`**

Vettori e Puntatori

Durante l'esecuzione di ogni programma C, ogni riferimento ad un elemento di un vettore è tradotto in un puntatore dereferenziato; per esempio:

V[0]	**viene tradotto in**	***(V)**
V[1]	**viene tradotto in**	***(V + 1)**
V[i]	**viene tradotto in**	***(V + i)**
V[expr]	**viene tradotto in**	***(V + expr)**

Esempio:

```
#include <stdio.h>

main ()
{  char a[ ] = "0123456789"; /* a è un vettore di 10 char */
   int i = 5;
   printf("%c%c%c%c%c\n",a[i],a[5],i[a],5[a],(i-1)[a]); /* !!! */
}
```

➔ **Stampa:** **5 5 5 5 4**

NB: Per il compilatore **a[i] e i[a] sono lo stesso elemento**, perché viene sempre eseguita la conversione: **a[i] =>*(a+i)** (senza eseguire alcun controllo né su a, né su i).

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
typedef char parola[21];

main()
{  parola w, *p;
   int i, j, N;

   printf("Quante parole? ");
   scanf("%d", &N);
   fflush(stdin);
   /* allocazione del vettore */
   p=(parola *)malloc(N*sizeof(parola));
   /* lettura della sequenza */
   for(i=0; i<N; i++)
       gets(p[i]);
```

Esercizio

```
/* ..Continuua: stampa */
   for(i=N-1; i>=0; i--)
   {   j=20;
        while( p[i][j] != '\0' ) {
              j--;
       }
       for( j-- ; j>=0 ; j-- )
             printf("%c",p[i][j]);
       printf("\n");
   }
/* deallocazione del vettore*/
   free(p);
}
```