

### ESERCIZIO: Input Output

Non è possibile leggere/scrivere un intero vettore con un'unica operazione (a parte il caso di *stringhe*); occorre leggere/scrivere ognuno dei suoi componenti

```
main() {
    int i,frequenza[25];
    for (i=0; i<25; i++)
    {   scanf("%d",&frequenza[i]);
        frequenza[i]=frequenza[i]+1;
    } /* legge a terminale le componenti del
        vettore frequenza e le incrementa
        */
}
```

### ESERCIZIO: Assegnamento

Anche se due variabili vettore sono dello *stesso tipo*, NON è possibile l'assegnamento diretto:

```
int F[25], frequenza[25];
F = frequenza; /* NO */
```

ma occorre copiare componente per componente:

```
for (i=0; i<25; i++)
    F[i] = frequenza[i];
```

### ESERCIZIO: MAX e min di un vettore

```
#define N 15 /* è noto a tutti che la dimensione
             del vettore è N */
```

```
int minimo (int vet[]);
int massimo (int vet[]);
```

```
main ()
{int i, a[N];
 printf ("Scrivi %d numeri interi\n", N);
 for (i = 0; i<N; i++)
     scanf ("%d", &a[i]);
 printf ("L'insieme dei numeri è: ");
 for (i = 0; i<N; i++)
     printf("%d",a[i]);
 printf ("Il minimo vale %d e il
         massimo è %d\n", minimo(a), massimo(a));
}
```

### ESERCIZIO: MAX e min di un vettore

```
int minimo (int vet[])
{int i, min;
 min = vet[0];
 for (i = 1; i<N; i++)
     if (vet[i]<min)
         min = vet[i];
 return min;
}
```

```
int massimo (int vet[])
{int i, max;
 max = vet[0];
 for (i = 1; i<N; i++)
     if (vet[i]>max)
         max = vet[i];
 return max;
}
```

### ESERCIZIO: Ricerca di un elemento

```
#include <stdio.h>
#define N 15

int ricerca (int vet[], int el);
main ()
{int i;
 int a[N];
 printf ("Scrivi %d numeri interi\n", N);
 for (i = 0; i<N; i++)
     scanf ("%d", &a[i]);
 printf ("Valore da cercare: ");
 scanf ("%d",&i);
 if (ricerca(a,i)) printf("\nTrovato\n");
 else printf("\nNon trovato\n");
}
```

### ESERCIZIO: Ricerca di un elemento

```
int ricerca (int vet[], int el)
{int i=0;
 int T=0;
 while ((i<N) && (T==0))
     { if (el==vet[i]) T=1;
       i++;}
 return T;
}
```