

Ricerca di un elemento in vettore

Scrivere un programma in linguaggio C che riceve in ingresso una sequenza di N numeri interi. I numeri sono memorizzati in un vettore.

Il valore N è inserito dall'utente, ma il vettore può contenere al massimo 30 numeri.

Terminato l'inserimento della sequenza di numeri, l'utente inserisce un valore di riferimento. Il programma deve indicare se tale valore di riferimento è contenuto nel vettore.

```

/* PROGRAMMAZIONE IN C */

/* File: ricerca_elemento.c */
/* Soluzione proposta esercizio "Ricerca di un elemento in un vettore" */
5
#include <stdio.h>
#include <stdlib.h>

int main(void)
10 {
    const int MAXN = 30 ;    /* dimensione massima del vettore */

    int N ;                  /* occupazione effettiva del vettore */
    int vet[MAXN] ;          /* sequenza di numeri interi */
15    int i ;                 /* indice dei cicli */
    int numero ;             /* numero da ricercare nella sequenza */
    int trovato ;            /* flag per indicare se la sequenza contiene
                                il numero inserito */

20    /* LEGGI LE DIMENSIONI DEL VETTORE */
    do
    {
        printf("Quanti numeri saranno inseriti?\n") ;
        scanf("%d", &N) ;
25
        /* LA DIMENSIONE MASSIMA DEL VETTORE E' COMPRESA TRA 1 E MAXN */
        if ( N > MAXN || N <= 0 )
            printf("Errore: il numero deve essere compreso tra %d e 0\n",
                    MAXN) ;
30    }
    while ( N > MAXN || N <= 0 ) ;

    /* LEGGI UNA SEQUENZA DI N NUMERI INTERI, MEMORIZZANDOLI IN UN VETTORE */
    printf("Inserisci una sequenza di %d numeri\n", N) ;
35    for ( i=0; i<N; i++ )
    {
        printf("Elemento %d: ", i+1) ;
        scanf("%d", &vet[i]) ;
    }
}

```

```

    }
    printf("\n") ;

    /* STAMPA IL VETTORE DI INTERI */
    printf("La sequenza inserita e' la seguente\n") ;
    for ( i=0; i<N; i++ )
45     printf("Elemento %d: %d\n", i+1, vet[i]) ;
    printf("\n") ;

    /* LEGGI IL NUMERO DA RICERCARE NELLA SEQUENZA */
    printf("Inserisci il numero da cercare nella sequenza: ") ;
50     scanf("%d",&numero) ;

    /* VERIFICA SE LA SEQUENZA DI NUMERI CONTIENE IL NUMERO INSERITO */

    /* INIZIALIZZA IL FLAG "trovato". IL FLAG ASSUME I VALORI
55  -- "trovato" E' UGUALE A 0 SE IL VETTORE "vet" NON CONTIENE IL VALORE "numero"
    -- "trovato" E' UGUALE A 1 SE IL VETTORE "vet" CONTIENE IL VALORE "numero" */
    trovato = 0 ;

    /* IL CICLO FOR SCANDISCE IL VETTORE "vet" E VERIFICA SE CONTIENE
60     IL VALORE "numero".

    LA RICERCA TERMINA QUANDO SI TROVA UNA CELLA "vet[i]"
    UGUALE A "numero" O QUANDO SONO STATE CONSIDERATE TUTTE LE CELLE DEL VETTORE */

65     for ( i=0; i<N && trovato==0; i++ )
    {
        if ( vet[i] == numero )
            /* SE "vet" CONTIENE IL VALORE IN "numero", AGGIORNA IL FLAG "trovato" */
            trovato = 1 ;
70     }

    /* STAMPA IL RISULTATO */
    if ( trovato == 0 )
        printf("Il numero %d non e' contenuto nella sequenza inserita\n", numero) ;
75     else
        printf("Il numero %d e' contenuto nella sequenza inserita\n", numero) ;

    return(0)
}

```

Verificare se un vettore contiene tutti elementi tra loro uguali

Scrivere un programma in linguaggio C che riceve in ingresso una sequenza di N numeri interi. I numeri sono memorizzati in un vettore. Il valore N è inserito dall'utente, ma il vettore può contenere al massimo 30 numeri.

Terminato l'inserimento della sequenza di numeri, il programma deve verificare se gli elementi del vettore sono tutti uguali tra loro.

```
/* PROGRAMMAZIONE IN C */
```

```
/* File: tutti_uguali.c */
```

```
/* Soluzione proposta esercizio "Verificare se un vettore contiene tutti  
5 elementi tra loro uguali" */
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
10 int main(void)
```

```
{
```

```
    const int MAXN = 30 ;    /* dimensione massima del vettore */
```

```
    int N ;                  /* occupazione del vettore */
```

```
15    int vet[MAXN] ;         /* sequenza di numeri interi */
```

```
    int i ;                  /* indice dei cicli */
```

```
    int uguali ;             /* flag per indicare se la sequenza contiene numeri  
tutti uguali */
```

```
20    /* LEGGI LE DIMENSIONI DEL VETTORE */
```

```
    do
```

```
    {
```

```
        printf("Quanti numeri saranno inseriti? ") ;
```

```
        scanf("%d", &N) ;
```

```
25
```

```
        /* LA DIMENSIONE MASSIMA DEL VETTORE E' COMPRESA TRA 1 E MAXN */
```

```
        if ( N > MAXN || N <= 0 )
```

```
            printf("Errore: il numero deve essere compreso tra %d e 0\n",  
MAXN) ;
```

```
30
```

```
    }
```

```
    while ( N > MAXN || N <= 0 ) ;
```

```

/* LEGGI UNA SEQUENZA DI N NUMERI INTERI, MEMORIZZANDOLI IN UN VETTORE */
35 printf("Inserisci una sequenza di %d numeri\n", N) ;
   for ( i=0; i<N; i++ )
   {
       printf("Elemento %d: ", i+1) ;
       scanf("%d", &vet[i]) ;
40   }
   printf("\n") ;

/* STAMPA IL VETTORE DI INTERI */
printf("La sequenza inserita e' la seguente\n") ;
45 for ( i=0; i<N; i++ )
    printf("Elemento %d: %d\n", i+1, vet[i]) ;
printf("\n") ;

/* VERIFICA SE TUTTI I NUMERI DELLA SEQUENZA SONO UGUALI */

50 /* INIZIALIZZA IL FLAG "uguali". IL FLAG ASSUME I VALORI
   -- "uguali" E' UGUALE A 0 SE ALMENO DUE CELLE DEL VETTORE NON CONTENGONO
      LO STESSO VALORE
   -- "uguali" E' UGUALE A 1 SE TUTTE LE CELLE DEL VETTORE CONTENGONO
55   LO STESSO VALORE */
   uguali = 1 ;

/* IL CICLO FOR SCANDISCE IL VETTORE "vet" E VERIFICA SE TUTTE LE COPPIE DI
   CELLE ADIACENTI CONTENGONO LO STESSO VALORE. LA RICERCA TERMINA QUANDO
60 SI TROVANO ALMENO DUE CELLE ADIACENTI CHE NON CONTENGONO LO STESSO VALORE O
   QUANDO SONO STATE CONSIDERATE TUTTE LE CELLE DEL VETTORE */

```

```

        /* NEL CICLO FOR SI CONFRONTA OGNI CELLA DEL VETTORE CON LA CELLA PRECEDENTE.
        SI OSSERVA CHE LA CELLA CON INDICE 0 (VET[0]) NON PUO' ESSERE CONFRONTATA
65     CON LA CELLA PRECEDENTE (CON INDICE -1). PERTANTO L'INDICE "i" DEL CICLO
        ASSUME I VALORI TRA 1 E N-1 */
        for ( i=1; i < N && uguali==1; i++ )
        {
            if ( vet[i] != vet[i-1] )
70             /* SE LE DUE CELLE NON CONTENGONO LO STESSO VALORE, AGGIORNA IL
                FLAG "uguali" */
                uguali = 0 ;
        }

75     /* STAMPA IL RISULTATO */
    if ( uguali == 0 )
        printf("La sequenza non contiene numeri tutti uguali\n") ;
    else
        printf("La sequenza contiene numeri tutti uguali\n") ;

80     return(0)
}

```

Stampa istogrammi

Scrivere un programma in linguaggio C che riceve in ingresso una sequenza di N numeri interi. Il valore N è inserito dall'utente, ma comunque sempre inferiore a 200. I numeri sono memorizzati in un vettore. Terminato l'inserimento della sequenza di numeri, il programma deve visualizzare una riga di asterischi per ogni numero inserito. Il numero di asterischi nella riga è pari al valore del numero inserito.

Ad esempio, dato il vettore 9 4 6 il programma deve visualizzare:

Elemento 1: 9 *****

Elemento 2: 4 ****

Elemento 3: 6 *****


```

/* PROGRAMMAZIONE IN C */

/* File: istogrammi.c */
/* Soluzione proposta esercizio "Stampa istogrammi" */
5
#include <stdio.h>
#include <stdlib.h>

int main(void)
10 {
    const int MAXN = 200 ; /* dimensione massima del vettore */

    int N ; /* occupazione del vettore */
    int vet[MAXN] ; /* sequenza di numeri interi */
15 int i, j ; /* indici dei cicli */

    /* LEGGI LE DIMENSIONI DEL VETTORE */
    do
    {
20 printf("Quanti numeri saranno inseriti? ") ;
    scanf("%d", &N) ;

    /* LA DIMENSIONE MASSIMA DEL VETTORE E' COMPRESA TRA 1 E MAXN */
    if ( N > MAXN || N <= 0 )
25 printf("Errore: il numero deve essere compreso tra %d e 0\n",
        MAXN) ;
    }
    while ( N > MAXN || N <= 0 ) ;

30 /* LEGGI UNA SEQUENZA DI N NUMERI INTERI, MEMORIZZANDOLI IN UN VETTORE */
    printf("Inserisci una sequenza di %d numeri\n", N) ;
    for ( i=0; i<N; i++ )
    {

```

```

        printf("Elemento_%d:_", i+1) ;
35     scanf("%d", &vet[i]) ;
    }
    printf("\n") ;

    /* STAMPA IL VETTORE DI INTERI */
40     printf("La sequenza inserita e' la seguente\n") ;
    for ( i=0; i<N; i++ )
        printf("Elemento_%d:_%d\n", i+1, vet[i]) ;
    printf("\n") ;

45     /* STAMPA GLI ISTOGRAMMI */
    printf("Stampa degli istogrammi\n") ;
    for ( i=0; i<N; i++ )
    {
        /* STAMPA IL NUMERO IN POSIZIONE "i" NEL VETTORE "vet" (OSSIA vet[i]) */
50     printf("Elemento_%d:_%d_", i+1, vet[i]) ;

        /* STAMPA L'ISTOGRAMMA PER IL NUMERO "vet[i]", OSSIA STAMPA UN
        NUMERO DI "*" UGUALE A vet[i] */
        for ( j=0; j < vet[i]; j++ )
55     printf("*") ;
        printf("\n") ;
    }

    return(0)
}

```

Operazione di shift di un vettore

Scrivere un programma in linguaggio C che riceve in ingresso una sequenza di N numeri interi. Il valore N è inserito dall'utente. I numeri sono memorizzati in un vettore. Il programma esegue le seguenti operazioni:

1. visualizza il vettore
2. esegue uno spostamento (shift) a sinistra di una posizione del contenuto del vettore. Pertanto ogni elemento del vettore deve assumere il valore dell'elemento immediatamente successivo all'interno del vettore. L'elemento di indice N-1 deve assumere il valore zero.

Ad esempio dato il vettore: 1 10 15 18

Il programma deve generare il vettore: 10 15 18 0

Il programma visualizza il vettore ottenuto.

3. esegue uno spostamento (shift) a destra di una posizione del contenuto del vettore ottenuto nel passo precedente. Pertanto ogni elemento del vettore deve assumere il valore dell'elemento immediatamente precedente all'interno del vettore. L'elemento di indice 0 deve assumere il valore zero.

Ad esempio dato il vettore: 10 15 18 0

Il programma deve generare il vettore: 0 10 15 18

Il programma visualizza il vettore ottenuto.

Nota. Nella definizione di “destra” e “sinistra” si immagini il vettore stampato orizzontalmente, a partire dalla cella di indice 0.

```

/* PROGRAMMAZIONE IN C */

/* File: shift_vettore.c */
/* Soluzione proposta esercizio "Operazione di shift di un vettore" */
5
#include <stdio.h>
#include <stdlib.h>

int main(void)
10 {
    const int MAXN = 200 ; /* dimensione massima del vettore */

    int N ;                /* dimensione del vettore */
    int vet[MAXN] ;        /* sequenza di numeri interi */
15 int i ;                /* indice dei cicli */

    /* LEGGI LE DIMENSIONI DEL VETTORE */
    do
    {
20         printf("Quanti numeri saranno inseriti? ") ;
        scanf("%d", &N) ;

        /* LA DIMENSIONE MASSIMA DEL VETTORE E' COMPRESA TRA 1 E MAXN */
        if ( N > MAXN || N <= 0 )
25             printf("Errore: il numero deve essere compreso tra %d e 0\n",
                    MAXN) ;
    }

```

```

30  /* LEGGI UNA SEQUENZA DI N NUMERI INTERI, MEMORIZZANDOLI IN UN VETTORE */
printf("Inserisci una sequenza di %d numeri\n", N) ;
for ( i=0; i<N; i++ )
{
    printf("Elemento %d: ", i+1) ;
35    scanf("%d", &vet[i]) ;
}
printf("\n") ;

/* STAMPA IL VETTORE DI INTERI */
40 printf("La sequenza inserita e' la seguente\n") ;
for ( i=0; i<N; i++ )
    printf("Elemento %d: %d\n", i+1, vet[i]) ;
printf("\n") ;

45 /* ESEGUI UNO SPOSTAMENTO (SHIFT) A SINISTRA DI UNA POSIZIONE DEL CONTENUTO
DEL VETTORE. ASSEGNA IL VALORE 0 ALLA CELLA vet[N-1] */
for ( i=0; i<N-1; i++ )
    /* COPIA NELLA CELLA vet[i] IL CONTENUTO DELLA CELLA SUCCESSIVA vet[i+1] */
    vet[i] = vet[i+1] ;

50 /* ASSEGNA IL VALORE 0 ALLA CELLA vet[N-1]. NOTA: QUESTA ASSEGNAZIONE DEVE
ESSERE FATTA AL TERMINE DEL CICLO FOR. INFATTI SE VIENE FATTA PRIMA DEL CICLO
FOR SI PERDEREBBE IL VALORE INIZIALMENTE CONTENUTO NELLA CELLA vet[N-1].
QUESTO VALORE DEVE INVECE ESSERE ASSEGNATO ALLA CELLA vet[N-2] */
55 vet[N-1] = 0 ;

/* STAMPA IL VETTORE DI INTERI */
printf("Stampa del vettore dopo l'operazione di shift a sinistra\n");
for ( i=0; i<N; i++ )
60    printf("Elemento %d: %d\n", i+1, vet[i]) ;
printf("\n") ;

```

```
/* ESEGUI UNO SPOSTAMENTO (SHIFT) A DESTRA DI UNA POSIZIONE DEL CONTENUTO  
DEL VETTORE. ASSEGNA IL VALORE 0 ALLA CELLA vet[0] */
```

```
65 for ( i=N-1; i>0; i-- )  
    /* COPIA NELLA CELLA vet[i] IL CONTENUTO DELLA CELLA PRECEDENTE vet[i-1] */  
    vet[i] = vet[i-1] ;
```

```
/* ASSEGNA IL VALORE 0 ALLA CELLA vet[0]. NOTA: QUESTA ASSEGNAZIONE DEVE  
70 ESSERE FATTA AL TERMINE DEL CICLO FOR. INFATTI SE VENISSE FATTA PRIMA DEL  
CICLO FOR SI PERDE IL VALORE INIZIALMENTE CONTENUTO NELLA CELLA vet[0].  
QUESTO VALORE DEVE INVECE ESSERE ASSEGNATO ALLA CELLA vet[1] */  
vet[0] = 0 ;
```

```
75 /* STAMPA IL VETTORE DI INTERI */  
printf("Stampa del vettore dopo l'operazione di shift a destra\n");  
for ( i=0; i<N; i++ )  
    printf("Elemento %d: %d\n", i+1, vet[i]) ;  
printf("\n") ;
```

```
80 return(0)
```

```
}
```