

Algoritmi notevoli

Inserimento di n elementi

Prof. Giuseppe Ascia

```
#include <stdio.h>

main()
{   int V[10],n, i;

    /* lettura del numero di elementi da leggere*/
    do {
        printf("Quanti elementi vuoi leggere ? (<=10)");
        scanf("%d",&n);
    } while (n <0 || n > 10);

    /* lettura degli n elementi */
    for(i=0;i<n;i++)
        {printf("Numero: ");
          scanf("%d",&V[i]);
        }

    /* visualizzazione degli n elementi*/
    for(i=0;i<n;i++)
        printf("V[%d]=%d\n",i,V[i]);

    scanf("%d",&i);
}
```

Inserimento di un elemento o visualizzazione in un vettore

Prof. Giuseppe Ascia

```
#include <stdio.h>

main()
{
    int A[10],n,x,operazione,i;

    /*inizializzazione */
    n=0;

    do
    {
        printf("Operazione:      0) inserimento 1) visualizzazione\n");
        scanf("%d",&operazione);
        if(operazione==0 && n<10)
            { /*Inserimento*/
                printf("Valore: ");
                scanf("%d",&x);
                A[n]=x;
                n++;
            }
        else if (operazione==1)
            /* Visualizzazione*/
            for(i=0;i<n;i++)
                printf("%d ",A[i]);
    } while (operazione==0 || operazione==1);
}
```

Ricerca sequenziale in un vettore (1)

Prof. Giuseppe Ascia

```
#include <stdio.h>
main()
{ double A[10],x;
  int n,i,trovato=0;
  /* Lettura del numero n di valori da leggere */
  do {
    printf("Quanti valori vuoi leggere ? (<=10)\n");
    scanf("%d",&n);
  } while (n<1 || n>10);

  /* Lettura degli n elementi */
  for(i=0;i<n;i++)
    {printf("Inserisci un nuovo numero\n");
     scanf("%lf",&A[i]);
    }

  /* Lettura dell'elemento da trovare */
  printf("Inserire il valore da trovare");
  scanf("%lf",&x);
```

Ricerca sequenziale in un vettore (2)

Prof. Giuseppe Ascia

```
/* Ricerca dell'elemento */
for(i=0;i<n;i++)
    if (A[i]==x)
        break;

if(i<n)
    printf("L'elemento e' in posizione %d",i);
else
    printf("L'elemento non e' presente");
}
```

Inserimento di un elemento o ricerca in un vettore (1)

Prof. Giuseppe Ascia

```
#include <stdio.h>

main()
{
    int A[10],n,x,operazione,i;

    /*inizializzazione */
    n=0;

    do
    { printf("Operazione:      0) inserimento 1) ricerca\n");
      scanf("%d",&operazione);
      if(operazione==0 && n<10)
        { /*Inserimento*/
          printf("Valore: ");
          scanf("%d",&x);
          A[n]=x;
          n++;
        }
    }
```

Inserimento di un elemento o ricerca in un vettore (2)

Prof. Giuseppe Ascia

```
else if (operazione==1)
    /* Ricerca*/
    {printf("Elemento da cercare: ");
    scanf("%d",&x);
    for(i=0;i<n;i++)
        if(A[i]==x) /* Trovato*/
            break;
    if(i<n) /* trovato*/
        printf("Trovato in posizione %d\n",i);
    else printf("Non trovato\n");
    }

    } while (operazione==0 || operazione==1);

}
```

Leggere n valori di un vettore di 10 elementi ed eliminare l'elemento in posizione p (1)

Prof. Giuseppe Ascia

```
#include <stdio.h>
main()
{ int A[10],i,n,p;

/* Lettura del numero n di valori da leggere */ esempio
do {
    printf("Quanti valori vuoi leggere ? (<=10)\n");
    scanf("%d",&n);
} while (n<1 || n>10);

/* Lettura degli n elementi */
for(i=0;i<n;i++)
    {printf("Inserisci un nuovo numero\n");
    scanf("%d",&A[i]);
    }
```

7	24	52	63	25	2				
0	1	2	3	4	5	6	7	8	9

Leggere n valori di un vettore di 10 elementi ed eliminare l'elemento in posizione p (2)

Prof. Giuseppe Ascia

```
/* Lettura della posizione p dell'elemento da eliminare */
do {
    printf("Posizione dell'elemento da eliminare\n");
    scanf("%d",&p);
} while (p<0 || p>n-1);

/* Eliminazione dell'elemento in posizione p.
E' ottenuta copiando l'elemento A[i+1] in A[i] con
p<= i < n-1
e decrementando n di 1 */

for(i=p;i<n-1;i++)
    A[i]=A[i+1];

n--;

/* Visualizzazione vettore */
for(i=0;i<n;i++)
    printf("%d\n",A[i]);
}
```

Leggere n valori di un vettore di 10 elementi ed eliminarne uno (1)

Prof. Giuseppe Ascia

```
#include <stdio.h>
main()
{ int A[10],i,n,p, x;

/* Lettura del numero n di valori da leggere */
do {
    printf("Quanti valori vuoi leggere ? (<=10)\n");
    scanf("%d",&n);
} while (n<1 || n>10);

/* Lettura degli n elementi */
    for(i=0;i<n;i++)
        {printf("Inserisci un nuovo numero\n");
          scanf("%d",&A[i]);
        }
}
```

Leggere n valori di un vettore di 10 elementi ed eliminarne uno (2)

Prof. Giuseppe Ascia

```
/* Lettura dell'elemento da eliminare */
printf("Elemento da eliminare\n");
scanf("%d", &x);
/* Ricerca dell'elemento */
for(p=0;p<n;p++)
    if(A[p]==x)
        break;
/* Eliminazione se trovato*/
if(p<n) /* trovato*/
{ for(i=p;i<n-1;i++)
    A[i]=A[i+1];
  n--;
}
else printf("Elemento non trovato");

/* Visualizzazione vettore */
for(i=0;i<n;i++)
    printf("%d\n",A[i]);
}
```

Ordinamento mediante bubble sort

Prof. Giuseppe Ascia

```
#include <stdio.h>

main()
{ int A[10], ind, sup, aux, n;
  printf("Quanti elementi? (<=10)");
  scanf("%d", &n);
  /* Lettura degli n elementi del vettore */
  for(ind=0; ind<n; ind++)
  { printf("Inserire un numero : ");
    scanf("%d", &A[ind]);
  }
  /*Ordinamento */
  for(sup=n-1; sup>0; sup--)
    for(ind=0; ind<sup; ind++)
      if (A[ind+1]<A[ind])
        { aux=A[ind];
          A[ind]=A[ind+1];
          A[ind+1]=aux;
        }
  for(ind=0; ind<n; ind++)
    printf("%d\n", A[ind]);
}
```

Ricerca binaria per un vettore ordinato (1)

Prof. Giuseppe Ascia

```
#include <stdio.h>
main()
{ int A[10],x;
  int inf,sup,med, n,i,ultimo,trovato=0;
/* Lettura del numero n di valori da leggere */
do {
  printf("Quanti valori vuoi leggere ? (<=10)\n");
  scanf("%d",&n);
} while (n<1 || n>10);

/* Lettura degli n elementi.
  Si suppone che i valori siano inseriti in modo crescente */

for(i=0;i<n;i++)
  {printf("Inserisci un nuovo numero\n");
   scanf("%d",&A[i]);
  }

/* Lettura dell'elemento da trovare */
printf("Inserire il valore da trovare");
scanf("%d",&x);
```

Ricerca binaria per un vettore ordinato (2)

Prof. Giuseppe Ascia

```
/* Ricerca binaria di x.
```

```
   Se l'elemento x e' in posizione A[med], la ricerca ha termine poiche' e'
   stato trovato. Altrimenti se A[med]<x, allora x, se esiste, si trova nella
   seconda meta' del vettore e pertanto viene aggiornato inf=med+1. Se anche questa
   condizione e' falsa, allora x, se esiste, si trova nella prima meta' del vettore.
   E viene aggiornato sup=med-1. Se l'elemento non esiste, aggiornando di volta in
   volta inf o sup, si arrivera' alla situazione in cui inf>sup. Pertanto, ha senso
   cercare x solo se inf <= sup */
```

```
inf=0; sup=n-1;
```

```
while(inf<=sup && !trovato)
```

```
{ med=(inf+sup)/2;
```

```
  if(A[med]==x)
```

```
    trovato=1;
```

```
  else if (A[med]<x)
```

```
    inf=med+1;
```

```
  else sup=med-1;
```

```
}
```

```
if (trovato) printf("%d e' in pos. %d", x,med);
```

```
else printf("%d non e' presente",x);
```

```
}
```