

Procedura inversione

- Scopo

Invertire la posizione degli elementi di un Array

- Specifiche

void inversione (int Array[], int N)

- Descrizione

a) Background del problema

L' inversione di un Array consiste nel invertire i valori contenuti nelle celle con la speculare.

Indicando con $\text{Array}=(a_0, \dots, a_{N-1})$ l' Array

b) Descrizione del algoritmo

L' algoritmo adoperato copia il contenuto delle celle da una parte all'altra del Array sfruttando due indici che si muovono in maniera speculare, ne consegue il seguente codice in Pascal-LIKE

```
for(i=0, j=N-1; i < N/2 ; i++, j--)
{
    App=Array[i];
    Array[i]=Array[j];
    Array[j]=App;
}
```

- Riferimenti bibliografici

A. Murli, G. Laccetti, et al., Laboratorio di Programmazione I Liguori 2003

- Lista dei parametri

int Array[]	:	Array. In Output invertito rispetto al Input
int N	:	Lunghezza del Array. Ricevuta in Input non va ad essere modificata
int i	:	Indice.
int j	:	Cella speculare a quella puntata.
int App	:	Variabile interna d' Appoggio.

- Indcatore d' errore

Nessuno

- Procedure ausiliarie

Nessuno

- Raccomandazioni sull'uso

Nessuno

- Complessità Computazionale

a) Complessità di tempo

b) Complessità di spazio

- Esempio d'uso

- Esempio di programma chiamante

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

```
//prototipo di funzione
```

```
void inversione (int Array[],int N);
```

```
main ()
```

```
{
```

```
    //Dichiarazione
```

```
    int      *Array ;
```

```
    int      i,
```

```
           N;
```

```
    //Inizializzazione del array da invertire
```

```
    printf ("Inserisci il numero di elementi del array: ");
```

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

```
    //alloco dinamicamente la memoria necessaria
```

```
    Array = (int *) malloc (N*sizeof(int ));
```

```
    //Riempo l'Array
```

```
    for (i=0; i<N; i++){
```

```
        printf("\nInserisci il valore della cella[%d]: ", i);
```

```
        scanf("%d",&Array[i]);
```

```
    }
```

```
    //Chiamata delle function
```

```
    inversione ( Array, N);
```

```
    //Stampo del risultato
```

```
    printf("\n L`Array e` stato invertito:");
```

```
    for (i=0; i<N; i++){
```

```
        printf("\nArray[%d]:\t%d", i, Array[i]);
```

```
    }
```

```
}
```

- Esempio di esecuzione

Inserisci il numero di elementi del array: 5

Inserisci il valore della cella[0]: 1

Inserisci il valore della cella[1]: 2

Inserisci il valore della cella[2]: 3

Inserisci il valore della cella[3]: 4

Inserisci il valore della cella[4]: 5

L`Array e` stato invertito:

Array[0]: 5

Array[1]: 4

Array[2]: 3

Array[3]: 2

Array[4]: 1