

Esercizi su vettori, puntatori, struct

Esempi di errori nell'uso di puntatori:

```
const int N=4;
```

`int a[N] = {0, 2, 2, 3, 4};` Il vettore contiene N=4 elementi ma viene inizializzato con 5. Errore: "too many initializer values"

`int b[N-5];` errore se N fosse uguale a 4, perché N-5 sarebbe uguale a -1 e non posso indicare come dimensione di un vettore un valore negativo.

`int c[3.0];` La dimensione di un vettore deve essere un intero (positivo) mentre qui viene utilizzata una costante double.

```
int *x, y;
```

```
y=5;
```

`x=y;` l'istruzione corretta sarebbe `x=&y`

```
int a[N] = {10,20,30};
```

`a++;` errore!! `a` è un puntatore costante e non può essere incrementato

corretto sarebbe stato:

```
int *aPtr;
```

```
aPtr=a;
```

```
aPtr++;
```

```
float x = 19.34;
```

`float xPtr = &x;` //errore `xPtr` è una variabile float mentre dovrebbe essere un puntatore a float

`cout << xPtr;` // errore

corretto sarebbe stato:

```
float x = 19.34;
```

```
float *xPtr = &x;
```

```
cout << *xPtr << endl;
```

Esercizio 1:

Scrivere un programma che esegua una ricerca dicotomica (o binaria) di un elemento in un array di numeri interi ordinato in modo crescente. L'algoritmo di ricerca dicotomica è iterativo e confronta, ad ogni iterazione, l'elemento da cercare con l'elemento centrale nell'array. Se il confronto fallisce si ripete la ricerca su una delle due metà dell'array.

Esercizio 2:

Scrivere un programma che legga N valori float, li memorizzi in un array e stampi a video il valore massimo, minimo, il valore medio e la deviazione standard.

Esercizio 3:

In un esercizio di telepatia, un sensitivo scommette di essere in grado di indovinare almeno 3 numeri consecutivi, in una sequenza di N numeri interi pensati da uno spettatore. Si scriva un programma che crei un array di dimensione N, con allocazione dinamica, in cui il numero N e la sequenza di numeri pensati dallo spettatore vengono letti da tastiera, e verifichi se il sensitivo dice la verità.

Il programma acquisisce dal sensitivo i 3 numeri (con input da tastiera).

Il programma verifica, se esiste, nella sequenza di N numeri contenuti nell'array una sotto-sequenza di 3 numeri esattamente uguale a quella inserita dal sensitivo.

Esercizio 4:

Scrivere un programma che, data una stringa di N caratteri, conti il numero di vocali nella stringa e scriva tale valore sullo schermo. Si supponga che la stringa contenga soltanto caratteri minuscoli.

Il programma deve inoltre stampare il massimo carattere e la sua posizione all'interno della stringa.

Stampare a video anche la stringa invertita. Costruire una nuova frase in cui il primo carattere di ciascuna parola nella frase di partenza è stato reso maiuscolo. Tutti gli altri caratteri devono essere resi minuscoli.

Esercizio 5:

Si scriva un programma in cui data una matrice bidimensionale di valori (interi o float o double) individui la colonna la cui somma degli elementi ha valore maggiore.

Esercizio 6:

Scrivere un programma per contare il numero coppie di punti (x,y) a distanza inferiore ad una certa soglia. I punti vengono salvati in un array di strutture dati. Le coordinate dei punti sono generate casualmente.

Esercizio 7:

Crivello di Eratostene. Un numero primo è un numero intero divisibile soltanto per 1 e per se stesso. Il crivello di Eratostene è un metodo per trovare i numeri primi. Funziona così:

1. Creare un array con tutti gli elementi inizializzati a 1. Gli elementi dell'array che hanno per indici dei numeri primi resteranno uguali a 1. Tutti gli altri invece verranno impostati a zero.
2. Partite dall'indice 2 (l'indice 1 è necessariamente primo), e ogni volta che trovate un elemento dell'array uguale a 1, impostate a 0 tutti gli elementi il cui indice è multiplo dell'elemento trovato. Quando avremo completato il procedimento, gli elementi dell'array che valgono ancora 1 avranno per indice un numero primo. Scrivere un programma per realizzare l'algoritmo su un array di 10000 elementi, per determinare i numeri primi compresi tra 2 e 9999.