

Es. 6.1 – Massimo in vettore

Scrivere una funzione `double massimo(...)` che dato un vettore di `double` in ingresso, restituisce il valore massimo presente nel vettore.

Scrivere un programma che dato il vettore di `double` {1.3, 4.5, 2.4, 8.4, -3.14, -3.14} ne calcola il massimo con la funzione `massimo(...)` e lo stampa a video.

Output di esempio:

```
Il valore massimo presente nel vettore e': 8.4
```

Es. 6.2 – Posizione minimo in vettore

Scrivere una funzione `int posMinimo(...)` che dato un vettore di `double` in ingresso, restituisce la posizione dell'elemento minimo. Se ci sono più minimi uguali, restituisce la posizione del primo.

Scrivere un programma che dato il vettore di `double` {1.3, 4.5, 2.4, 8.4, -3.14, -3.14} ne calcoli la posizione del minimo con la funzione `posMinimo(...)` e la stampi a video.

Output di esempio:

```
La posizione del primo minimo all'interno del vettore e': 4
```

Es. 6.3 – Somma pari di vettore

Scrivere una funzione `bool pari(...)` che prende in ingresso un vettore di interi e restituisce vero se la somma degli elementi del vettore è pari, falso altrimenti.

Scrivere un programma che crea un vettore di 10 interi, chiede all'utente di digitare da tastiera 10 interi e li inserisce nelle 10 locazioni del vettore; successivamente invoca la funzione e scrive a video "PARI" se la somma degli elementi dell'array è pari, "DISPARI" altrimenti.

Output di esempio:

```
Inserisci 10 elementi dell'array:  
1  
2  
3  
3  
3  
4  
8  
9  
12  
11  
  
La somma degli interi inseriti e' PARI.
```

Es. 6.4 – Concatena vettori

Scrivere una funzione void `concatena(const int* v1, const int* v2, int n1, int n2, int* v3)` che prende in ingresso i vettori `v1` e `v2` di dimensioni rispettivamente `n1` ed `n2`, e restituisce mediante l'argomento `v3` il vettore risultante dalla loro concatenazione. Si assuma che il vettore `v3` sia stato correttamente allocato della dimensione opportuna dal chiamante.

Scrivere un programma che chiama la funzione `concatena(...)` sui vettori `{2, 6}` e `{3, 15, 4}` e stampi il vettore risultante.

Output di esempio:

```
v3: 2 6 3 15 4
```

Es. 6.5 – Ruota vettore

Scrivere una funzione void ruotaVettore(...) che trasla a destra di una posizione ciascun elemento di un array di interi in modo circolare, cioè l'elemento che esce rientra sulla sinistra.

Scrivere un main che:

- a) dichiara una costante N=6
- b) richiede l'inserimento da tastiera di N interi da tastiera;
- c) richiede l'inserimento da tastiera di un numero di passi di traslazione k;
- d) ruota il vettore di k passi, sfruttando la funzione ruotaVettore(...);
- e) stampa a video il contenuto del vettore.

Output di esempio:

```
Inserire gli elementi del vettore:  
3 4 5 -1 3 0  
Inserire un numero di passi di rotazione:  
2  
Vettore risultante:  
3 0 3 4 5 -1
```

Es. 6.6 – Stringa palindroma

Scrivere la funzione `bool palindroma(...)` che restituisce `true` se la C-stringa passata come parametro è palindroma (cioè se è identica sia che venga letta da sinistra a destra che da destra a sinistra), `false` altrimenti. Esempi di C-stringhe palindrome sono "ingegni", "otto", "radar", "itopinonavevanonipoti". Si consideri le lettere maiuscole diverse dalle minuscole, quindi per esempio 'a' è diversa da 'A'.

Scrivere un programma che:

- legge da tastiera una parola di al più 40 caratteri,
- richiama la funzione `palindroma(...)` su di essa,
- stampa a video se la parola è palindroma oppure no.

Output di esempio:

```
Inserisci una parola:  
radar  
La parola e' palindroma
```

```
Inserisci una parola:  
Anna  
La parola NON e' palindroma
```

Es. 6.7 – Trova carattere in C-stringa

Scrivere una funzione `my_strchr(...)`, che riceve in ingresso una C-stringa e un carattere, e che cerca la prima occorrenza del carattere all'interno della stringa. La funzione deve restituire un puntatore al carattere trovato all'interno della C-stringa, oppure `nullptr` se il carattere non si trova nella C-stringa.

Esempio: Nel caso in cui la C-stringa e il carattere abbiano i seguenti valori:

```
str = "123abc321"
```

```
chr = '3'
```

la funzione deve restituire un puntatore che punta al terzo carattere della C-stringa di input. Per realizzare la funzione `my_strchr(...)` non è possibile (ovviamente) utilizzare la stessa `strchr(...)`.

Scrivere un programma che definisce una C-stringa pari a "Fondamenti di Programmazione" e un carattere 'd', chiama la funzione `my_strchr(...)` e stampa a video la posizione del carattere nella C-stringa. La posizione stampata deve essere un numero intero, considerando 0 come prima posizione.

Output di esempio:

```
C-stringa: "Fondamenti di Programmazione"  
Carattere: 'd'  
Il carattere si trova in posizione: 3
```

Es. 6.8 – Conta parole in una C-stringa

Scrivere la funzione `int contaParole(...)` che restituisce il numero di parole presenti nella C-stringa. Si consideri come separatore di due parole una sequenza di uno o più caratteri spazio.

Ad esempio la seguente C-stringa ha 4 parole: "Il tempo e' bello".

Scrivere un programma che chiama la funzione `contaParole(...)` sulla C-stringa:

`"Lorem ipsum dolor sit amet "` (notare gli spazi multipli alla fine e tra le parole)
e poi sulla C-stringa:

`" Lorem ipsum dolor sit amet "` (notare gli spazi multipli all'inizio)
e stampa a video entrambi i risultati.

Output di esempio:

```
Numero di parole della prima C-stringa: 5  
Numero di parole della seconda C-stringa: 5
```