

Esercitazioni di Fondamenti di Programmazione

Anno accademico 2018/2019

Es. 5.1 – Puntatore

Scrivere una funzione $f()$ che ha come argomento un puntatore a intero. Sia n il valore dell'oggetto puntato. La funzione restituisce come valore di ritorno 0 se l'input non è valido; il quadrato di n se n è multiplo di 2; il cubo di n se n è multiplo di 3; il numero n altrimenti. La funzione stampa anche il valore del puntatore e il valore dell'oggetto puntato.

Scrivere un programma che accetta un intero da tastiera, chiama la funzione e stampa a video il valore restituito dalla funzione.

Output di esempio:

```
Inserisci k:  
10  
puntatore p: 0x7ffe0a0eba4c  
oggetto puntato da p: 10  
Il risultato e': 100
```

Es. 5.2 – Annullamento puntatore

Scrivere una funzione `inizializzaP()` che ha come argomento un puntatore a intero e come valore di ritorno `void`. La funzione assegna il valore NULL al puntatore.

Scrivere un programma che inizializza un intero e un puntatore ad esso, stampa il valore del puntatore, chiama la funzione, e stampa nuovamente il valore del puntatore.

Output di esempio:

```
Puntatore prima della chiamata: 0x7ffee6e7651c  
Puntatore dopo la chiamata: 0
```

Es. 5.3 – Copia di stringa

Scrivere una funzione `string_copy()` che prende in ingresso due stringhe `dest` e `src` e copia `src` su `dest`. La funzione non fa nulla se uno dei due parametri di ingresso non è valido. La funzione restituisce `void`. Assumere che gli spazi di memoria a cui puntano le due stringhe siano stati allocati dal chiamante.

Scrivere un programma che inizializza le stringhe `src` e `dest`, le stampa, chiama la funzione, e stampa nuovamente le stringhe.

Nota: Non utilizzare la funzione di libreria `strcpy()`.

Output di esempio:

```
dest = Prima stringa
src = Seconda stringa
Chiamo la funzione
dest = Seconda stringa
src = Seconda stringa
```

Es. 5.4 – Rotazione di un vettore

Scrivere una funzione `void traslaVettore(int vett[], int n)` che trasla a destra di una posizione ciascun elemento dell'array di interi `vett`, lungo `n` elementi. La traslazione è circolare, ovvero l'elemento in posizione `n-1` viene copiato in posizione `0`.

Scrivere un programma che:

- a) richiede l'inserimento da tastiera di 6 interi da tastiera e li memorizza in un vettore;
- b) richiede l'inserimento da tastiera di un numero di passi di traslazione `k`;
- b) trasla a destra il vettore di `k` passi (circolarmente), sfruttando la funzione `traslaVettore()`;
- c) stampa a video il vettore risultante.

Output di esempio:

```
Inserire gli elementi del vettore:
1
2
3
4
5
6
Inserire un numero di passi di traslazione:
2
Vettore risultante:
5 6 1 2 3 4
```