

Tecniche di Programmazione (2015/16)

Esercitazione 6

Argomento: funzioni e passaggio dei parametri

Esercizio 6.1

Scrivere una funzione C

```
void scambia(int *a, int *b);
```

che dati in ingresso due puntatori ad intero a e b, scambi i loro contenuto. Scrivere un adeguato main di prova per testare il corretto funzionamento della funzione.

Esercizio 6.2

Scrivere una funzione C

```
int risolviEquazione(float *x1, float *x2, float a, float b, float c);
```

che dati in input i coefficienti di un equazione di secondo grado a, b e c, scriva le soluzioni nelle variabili puntate da x1 e x2. La funzione deve ritornare un intero negativo, nullo o positivo a seconda che l'equazione abbia nessuna, una o due differenti soluzioni. Scrivere un adeguato main di prova per testare il corretto funzionamento della funzione.

Esercizio 6.3

Ripetere l'esercizio 3.5 definendo la funzione

```
void stampaAsterischi(int n);
```

che dato in ingresso un intero n stampi una riga di n asterischi.

Esercizio 6.4

Ripetere l'esercizio 2.3.b definendo la funzione

```
void stampaRigaAsterischi(int n, int fill);
```

che dati in input:

- un intero n
- un intero fill

stampi :

- una riga completa di n asterischi se il parametro fill e' diverso da zero,
 - una riga di spazi delimitata da un asterisco in posizione 0 e in posizione n-1.
- (ovvero la cornice di un quadrato)

Esercizio 6.5

Scrivere una funzione C

```
int* allocaEdInizializza(int n, int valore);
```

che dati in ingresso due interi n e valore, allochi una zona di memoria adatta a contenere n interi e la inizializzi con il contenuto del parametro valore. La funzione deve restituire il puntatore alla zona di memoria allocata. Inoltre, scrivere una funzione main che legga in input da tastiera i dati 'n' e 'valore', invochi la funzione allocaEdInizializza, verifichi la corretta allocazione e inizializzazione stampando i valori della memoria allocata (si usi l'aritmetica dei puntatori) e infine rilasci la memoria allocata.

Ultime modifiche: giovedì, 31 marzo 2016, 12:01