

Laboratorio di programmazione

Docente: Violetta Lonati

1 Esercizi da svolgere in laboratorio

1.1 Trasformazione orario in secondi

Scrivete una funzione con prototipo `void split_time (long int tot_sec, int *h, int *m, int *s)` che, dato un orario fornito in numero di secondi dalla mezzanotte, calcoli l'orario equivalente in ore, minuti, secondi, e lo memorizzi nelle tre variabili puntate da (h), (m) e (s) rispettivamente.

1.2 Scambio di valori

Scrivete una funzione con prototipo `void scambia(int *p, int *q)` che scambi i valori delle due variabili puntate da p e q.

1.3 Rovescia e palindroma

Rivedete gli esercizi “Rovescia” e “Palindrome” (scheda L02.pdf) scorrendo la stringa con un puntatore anziché con un contatore intero.

1.4 Da minuscolo a maiuscolo

Scrivete una funzione con prototipo `char *maiuscolo(char *stringa)` che trasformi da minuscolo in maiuscolo tutte le lettere del suo argomento `stringa` e ne restituisca un puntatore al primo carattere. Potete assumere che `stringa` sia dato da una stringa terminata da `'\0'` contenente caratteri ASCII (non solo lettere). Potete usare la funzione `toupper` della libreria `ctype.h`.

1.5 Lunghezza di una stringa

Scrivete una funzione con prototipo `int lung_stringa(char *s)` che, data una stringa `s`, ne calcoli la lunghezza. Provate a scrivere il programma usando un puntatore a carattere per scorrere la stringa.

1.6 Rettangoli

Modificate il programma dell'esercizio “Figure geometriche” (scheda L02.pdf) scrivendo delle funzioni che svolgano le seguenti operazioni:

- dati per argomenti due punti, creare e restituire una nuova struttura rettangolo;
- stampare i dati del rettangolo passato come argomento;

- calcolare l'area del rettangolo passato come argomento;
- calcolare il centro (l'intersezione delle diagonali) del rettangolo passato come argomento;
- traslare il rettangolo passato come argomento di x unità nella direzione x e y unità nella direzione y ;
- stabilire se un punto p cade dentro il rettangolo passato come argomento oppure no, restituendo VERO o FALSO.

NOTA: Per evitare che ad ogni chiamata venga copiata tutta la struttura, è utile passare come argomento un puntatore al rettangolo. In questo caso, sarà utile l'operatore \rightarrow .

1.7 I due valori più grandi

Scrivete una funzione con prototipo `void max_secondmax (int a[], int n, int *max, int *second_max)` che, dato un array a di lunghezza n individui il valore più grande in a e il secondo elemento per grandezza in a , e li memorizzi nelle variabili puntate da max e $second_max$.

1.8 Puntatore al minimo

Scrivete una funzione con prototipo `int *smallest(int a[], int n)` che, dato un array a di lunghezza n , restituisca un puntatore all'elemento più piccolo dell'array.