

Complementi di Programmazione

Esercitazione 6

Usare un main per testare il corretto funzionamento delle funzioni implementate.

Nota: tutti gli esercizi vanno svolti in maniera ricorsiva

Ricorsione Array

Esercizio 6.1

Scrivere una funzione:

```
float product(float vec[], int n);
```

che, dati in ingresso un array `vec` e la sua lunghezza `n`, calcoli e restituisca il prodotto degli elementi di `vec`. (la ricorsione va eseguita sia su `vec` ed `n`).

Esercizio 6.2

Scrivere la funzione

```
void vec_integral(float* v, int n);
```

che scrive in ogni elemento dell'array `v[i]` la somma di tutti gli elementi che lo precedono da 0 a `i-1`.

Ricorsione Stringhe

Esercizio 6.3

Scrivere la funzione

```
int length(char * s)
```

che, data in ingresso una stringa, calcoli e restituisca la lunghezza della stringa.

Esercizio 6.4

Scrivere la funzione

```
int char_position(char* s, char ch);
```

che, data in ingresso una stringa e un carattere, calcoli e restituisca la posizione del primo carattere `ch` nella stringa. Se il carattere non e' presente, la funzione deve restituire -1.

Esercizio 6.4 variante

Scrivere la funzione

```
int char_last_position(char* s, char ch);
```

che, data in ingresso una stringa e un carattere, calcoli e restituisca la posizione dell'ultimo carattere `ch` nella stringa. Se il carattere non e' presente, la funzione deve restituire -1.

Esercizio 6.5

Scrivere la funzione

```
bool is_lowercase(char *s)
```

che, data in ingresso una stringa, restituisca true se la stringa contiene solo lettere minuscole, false altrimenti.

Esercizio 6.5 variante

Scrivere la funzione

```
bool is_any_lowercase(char *s)
```

che, data in ingresso una stringa, restituisca true se la stringa contiene almeno una lettera minuscola, false altrimenti.

Esercizio 6.6

Scrivere la funzione

```
void to_uppercase(char *s);
```

che modifica la stringa in ingresso convertendo tutti i caratteri minuscoli in essa presenti nei corrispondenti caratteri maiuscoli.

Esercizio 6.7

Scrivere una funzione:

```
void copy(char * dest, char * src);
```

che, date in ingresso due stringhe, faccia la copia della stringa `src` dentro `dest`. Assumere che la memoria di `dest` sia già stata allocata e lo spazio sia sufficiente a contenere il risultato.

Esercizio 6.8

Scrivere una funzione:

```
void concat(char * dest, const char * src);
```

che, date in ingresso due stringhe, concateni la stringa `src` a `dest` e la memorizzi in `dest`. Assumere che la memoria di `dest` sia già stata allocata e lo spazio sia sufficiente a contenere il risultato.

Ricorsione File

Esercizio 6.9

Scrivere una funzione:

```
void filter_write(char * message, FILE * file);
```

che dato in input una stringa con un messaggio e un puntatore a file già aperto in scrittura, scriva nel file tutti i caratteri nel messaggio compresi tra 'a' e 'z' (niente maiuscole, numeri o caratteri speciali).

Esercizio 6.10

Scrivere una funzione:

```
bool same_content(FILE *f1, FILE *f2);
```

che, ricevuti in input due puntatori a file già aperti in lettura, ritorni true se il contenuto dei due file è uguale.