UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dispense del Corso di Laboratorio di Fondamenti di Informatica II e Lab

Esercitazione 02: Ricorsione

Ultimo aggiornamento: 06/03/2019

Ricorsione

- Esercizio 1:

Nel file `somma.c` implementare la definizione della funzione:

```
int Somma(int n);
```

La funzione accetta come parametro un numero intero positivo n e ritorna la somma dei primi n numeri naturali (0 escluso) calcolata ricorsivamente. Nel caso che n sia minore o uguale a 0 la funzione deve ritornare -1. Se ad esempio n vale 2 la funzione deve ritornare 3 (1 + 2).

Ricorsione

- Esercizio 2:

Nel file `fibonacci.c` implementare la definizione della funzione:

```
int Fibonacci(int n);
```

La funzione accetta come parametro un numero intero positivo n e ritorna l'n-esimo numero della successione di Fibonacci calcolato ricorsivamente. Una delle possibili formulazioni della successione è la seguente:

$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-1} + F_{n-2}$

Nel caso che n sia minore di zero la funzione deve ritornare -1.

Ricorsione

- Esercizio 3:

Nel file `minimo.c` implementare la definizione della funzione:

```
int Minimo(int *vec, int len);
```

La funzione accetta come parametri un puntatore ad array di interi (vec) e un numero intero positivo (len) che rappresenta la lunghezza dell'array. Il valore di ritorno deve essere il minimo tra gli elementi dell'array, calcolato ricorsivamente.

Si assuma che il puntatore vec sia sempre diverso da NULL e che len sia sempre maggiore o uguale ad 1.

Gli elementi dell'array da passare alla funzione `Minimo()` devono essere letti da un file di testo, dove sono memorizzati separati da spazi.

Ricorsione

- Esercizio 4:

Nel file `prodotto.c` implementare la definizione della funzione:

```
int Prodotto(int a, int b);
```

La funzione accetta come parametri due numeri interi positivi e ritorna il loro prodotto calcolato ricorsivamente utilizzando solo la somma.

Se uno dei due numeri è negativo la funzione deve ritornare -1.

Ricorsione

- Esercizio 4 - bis:

Nel file `prodotto_negativi.c` implementare la definizione della funzione:

```
int ProdottoNegativi(int a, int b);
```

La funzione accetta come parametri due numeri interi e ritorna il loro prodotto calcolato ricorsivamente utilizzando solo la somma.

Ricorsione

- Esercizio 5:

Nel file `fattoriale.c` implementare la definizione della funzione:

```
unsigned long long Fattoriale(int n);
```

La funzione accetta come parametro un numero intero n e ritorna il suo fattoriale (n!) calcolato ricorsivamente.

Se n è minore di zero la funzione deve ritornare 0.

Ricorsione

- Esercizio 6:

Nel file `divisione.c` implementare la definizione della funzione:

```
int Divisione(int a, int b);
```

La funzione accetta come parametri due numeri interi positivi e ritorna il loro quoziente (a/b) calcolato ricorsivamente utilizzando solo la sottrazione.

Se uno dei due valori è negativo o se la divisione è impossibile la funzione deve ritornare -1.

Ricorsione

- Esercizio 6 - bis:

Nel file `divisione_negativi.c` implementare la definizione della funzione:

```
int DivisioneNegativi(int a, int b);
```

La funzione accetta come parametri due numeri interi e ritorna il loro quoziente (a/b) calcolato ricorsivamente utilizzando solo la sottrazione.

Se la divisione è impossibile la funzione deve ritornare INT_MAX.