

Laboratorio di algoritmi e strutture dati

Docente: Violetta Lonati

Esercizi introduttivi sulle funzioni in C*

1 Garibaldi fu ferito

Considerate il codice contenuto nel file `garibaldi.c`. Analizzate il codice sorgente e rispondete, possibilmente per iscritto, alle seguenti domande. Se avete dubbi, potete testarlo, eseguendolo su casi di input significativi e modificandolo.

```
1 #include <stdio.h>
2
3 char garibaldi( char a, char b ){
4     if ( b == 'a' || b == 'e' || b == 'i' || b == 'o' || b == 'u' )
5         return a;
6     else return b;
7 }
8
9 int main() {
10     char c, vocale;
11
12     vocale = getchar();
13     getchar();
14
15     while ( ( c = getchar() ) != '.' )
16         printf( "%c", garibaldi( vocale, c ) );
17
18     printf( "\n" );
19     return 0;
20 }
```

1. Senza eseguire il programma al computer, simulatene l'esecuzione su carta e stabilite cosa stampa il programma quando riceve da standard input la sequenza di caratteri:
u garibaldi fu ferito, fu ferito in una gamba.
2. Descrivete a parole cosa fa la funzione `garibaldi`.
3. Riassumete a parole cosa fa il programma.

2 Primo

Scrivete una funzione con parametro un intero n che stabilisca se n è un numero primo. Scrivete la funzione partendo dal programma che avete scritto per l'esercizio 5 della scheda "L01-lab".

*Ultima modifica 18 ottobre 2019

3 Potenza

Scrivete una funzione ricorsiva avente due parametri interi b ed e che calcoli la potenza b^e .

4 Sequenze di Collatz

Considerate la seguente regola: dato un numero intero positivo n , se n è pari lo si divide per 2, se è dispari lo si moltiplica per 3 e si aggiunge 1 al risultato. Quando n è 1 ci si ferma.

Questa semplice regola permette di costruire delle sequenze: la sequenza che si costruisce a partire dal numero n è detta *sequenza di Collatz di n* . Ad esempio, la sequenza di Collatz di 7 è:

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

E' un noto problema aperto stabilire se ogni sequenza di Collatz termina (cioè, se arriva a 1).

Scrivete innanzitutto una funzione che, dato un numero, dia il successivo in una sequenza di Collatz. Quindi, inseritela in un programma che chiede all'utente un numero e mostra la sequenza di Collatz del numero (con tanto di lunghezza).

Esempi di funzionamento

```
Numero: 7
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Lunghezza: 17
```

```
Numero: 9
9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Lunghezza: 20
```

5 Ricorsione e iterazione

Considerate questa funzione ricorsiva `f_rec`

```
1 unsigned long f_rec(int n){
2     if ( n == 1 || n == 2 ){
3         return 1;
4     }
5     return f_rec( n - 1 ) + f_rec( n - 2 );
6 }
```

Senza eseguire la funzione al computer, rispondete alle seguenti domande:

1. Cosa restituisce la funzione $f(7)$?
2. Perché n è dichiarato come intero mentre il valore restituito è di tipo `unsigned long`?
3. Riassumete a parole cosa restituisce la funzione se riceve come argomento un intero positivo n maggiore di 0.

Considerate ora le due funzioni `f_iter1` e `f_iter2`

```

1 unsigned long f_iter1(int n){
2     unsigned long f, f1 = 1, f2 = 1;
3     if ( n == 2 | n == 1 ) {
4         return 1;
5     }
6     while (n-- >= 3){
7         f = f1 + f2;
8         f1 = f2;
9         f2 = f;
10    }
11    return f;
12 }

```

```

1 unsigned long f_iter2(int n){
2     unsigned long f, f1 = 1, f2 = 1, i;
3     if ( n == 2 | n == 1 ) {
4         return 1;
5     }
6
7     for ( i = 2; i <= n; i++ ) {
8         f = f1 + f2;
9         f1 = f2;
10        f2 = f;
11    }
12    return f;
13 }

```

Senza eseguire la funzione al computer, rispondete alle seguenti domande:

4. Considerando solo il valore restituito, le due funzioni sono equivalenti? (ovvero: restituiscono sempre lo stesso valore?)
5. Le due funzioni sono equivalenti alla funzione `f_rec`?
6. Modificate (se necessario) le funzioni `f_iter1` e `f_iter2` in modo che risultino essere equivalenti a `f_rec`.
7. Stimate il numero di operazioni che si svolgono durante l'esecuzione di `f_rec`, `f_iter1` e `f_iter2`: sono paragonabili?

Considerate infine la seguente funzione ricorsiva `f_riter`

```

1 unsigned long f_riter(unsigned long a, unsigned long b, int n){
2     if ( n == 2 ) {
3         return a;
4     }
5     if ( n == 1 ) {
6         return b;
7     }
8     return f_riter( a + b, a, n - 1 );
9 }

```

Senza eseguire la funzione al computer, rispondete alle seguenti domande:

8. Convincetevi che questa funzione può essere usata per calcolare `f_rec`. In particolare: con quali argomenti devo invocare `f_riter` per ottenere il valore restituito da `f_rec(n)`?
9. Rappresentate graficamente lo schema delle chiamate ricorsive definiti dall'invocazione `f_rec(7)` e dalla chiamata equivalente del tipo `f_riter(..., ..., ...)`.
10. Considerate il numero di chiamate ricorsive effettuate da `f_rec(n)` e dalla chiamata equivalente del tipo `f_riter(..., ..., ...)`. Sono paragonabili?
11. Usate una variabile globale `counter` per tenere traccia del numero delle chiamate ricorsive; quindi scrivete un programma che invoca `f_rec` e `f_riter` e stampa, oltre al valore restituito, anche il numero di chiamate della funzione.

Una volta concluso l'esercizio potete usare il file `rec-iter.c` per fare degli esperimenti. Analizzate il codice per capire come usarlo e come interpretarne l'output!