Fondamenti di Informatica e Laboratorio T-AB e Fondamenti di Informatica T1
Ingegneria Elettronica e Telecomunicazioni e
Ingegneria dell'Automazione
a.a. 2010/2011

Lab 11
Funzioni ricorsive

Esercizio 1

Scrivere una funzione ricorsiva:

int ric(int x)

che calcoli, ricorsivamente, la somma di tutti i numeri compresi tra 0 ed x.

Esercizio 1 – Soluzione

```
int ric(int x) {
  if (x == 0)
     return 0;
  else
     return x + ric(x-1);
}
```

Esercizio 2

Scrivere una procedura ricorsiva:

void print(int list[], int length)

che stampi, *ricorsivamente*, tutti i numeri contenuti nell'array **list**.

Esercizio 2 – Soluzione

```
void print (int list[], int length) {
   if (length != 0)
   {   print(list, length-1);
       printf("%d\n", list[length-1]);
   }
}
```

Esercizio 3

Scrivere una funzione ricorsiva che, ricevuto in ingresso un array di interi, esegua la somma degli interi in posizione con indice dispari.

Esercizio 3 – Soluzione

```
int sumOdd2(int list[], int length, int pos) {
   if (pos >=length)
       return 0;
   else
       return list[pos] + sumOdd2(list, length, pos+2);
}

int sumOdd(int list[], int length) {
   return sumOdd2(list, length, 1);
}
```

Esercizio 3 – Soluzione

(variante)

```
int sumOdd2(int list[], int length, int pos) {
   if (pos >=length)
        return 0;
   else
        return list[pos] + sumOdd2(list, length, pos+2);
}

int sumOdd(int list[], int length) {
   if (length >=0) {
        if ( lenghth>=0 && ((length-1)%2) == 1 )
               return sumOdd(list, length-1);
        else
               return list[length-1] + sumOdd(list, length-2);
   }
   else
        return 0;
}
```

Esercizio 4

Si scrivano le versioni ricorsiva ed iterativa (utilizzo di while) di una funzione:

double f(double a, int n);

che calcoli il seguente valore:

$$\sum_{i=1}^{n}\left(a-\frac{i}{a}\right)$$

Esercizio 4 – Soluzione

```
double f(double a, int n)
{ if (n==1) return a - 1/a;
  else return a - n/a + f(a, n-1);
}

double f(double a, int n)
{ int i=1;
  double sum=0;
  while(i<=n)
       {sum = sum + a - i/a;
        i++;}
  return  sum;
}
```

Esercizio 5

Si definisca una procedura ricorsiva:

void somme2(int l1[], int length)

che, ricevuto in ingresso un array l1 di interi, stampi a video gli interi dell'array di ingresso il cui valore è uguale alla somma dei due interi seguenti nell'array (a tal fine, gli ultimi due numeri di un'array sono automaticamente esclusi). Ad esempio, se invocata con l1 = [5, 6, 4, 2, 1, 1, 3, 1], la procedura deve stampare [6, 2]. Infatti, considerando il primo valore (5): i due valori successivi sono 6 e 4, e la loro somma vale 10; quindi 5 è scartato. Per il secondo valore, pari a 6, la somma dei due valori successivi è proprio 6 e quindi il valore viene selezionato per la stampa.

Esercizio 5 – Soluzione

```
void somme2(int l1[], int length) {
   if (length<3)
       return;
   else {
       if (l1[0] == l1[1] + l1[2]) {
             printf("%d, ", l1[0]);
       }
       somme2(&(l1[1]), length-1);
   }
}
```

Esercizio 6

Si scriva una programma che inverta le cifre di un numero intero N usando una funzione apposita. A tal fine, si realizzi sia una versione ricorsiva, sia una versione iterativa della funzione.

Per esempio:

dato N=4325, il programma stampa: 5234

Esercizio 6 – Soluzione

```
int reverse2(int num, int part) {
   if (num == 0)
       return part;
   else {
       return reverse2(num/10, part*10 + num%10);
   }
}

int reverse_it(int num) {
   int result = 0;
   while (num!=0) {
       result = result*10 + num%10;
       num = num/10;
   }
   return result;
}
```