

Università degli Studi di Verona  
Dipartimento di Informatica  
Corso di Laurea in Informatica

---

# ELABORATO ASM

Architettura degli elaboratori

Ilaria Chitò VR370279

Francesca Filini VR369132

---

Anno Accademico 2012/2013

## Indice

---

<b>1</b>	Presentazione elaborato	pag. 3
<b>2</b>	Variabili	pag. 4
<b>3</b>	Funzioni	pag. 5
<b>4</b>	Codice C	pag. 6
<b>5</b>	Scelte progettuali	pag. 7

# 1 Presentazione elaborato

Si intende realizzare un software che calcoli “in quanti modi  $n$  coppie possono sedersi a un tavolo circolare in modo che ci sia sempre un uomo seduto tra due donne e che nessun uomo sieda vicino a sua moglie”. La formula utilizzata è la *ricorrenza di Laisant*:

$$A_n = \frac{(n^2 - 2n)A_{n-1} + nA_{n-2} + 4(-1)^{n-1}}{n - 2}$$

con  $A_2 = 0, A_3 = 1$ .

Il programma chiede all'utente di inserire due numeri interi  $n0$  e  $nu$ , con  $nu > n0 > 1$ , verificando la loro correttezza (le specifiche non richiedono di verificare che siano effettivamente numeri interi).

L'output sarà del tipo, ad esempio con  $n0 = 3$  e  $nu = 6$ :

n	An
3	1
4	2
5	13
6	80

Il valore  $A_n$  viene calcolato mediante una procedura avente struttura ricorsiva sulla base della ricorrenza di Laisant e dei casi base. Tale procedura riceve in ingresso un solo parametro  $n0$  e restituisce solamente il valore calcolato  $A_n$ .

In accordo con le specifiche, per i calcoli vengono utilizzate solamente istruzioni in aritmetica intera.

## 2 Variabili

---

Nel file “sorgente.s” sono state utilizzate le seguenti variabili:

- **inserisci\_primo\_numero** (di tipo ascii): serve per chiedere all’utente di inserire il numero iniziale di coppie;
- **inserisci\_primo\_numero\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `inserisci_primo_numero`;
- **inserisci\_secondo\_numero** (di tipo ascii): serve per chiedere all’utente di inserire il numero finale di coppie;
- **inserisci\_secondo\_numero\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `inserisci_secondo_numero`;
- **errore** (di tipo ascii): messaggio di errore nel caso in cui i numeri inseriti non siano validi;
- **errore\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `errore`;
- **prima\_riga\_output** (di tipo ascii): serve per stampare l’intestazione dell’output;
- **prima\_riga\_output\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `prima_riga_output`;
- **stringa\_spazio** (di tipo ascii): serve per stampare uno spazio;
- **stringa\_spazio\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `stringa_spazio`;
- **stringa\_a\_capo** (di tipo ascii): serve per andare a capo nell’output;
- **stringa\_a\_capo\_len** (di tipo long): numero intero che rappresenta la lunghezza della stringa contenuta nella variabile `stringa_a_capo`.

Nei file: “atoi.s”, “atoi2.s” e “itoa.s” è stata utilizzata la variabile:

- **car** (di tipo byte): serve come puntatore alla stringa da stampare.

### 3 Funzioni

---

Nei file **atoi.s** e **atoi2.s** viene sviluppata la funzione che serve per acquisire un parametro da tastiera. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- vengono caricate nello stack i valori dei registri EBX, ECX, EDX in modo da poterli utilizzare senza perdere le informazioni già registrate;
- vengono eseguite le operazioni di lettura del carattere;
- vengono ripristinati i registri salvati sullo stack;
- ritorno alla funzione chiamante.

I file **atoi.s** e **atoi2.s** differiscono per un'operazione iniziale di caricamento sullo stack del registro EBX, la quale non era necessaria nella seconda funzione.

Nel file **itoa.s** viene sviluppata la funzione che serve per convertire un numero in stringa e stampare un numero decimale un carattere alla volta. In seguito alla dichiarazione della variabile globale e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- vengono caricate nello stack i valori dei registri EAX, EBX, ECX in modo da poterli utilizzare senza perdere le informazioni già registrate;
- vengono eseguite le operazioni di stampa del carattere;
- vengono ripristinati i registri salvati sullo stack;
- ritorno alla funzione chiamante.

Il file **sorgente.s** è il programma principale che, richiamando le varie funzioni, calcola e stampa la ricorrenza di Laisant. In seguito alla dichiarazione delle variabili globali, nella sezione istruzioni appaiono le seguenti operazioni:

- viene richiesto all'utente di inserire il primo numero che viene acquisito tramite la chiamata alla funzione "atoi";
- viene richiesto all'utente di inserire il secondo numero che viene acquisito tramite la chiamata alla funzione "atoi2";

- viene verificata la correttezza dei numeri;
- viene stampata la prima riga dell'output;
- inizio del ciclo che calcola e stampa i valori che assume la funzione "ricorsione" per il parametro passato.

## 4 Codice C

---

Per sviluppare il software, ci si è basati sul seguente programma in linguaggio ad alto livello:

```
#include <stdio.h>

//Funzione che calcola se l'esponente è pari o dispari
int esponente (int esp) {
    if (esp % 2 == 0)
        return 1;
    else
        return -1;
}

//Funzione ricorsiva di Laisant
int coppia (int n0) {
    if (n0 == 2)                //Caso base: n0 = 2
        return 0;
    else if (n0 == 3)           //Caso base: n0 = 3
        return 1;
    else
        return (((n0*n0 - (2*n0)) * coppia(n0-1) + (n0 *
coppia(n0-2)) + 4 * (esponente(n0-1))) / (n0-2));
}

//Funzione principale
int main (void) {
    int n0, nu, res, i;

    printf ("Inserisci il primo numero: "); scanf ("%i", &n0);
    printf ("Inserisci il secondo numero: "); scanf ("%i", &nu);
```

```
    for (i = n0; i <= nu; i++) {  
        res = coppia (i);  
        printf ("%i  %i\n", i, res);  
    }  
    return 0;  
}
```

## 5 Scelte progettuali

---

Si è scelto di suddividere il programma in più file per aver maggiore chiarezza nella fase progettuale.

Nel caso che il numero di coppie inserito superi il numero 13 si presenta un caso di overflow nei registri. Per tale motivo si è supposto che non venga inserito un numero maggiore di 13.