

# Johnny1.00

Simulazione semplificata di un  
Computer di von Neumann

Peter Dauscher, 2009-2012



**Manuale utente**



## Contenuto

1. Osservazioni .....	3
2. Introduzione .....	3
3. Semplificazioni .....	3
4. Il Simulatore .....	4
4.1. La Memoria (RAM) .....	5
4.2. L' Unità Aritmetico-Logica .....	5
4.3. L'Unità di Controllo .....	6
5. Il Set di istruzioni (MACRO) ....	7
6. Semplici esempi di programmi .....	8
6.1. Somma .....	8
6.2. Moltiplicazione .....	8
7. L'interfaccia utente .....	9
8. La creazione di istruzioni MACRO .....	11
9. La modalità BONSAI .....	12
9.1. Passare in modalità BONSAI .....	12
9.2. Apri e Salva. ....	12
10. Informazioni legali, materiale tecnico e Ringraziamenti. ....	13
10.1. Informazioni legali .....	13
10.2. Informazioni tecniche .....	14
10.3. Ringraziamenti. ....	15
10.4. Note per la lingua italiana .....	16

## 1 . Osservazioni (per il manuale in italiano)

La traduzione in italiano del manuale è scaturita da un'idea dei docenti di TPSI (Tecnologie e Progettazione di Sistemi Informatici e di Telecomunicazioni) del terzo anno dell'istituto tecnico IIS "A.Meucci" di Casarano (Le).

Nella programmazione didattica è prevista l'introduzione all'architettura del personal computer con la descrizione della macchina di Von Neumann.

Cercando nella rete è stato trovato questo ottimo simulatore off-line di Peter Dauscher (squisita persona che ha accolto con entusiasmo il progetto di traduzione fornendo utili consigli).

Grazie al simulatore i ragazzi sono stati in grado di capire meglio un argomento molto importante acquisendo nel contempo nozioni propedeutiche al linguaggio assembler.

## 2 . Introduzione

L'architettura dei moderni computer si basa sul modello pubblicato da Von Neumann a metà degli anni '40. A causa della miniaturizzazione, la maggior parte dei processi sono nascosti agli utenti e non possono essere osservati. Pertanto, la simulazione è un metodo efficace per dare agli studenti una panoramica su ciò che accade all'interno del computer.

JOHNNY è stato sviluppato appositamente per scopi didattici di conseguenza risulta semplificato rispetto ai computer reali.

## 3 . Semplificazioni

- Il simulatore può essere osservato a livello di micro istruzioni nonché di macro istruzioni. La struttura interna della Control Unit può essere nascosta in modo da rendere inizialmente le cose più semplici.
- Gli utenti possono programmare il simulatore che utilizza un linguaggio tipo assembler. Le relative macro istruzioni possono essere scritte direttamente nella memoria utilizzando un'interfaccia grafica in modo da evitare errori di sintassi.
- L'Unità Aritmetico-Logica è composta da un unico registro usato come accumulatore.
- Il simulatore e la sua interfaccia grafica utilizzano il sistema di numerazione decimale. Anche se questo tecnicamente non è molto realistico, rende più semplice l'utilizzo soprattutto ai principianti che non hanno troppa confidenza i numeri esadecimale. Quindi, in Johnny,  $9+8$  uguale 17 e non  $0x11$ .
- L'intervallo dati è 0..19999; l'intervallo di indirizzi è 0..999. Gli Overflow non sono gestiti:  $0-1 \rightarrow 0$  e  $19999+1 \rightarrow 19999$ .
- Le istruzioni usate sono solo 10. Tutte le istruzioni utilizzano un indirizzamento assoluto (un indirizzo per l'istruzione) per ragioni di semplicità. Il codice dell'operazione (OP) è composto dalle migliaia e dalle centinaia; la prima posizione rappresenta l'indirizzo (ADD 42 è rappresentato da 02 042).

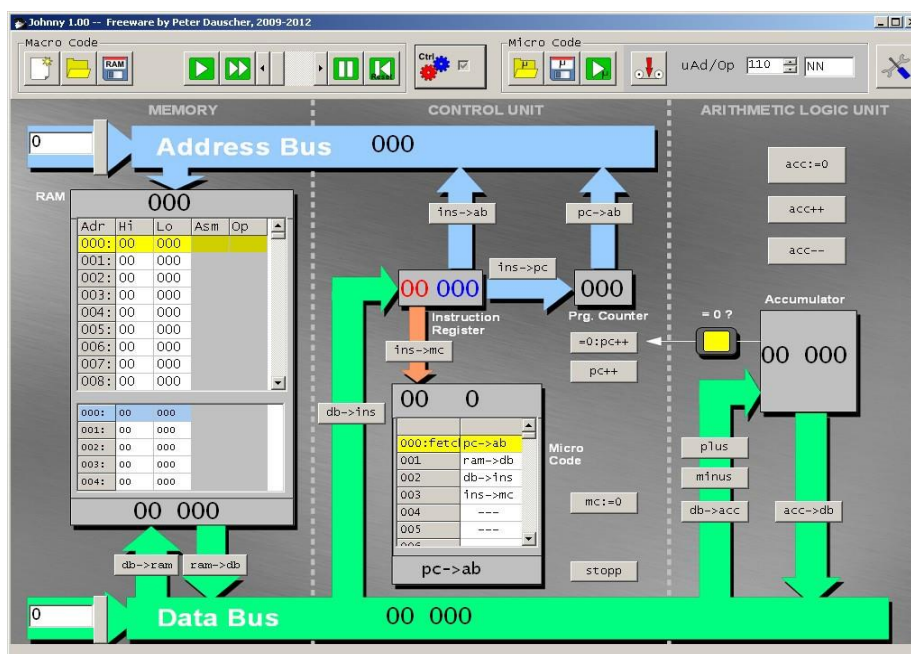
- Nei processori reali una micro istruzione, in genere, attiva diverse linee del control bus. In questo caso, se l'istruzione non è corretta, si possono verificare facilmente degli errori. Pertanto, è stata fatta una semplificazione (alquanto irrealistica): ogni micro istruzione corrisponde ad un singolo pulsante in modo che possa essere premuto manualmente dall'utente.
- Una macro istruzione è semplicemente una sequenza di tali micro istruzioni. Al fine di evitare problemi di conflitti, il bus dati ha la possibilità di memorizzare i dati di tipo word (cioè naturalmente non è realistico: i bus reali sono semplicemente linee; la memorizzazione viene fatta dai registri del processore).

Il trasferimento di un dato da un posto all'altro tramite il bus dati viene eseguita in due passi: a) il mittente copia il dato sul bus; b) il ricevente copia il dato dal bus. Conflitti di bus sono quindi esclusi.

- Un Micro Codice è modificabile: l'utente può creare lui stesso una macro istruzione semplicemente scegliendo un nome appropriato e poi cliccando la sequenza di micro istruzioni con il mouse.

### 3 . Il Simulatore

Il simulatore è costituito da tre parti: la memoria (RAM), l'unità aritmetico-logica e l'unità di controllo. Queste unità sono collegate attraverso i bus. Di seguito ogni unità sarà spiegata nel dettaglio.

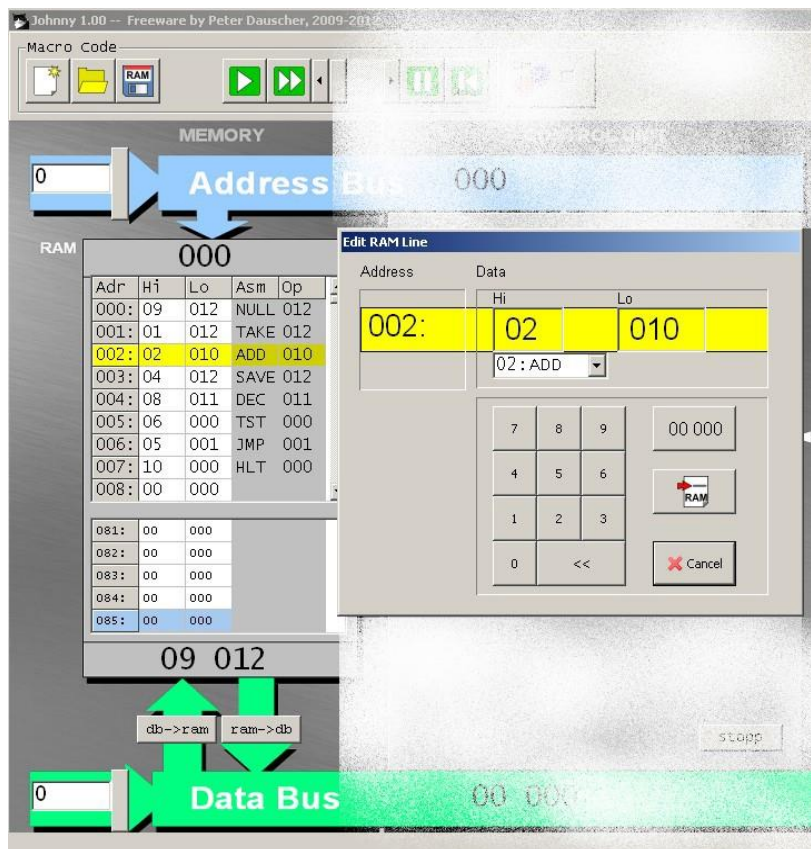


#### 4.1. La Memoria (RAM)

La memoria ad accesso casuale (RAM) è formata da 1000 locazioni, ciascuna ha la capacità di memorizzare numeri compresi nell'intervallo 0 ...19999. Di conseguenza, tre cifre decimali sono sufficienti per indirizzare ogni locazione.

La colonna **Hi** e la colonna **Lo** sono separate un po' poiché rappresentano, rispettivamente, il codice operativo e l'indirizzo dell'operando della macro istruzione.

Premendo il pulsante **ram->db** i dati possono essere copiati dalla locazione di memoria sul bus dati; premendo **db->ram** avviene il contrario.



Le locazioni sono modificabili utilizzando la GUI; le macro istruzioni possono essere scelte utilizzando il menù. Nella GUI sono mostrate due sezioni della RAM (che possono sovrapporsi). Così le istruzioni e i dati possono essere mostrati contemporaneamente.

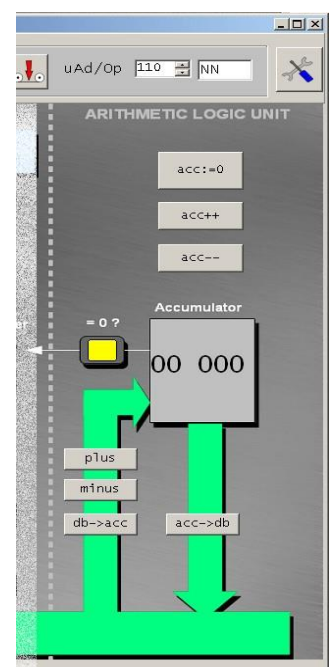
#### 4.2. L'Unità Aritmetico-logica

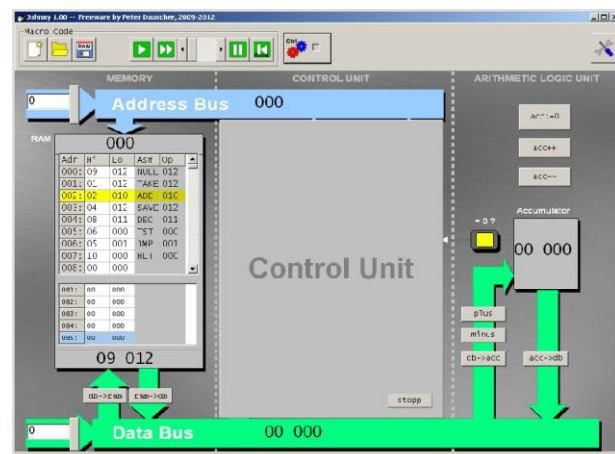
L'Unità Aritmetico Logica è formata soltanto dall'accumulatore.

L'accumulatore può essere resettato (**acc:=0**), incrementato (**acc++**), decrementato (**acc--**). **db->acc** trasporta un dato dal bus nell'accumulatore; **acc->db** fa il contrario.

Un valore del data bus può essere aggiunto (**plus**) o sottratto (**minus**).

Nella cosiddetta modalità BONSAI (spiegata dopo nella sezione 9) alcune di queste micro istruzioni sono state eliminate.







## 1. Il Set di istruzioni (MACRO)

Il set di macro istruzioni include 10 comandi standard:

- **TAKE** Il valore della locazione (indicato dall'indirizzo assoluto) è copiato nell'accumulatore.
- **SAVE** Il valore presente nell'accumulatore è copiato nella locazione di memoria indicata dall'indirizzo assoluto.
- **ADD** Il valore della locazione (indicato dall'indirizzo assoluto) è sommato al contenuto dell'accumulatore.
- **SUB** Il valore della locazione (indicato dall'indirizzo assoluto) è sottratto al valore contenuto nell'accumulatore.
- **INC** Il valore della cella (dato dall'indirizzo assoluto) è incrementato di 1.
- **DEC** Il valore della cella (dato dall'indirizzo assoluto) è decrementato di 1.
- **NULL** Il valore nella locazione di memoria indicata (in indirizzo assoluto) è impostato a zero.
- **TST** Se la cella indicata (in indirizzo assoluto) ha come valore lo zero, la successiva istruzione verrà saltata, altrimenti sarà eseguita. (Questo è un cambiamento sostanziale rispetto alla versione 0.98 del simulatore. Il cambiamento è stato fatto per uniformità).
- **JMP** Il programma prosegue saltando alla locazione indicata.
- **HLT** Il programma termina visualizzando un messaggio.



## 2. Semplici esempi di programmi

### 6.1. Somma

Il seguente programma di esempio somma i valori delle celle <10> e <11> e memorizza il risultato nella locazione 12:

001:	TAKE	010
002:	ADD	011
003:	SAVE	012
004:	HLT	000

### 6.2. Moltiplicazione

La moltiplicazione può essere implementata usando la tecnica delle addizioni successive.

I valori dei 2 fattori della moltiplicazione si inseriscono nelle celle 10 e 12; nella cella 12 si conserva il risultato delle addizioni. Ogni volta che si effettua un'addizione si decrementa di 1 il valore della cella 11. Si controlla, poi, il valore presente nella cella 11, se è diverso da 0 si continua con l'operazione di somma, altrimenti ci si ferma.









000:	NULL	012
001:	TAKE	012
002:	ADD	010
003:	SAVE	012
004:	DEC	011
005:	TST	011
006:	JMP	001
007:	HLT	000








### 3. L'interfaccia Utente

I tasti sono disposti in due gruppi (Macro e Micro codice) l'ultimo dei quali viene visualizzato solo se l'unità di controllo è mostrata in dettaglio.



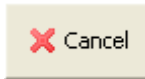
I pulsanti nel gruppo di Macro codice:

	Azzera la RAM (00 000 per ogni locazione)
	Apri un programma.
	Salva il programma corrente.
	Esegue la prossima Macro Istruzione presente nella RAM.
	Esegue il programma automaticamente (la velocità può essere controllata dalla barra di scorrimento a destra del pulsante).
	Ferma l'esecuzione del programma.
	Azzera il program counter e tutti gli altri registri.
	Mostra la finestra delle opzioni.

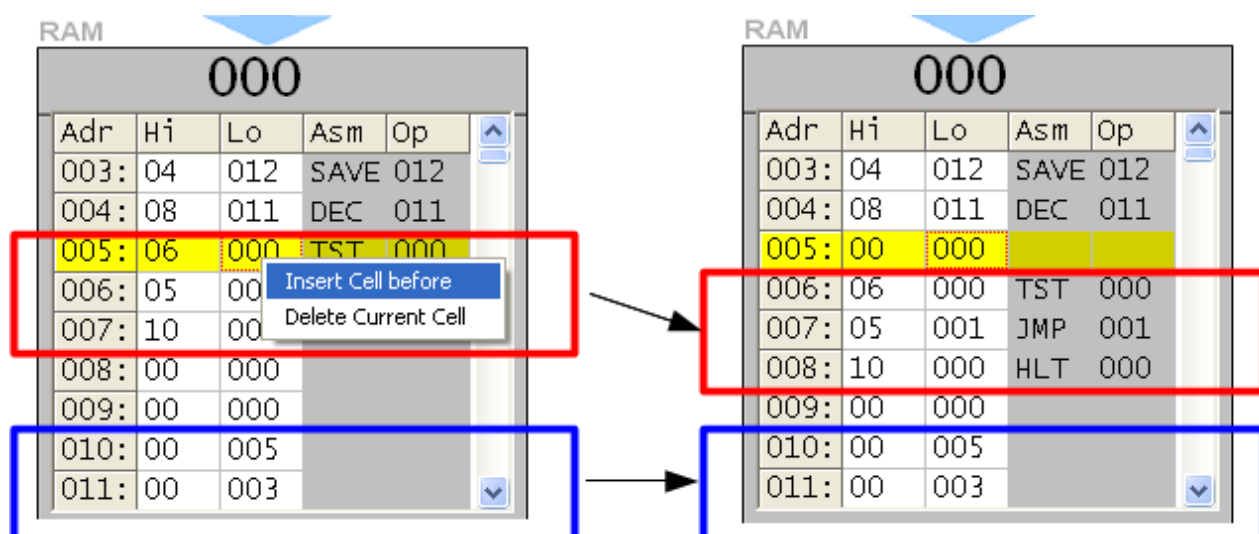
Quando viene selezionato il pulsante , per mostrare l'unità di controllo in dettaglio, sono visualizzati ulteriori pulsanti:

	Apri un file di micro codice.
	Salva il micro codice corrente.
	Esegui una singola micro istruzione.
	Registra una sequenza di micro istruzioni per creare una nuova macro istruzione.

Se si fa clic su una posizione della RAM, viene visualizzata una finestra nella quale si può modificare il valore numerico o può essere scelta una macro istruzione. L'indirizzo può essere inserito usando la tastiera reale o quella virtuale. Un doppio clic sulla zona dell'indirizzo imposta quest'ultimo a zero.

	La locazione è impostata a zero.
	Salva il dato o l'istruzione inserita nella RAM.
	Annulla le modifiche.

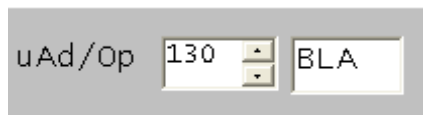
Usando il tasto destro del mouse, viene visualizzato un menu contestuale. Questo menu consente l'inserimento o la cancellazione di una riga.



Con l'inserimento, il blocco delle locazioni successive, diverse da zero, è spostato sotto di una riga. Viene cancellata una locazione con valore zero alla fine del blocco. Analogamente, con la cancellazione di una riga, il blocco con il valore diverso da zero torna su e la locazione con valore zero viene inserita sotto il blocco.

## 4. La creazione di istruzioni MACRO

Al fine di creare una propria macro istruzione (o per cambiarne una esistente) deve essere scelto prima un codice operativo e poi uno mnemonico. Quest'ultimo comparirà nel menù a tendina insieme alle istruzioni predefinite.



Dopo si preme il tasto di registrazione:



In un primo momento la parte del micro codice è impostata a zero. Premendo i pulsanti delle micro istruzioni, viene registrata la sequenza. La registrazione in corso è indicata da una zona lampeggiante dell'interfaccia utente. Premendo di nuovo il pulsante di registrazione essa si interrompe.

Il nuovo micro codice può essere salvato su memoria di massa e riaperto successivamente.



Il salvataggio di un micro codice crea due file: (.mpc), che contiene il micro codice stesso, (.nam), che archivia le istruzioni mnemoniche.

## 5. La modalità BONSAI

Nel 1990 Klaus Merkert e Walter Zimmer hanno realizzato un simulatore simile, BONSAI. Al fine di poter utilizzare il set di istruzioni BONSAI (che consiste solo di cinque istruzioni, vale a dire **INC**, **DEC**, **TST**, **JMP** e **HLT**), Johnny può essere utilizzato in tale modalità.

Il menù a discesa sarà adattato a queste istruzioni, alcune micro istruzioni (non necessarie in questa modalità) saranno soppresse.

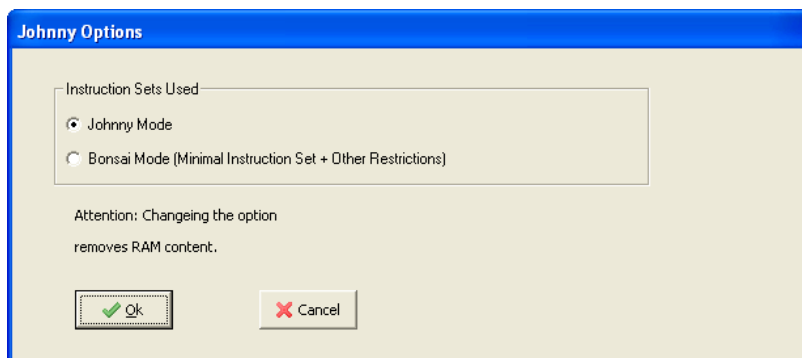
### 9.1. Passare in modalità Bonsai

Utilizza



Viene visualizzata una finestra che permette all'utente di cambiare la modalità.

Attenzione: cambiando modalità il contenuto della RAM sarà completamente cancellato.



### 9.2. Apri e Salva

I file del simulatore BONSAI originale (.bma) possono essere aperti e salvati. Pertanto, è possibile creare un programma macchina BONSAI utilizzando JOHNNY e poi trasferirlo nel simulatore originale (che è più complesso ma anche più realistico).

Quindi ci sono tre tipi di file per salvare il contenuto della RAM:

.ram	Formato standard di JOHNNY
.bma	Programmi standard della Bonsai-macchina
.bij	Programmi Bonsai salvati in formato standard di JOHNNY

## 6. Informazioni legali, materiale tecnico e Ringraziamenti

### 10.1. Informazioni legali

Il programma è Open Source e rilasciato sotto licenza GNU GPLv3

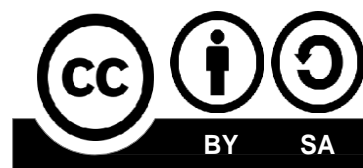
<http://www.gnu.org/licenses/gpl-3.0.txt>



IL SOFTWARE VIENE FORNITO "COSÌ COM'È", SENZA ALCUN TIPO DI GARANZIA, ESPRESSA O IMPLICITA, INCLUSE, MA NON SOLO, LE GARANZIE DI COMMERCIALIZZABILITÀ, IDONEITÀ PER UN PARTICOLARE SCOPO. IN NESSUN CASO PETER DAUSCHER O QUALSIASI ALTRO COLLABORATORE POTRANNO ESSERE RITENUTI RESPONSABILI PER EVENTUALI RICHIESTE, DANNI O ALTRE RESPONSABILITÀ RISULTANTI DAL CONTRATTO, ILLECITO O ALTRO, DERIVANTE DALL'USO COLLEGATO AL SOFTWARE E DI ALTRE OPERAZIONI RELATIVE AL SOFTWARE.

La documentazione stessa è sotto licenza Creative Commons License CC-BY-SA

<http://creativecommons.org/licenses/by-sa/3.0/de/legalcode>



## 10.2. Informazioni tecniche

Per poterlo usare, il programma non ha bisogno di essere installato. Può essere direttamente eseguito da un supporto di memoria di massa.

Il programma è stato sviluppato utilizzando l' IDE Lazarus (versione 0.9.30.4):

<http://www.lazarus.freepascal.org>

Quindi, almeno in teoria, il programma potrebbe essere compilato per qualsiasi sistema operativo in grado di eseguire Lazarus.

Tutti gli elementi grafici sono stati creati con OpenOffice, LibreOffice, Gnu Image Manipulation Program (GIMP) e Inkscape:

<http://de.openoffice.org>

<http://de.libreoffice.org>

<http://www.gimp.org>

<http://www.inkscape.org>

### 10.3. Ringraziamenti

Grazie a tutti coloro che hanno contribuito alla creazione di questo simulatore.

Inoltre, vorrei ringraziare tutte le persone che mi hanno dato consigli utili e segnalato errori:

- I miei studenti del Gymnasium am Kaiserdom , Speyer.
- I miei colleghi Jens Fiedler , Ewald Bickelmann , Bernd Frohlich , Ernst-Lothar Stegmaier e tutti i colleghi che hanno testato il programma con i loro studenti e trovato bug .
- Alexander Gussow e Joachim Brehmer - Moltmann che hanno effettuato test approfonditi sul programma.
- Alexander Domay per la prima compilazione su un sistema Linux .
- Klaus Merkert e Tobias Selinger, Martin Oehler, David Meder-Marouelli per gli utili commenti e l'incoraggiamento a fare di Johnny un progetto Open Source .
- I responsabili delle conferenze imedia e MNU che mi hanno dato la possibilità di presentare Johnny a un pubblico più vasto .
- Soprattutto quelle persone che ho dimenticato nel mio elenco.

Auguro a tutti tanto divertimento e lavoro produttivo con Johnny.

Ogni ulteriore segnalazione di bug o suggerimento sarà apprezzato. Grazie in anticipo!

Peter Dauscher, 10.07.2012

peter.dauscher@gmail.com





**10.4. Note per la lingua italiana:****16.12.2013**

Traduzione:

<b>Istituto:</b>	Tecnico Tecnologico - <u>IIS “Antonio Meucci”</u> - Casarano (Lecce) – Italy
<b>Studenti delle classi:</b>	3BI, 3CI e 3DI settore informatico
<b>Coordinatore e supervisore:</b>	prof. GERVASI Antonio ( <i>antonio.gervasi@gmail.com</i> )
<b>Johnny scout</b>	prof.ssa NICOLACI Graziana
<b>Ideatrice e consulente:</b>	prof.ssa PETRACHI Maria Rosaria
<b>Dirigente Scolastico</b>	prof.ssa AMANTI Concetta

<b>3BI</b>	<b>3CI</b>	<b>3DI</b>
ADAMO NICHOLAS VINCENZO	BAGNATO SIMONE	AMATO MARCO
COLACI ANDREA ANTONIO	BRAMATO MICHAEL	ANCORA MAURO
CORONESE GABRIELE	CASARANO FRANCESCO	ARGIRO' VALENTINO
DE LORENZIS ALESSANDRO	CIULLO CRISTIAN	CALZOLARO CARLO
DI FIGLIA LORENZO	D'AMATO LUCA	CAMPA ANDREA
GIORGINO DAVIDE	DE SALVATORE SAMUELE	CAUSO SIMONE
GIORGINO LUCA	GALATI ANTONIO LORENZO	CIULLO GIORGIO
MASTROLEO MARCO	GRECUCCI RICCARDO	CULIERSI ALBERTO
MELE LORENZO	LATINO MATTEO	D'AMICO LORENZO
MUZZACHI MANUELA	MELE SEBASTIANO	FIUME MARCO
NAPOLI EDOARDO	MERCURIO FRANCESCO	GIARACUNI GIOVANNI
ORLANDINI ANDREA	MILELLI ANDREA	GRAZIOLI FRANCESCO
PANTALEO GIULIO	PALMA GABRIELE LUCIANO	LUCATELLI SIMONE
PERRUCCIO FILIPPO	PINDINELLO ANDREA	LUPO DAVIDE
PICCOLO GIANLUCA	PISANO' FRANCESCO	MANCO ANDREA
PIZZOLANTE MARCO	SANDALO MATTEO	PISANELLO GIANMARCO
POTENZA FEDERICO	SCORDELLA LUCA	RIZZO MARCO
RAINO' MARCO	SCOZZI LUIGI	SABATO LEONARDO
RIA SIMONE	STRADIOTTI LUCA	SOLIDORO STEFANO
ROMANO MATTIA	VINBORSATI SAMUELE	STENDARDO ANTONIO
SPANO LUCA	VINCENTI PARIDE	
	ZOCCO MIRKO	