

# ASSEMBLY

**Fabio Danubbio III C INF**

PROGRAMMARE  
PROGRAMMA  
SOFTWARE  
ALGORITMI



# BREVE STORIA DELLA PROGRAMMAZIONE

0101010101011101001010010011101000100100010101001001011101001001  
10101101001001001010001001001000101001001110100101010101010001010  
101101001001001001110101010011101001001110100100011001010010100101  
10100100101001001011110100101001010010101000100100010010001001000  
0010110111010010010100111001000111010010100010111001101000100101000  
0101010101011101001001001110100100010100010101001001011101001001001  
1010110100100100101000100100100010100100111010010001001001001001001  
101101001001001001110101010011101001001110100100011001001001001001  
10100100100100101111010010100101001010001001000100100010001001000  
00101101110100100101001110010001110100100010111001101000100101000  
01010101010111010010010011101000100100010101001001001011101001001001  
10101101001001001010001001000101001001110100100010010010010010001010  
10111110100100100111010101001110100100111010010001100100100100010100  
10110100100100101111010010100101001010100010010001001000100010001001  
00001011011101001001010011100100011101001000101110011010001001001010  
000101010101110100100100111010001001000101010010010010010111010010010  
1010110100100100101000100100010100100111010010001010101010100010001010  
101101001001001001110101010011101001001110100100011001001001001001001  
101001001001001011110100101001010010100010010001001000100010010001000  
00101101110100100100111001000111010010010100110011101001001001001000  
010111001101000100101000010101010111010010010011101001001110101000100100  
01010100100101110100100100100111010010010010001001000100010010001000  
101011010010010010100010010001010010011101001001010101010100010001010  
101101001001001001110101010011101001001110100100011001001001001001001  
1010010010010010111101001010010101000100100010010001000100010010001000  
001011011101001001001110010001110100100101000101110011010001001001000  
01010101011101001001001110101000100100010101001001001011101001001001  
101011010010010010100010010001010010011101001001010101010100010001010  
101101001001001001110101010011101001001110100100011001001001001001001  
1010010010010010111101001010010101000100100010010001000100010010001000  
001011011101001001001110010001110100100101000101110011010001001001000

# LINGUAGGIO BASSO LIVELLO

MONITOR FOR 6802 1.4

9-14-80 TSC ASSEMBLER PAGE

```

C000          ORG      ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START    LDS      #STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A

0013          RESETA  EQU      %00010011
0011          CTLREG  EQU      %00010001

C003 86 13    INITA   LDA A   #RESETA  RESET ACIA
C005 B7 80 04           STA A   ACIA
C008 86 11    LDA A   #CTLREG  SET 8 BITS AND 2 STOP
C00A B7 80 04           STA A   ACIA

C00D 7E C0 F1    JMP     SIGNON  GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04  INCH    LDA A   ACIA      GET STATUS
C013 47          ASR A   RDRF      SHIFT RDRF FLAG INTO
C014 24 FA    BCC     INCH      RECIEVE NOT READY
C016 B6 80 05  LDA A   ACIA+1   GET CHAR
C019 84 7F    AND A   #$7F    MASK PARITY
C01B 7E C0 79  JMP     OUTCH   ECHO & RTS

*****
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: Digit in acc A
* CALLS: INCH
* DESTROYS: acc A
* Returns to monitor if not HEX input

C01E 8D F0    INHEX   BSR     INCH      GET A CHAR
C020 81 30          CMP A   #'0    ZERO
C022 2B 11          BMI     HEXERR  NOT HEX
C024 81 39          CMP A   #'9    NINE
C026 2F 0A          BLE     HEXRTS  GOOD HEX
C028 81 41          CMP A   #'A    NOT HEX
C02A 2B 09          BMI     HEXERR
C02C 81 46          CMP A   #'F    HEXERR
C02E 2E 05          BGT     HEXERR
C030 80 07          SUB A   #7     FIX A-F
C032 84 0F          HEXRTS  AND A   #$0F    CONVERT ASCII TO DIGI
C034 39          RTS

C035 7E C0 AF    HEXERR  JMP     CTRL      RETURN TO CONTROL LOC

```

# LINGUAGGIO ALTO LIVELLO

```
int main()
{
    char ci; //riprova
    int i, LOL; //LOL > riprova2
    double N=0, q_tot=0, c_tot=0, q_media, c_media, q, c;

    /*** CICLO FINITO SOLO IN CASO SE L'UTENTE DIGITA 'n' ***/
    do
    {
        riprova_2:
        cout<<"Inserisci quantita' di numeri: ";
        cin>>LOL;

        /*** IN CASO SE L'UTENTE INSERISCE UN VALORE MINORE DI 0 ***/
        if(LOL<=0)
        {
            cout<<"Errore, numero minore di 1\n";
            goto riprova_2;
        }

        /*** CALCOLI DI VARI ISTRUZIONI RICHIESTA DALL'UTENTE ***/
        for(i=1; i<=LOL; i++)
        {
            q=quadrato(N);
            q_tot+=totale_quadrato(q);
            c=cubo(N);
            c_tot+=totale_cubo(c);

            cout<<"Numero: " <<i <<endl
                <<"Quadrato: " <<q <<endl
                <<"Cubo: " <<c <<endl
                <<"Totale quadrati: " <<q_tot <<endl
                <<"Totale cubi: " <<c_tot <<endl <<endl;

            N=N+2;
        }

        q_media=q_tot/(i-1);
        cout<<"Media totale quadrato: " <<q_media <<endl;

        c_media=c_tot/(i-1);
        cout<<"Media totale cubo: " <<c_media <<endl;

        /***IN CASO DI ERRORE DI DIGITAZIONE ***/
        riprova:
        cout<<"Riprovare? (s/n)";
        cin>>ci;

        if(!(ci=='s' || ci=='n'))
            goto riprova;
    }
}
```



LINGUAGGIO MEDIO-ALTO LIVELLO

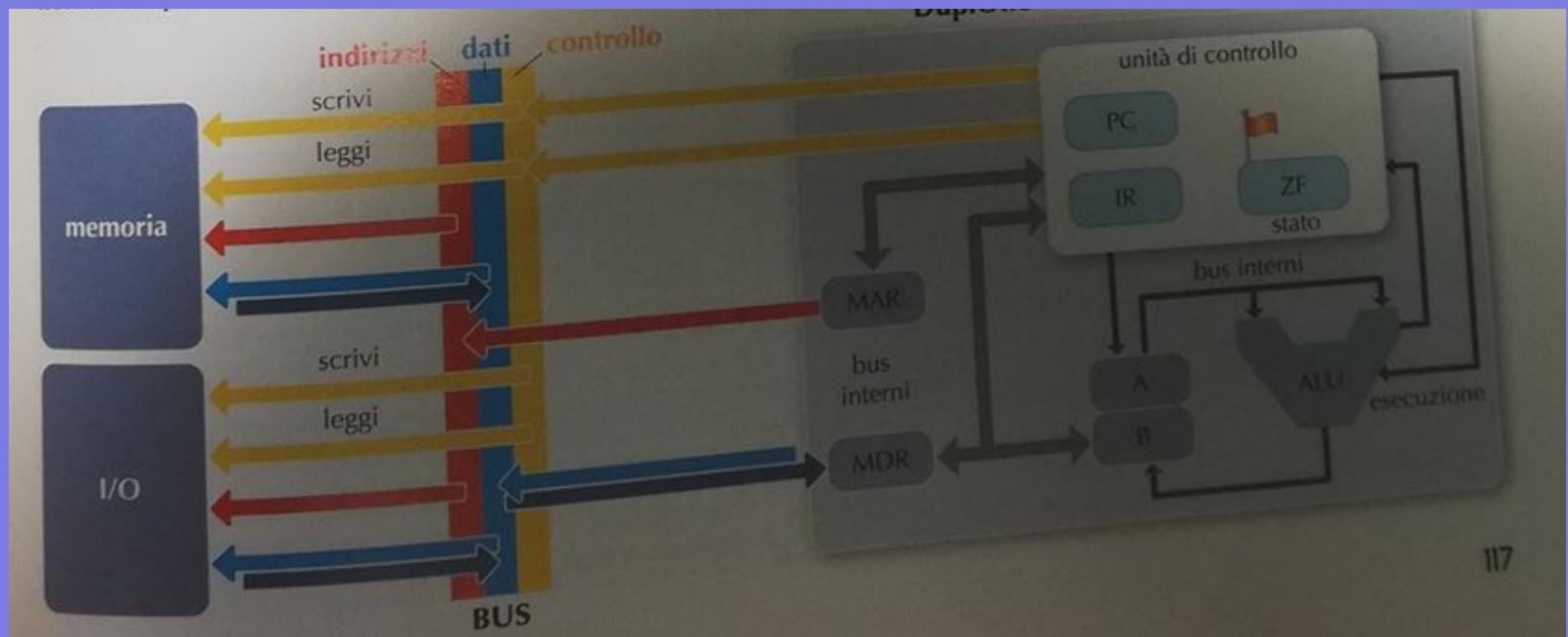
```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML  
2 <html>  
3     <head>  
4         <title>Example</title>  
5         <link href="screen.css" rel="sty  
6     </head>  
7     <body>  
8         <h1>  
9             <a href="/">Header</a>  
10        </h1>  
11        <ul id="nav">  
12            <li>  
13                <a href="one/">One</a>  
14            </li>  
15            <li>  
16                <a href="two/">Two</a>  
17            </li>
```

The PHP logo consists of the letters "php" in a bold, black, sans-serif font. The letters are partially cut off at the top by a dark grey oval shape.The JS logo consists of the letters "JS" in a large, white, sans-serif font. It is set against a solid teal circular background.The word "python" is written in a large, dark grey, lowercase sans-serif font. A small "TM" symbol is positioned to the right of the "n".

# SINTASSI GENERICA ASSEMBLY

**<CODICE OPERATIVO> <OPERANDI>**

# DUPLONE



# CATEGORIE DI ISTRUZIONI

(GENERALI)

ISTRUZIONI ARITMETICHE/LOGICHE

ADD - SUB - DIV - AND - OR - NOT

ECC....

# ISTRUZIONI DI SALTO

## JMP (NC) - JNZ - JZ

ISTRUZIONI DI TRASFERIMENTO  
MOV B [LOCAZIONE]

CALL CONDIZIONATO

CALL NON CONDIZIONATO

METODI DI INDIRIZZAMENTO

INDIRIZZAMENTO A REGISTRO

INDIRIZZAMENTO IMMEDIATO

INDIRIZZAMENTO DIRETTO

INDIRIZZAMENTO INDIRETTO