

A composite image with a purple and pink background. On the left is a stack of white papers, and on the right is a blurred clock face.

Prima esercitazione

A composite image with a green and yellow background. On the left is a stack of white papers, and on the right is a clear yellow clock face.

Prorami e Dati



Programma 01

- Prendendo spunto dagli esempi commentati a lezione, programmare l'Eniac per calcolare il prodotto di due numeri X e Y mettendo il risultato in una variabile Z .
- Si usi l'istruzione "MUL"

Hints 01

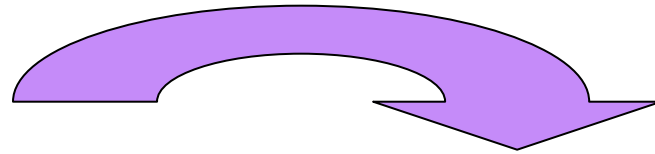
OPCODE E TIPO	SINTASSI	SEMANTICA	ESEMPIO
11001000000	ADD Z	$AX + Z \rightarrow AX$	ADD -10
11001000001	ADD @N	$AX + (N) \rightarrow AX$	ADD @10
11001000100	ADD R	$AX + R \rightarrow AX$	ADD BX
11000000000	SUB Z	$AX - Z \rightarrow AX$	SUB -5
11000000001	SUB @N	$AX - (N) \rightarrow AX$	SUB @10
11000000011	SUB R	$AX - R \rightarrow AX$	SUB BX
11110000000	MUL N	$AX * N \rightarrow BX : AX$	MUL 10
11110000001	MUL @N	$AX * (N) \rightarrow BX : AX$	MUL @10
11110000100	MUL R	$AX * R \rightarrow BX : AX$	MUL BX
11111000000	DIV N	$AX / N \rightarrow AX$	DIV 20
11111000001	DIV @N	$AX / (N) \rightarrow AX$	DIV @10
11111000100	DIV R	$AX / R \rightarrow AX$	DIV BX
11011000001	INC @N	$(N) + 1 \rightarrow (N)$	INC @10
11011000100	INC R	$R + 1 \rightarrow R$	INC BX
11010000001	DEC @N	$(N) - 1 \rightarrow (N)$	DEC @10
11010000100	DEC R	$R - 1 \rightarrow R$	DEC BX
01000001000	NOT	$NOT\{AX\} \rightarrow AX$	NOT
01000000000	NEG	$NOT\{AX\} + 1 \rightarrow AX$	NEG

OPCODE E TIPO	SINTASSI	SEMANTICA	ESEMPIO
10100000000	LD Z	$Z \rightarrow AX$	LD -2
10100000001	LD @N	$(N) \rightarrow AX$	LD @10
10100000100	LD R	$R \rightarrow AX$	LD BX
10111000001	ST @N	$AX \rightarrow (N)$	ST @10
10111000100	ST R	$AX \rightarrow R$	ST BX

OPCODE E TIPO	SINTASSI	SEMANTICA	ESEMPIO
00000001000	NOP	Nessuna operazione.	NOP
00000000000	HLT	Termina l'esecuzione.	HLT
00100000000	JMP N	$N \rightarrow IP$	JMP 10
00101010000	JZ N	$ZE = 1 \Rightarrow N \rightarrow IP$	JZ 10
00101011000	JNZ N	$ZE = 0 \Rightarrow N \rightarrow IP$	JNZ 10

01100001000	IN N	$[N] \rightarrow AX$	IN 10
01100000000	OUT N	$AX \rightarrow [N]$	OUT 10

Hints 02



Non ci serve alcun ciclo, si calcola in AX

0		JMP 5	
1			0
2	w	w	
3	y	y	
4	z	risultato	
5	BX	sp=0	LOAD 0
6			STORE BX
7	CX	ns=y	LOAD @3
8			STORE CX
9		while ns<>0	LOAD CX
10			JZ 16
11		sp=sp+w	LOAD BX
12			ADD @2
13			STORE BX
14		ns=ns-1	DEC CX
15			JMP 9
16		z=sp	LOAD BX
17			STORE @4
18			HLT

0		JMP 5	
1			0
2	x	x	
3	y	y	
4	z	risultato	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			HLT

Hints 03

0		JMP 5	
1			0
2	x	x	
3	y	y	
4	z	risultato	
5	AX <-- x	LOAD @2	
6	AX <-- AX * y	MUL @3	
7	z <-- AX	STORE @4	
8	end	HLT	



Programma 02

- Modificare l'esempio precedente calcolando la seguente espressione
- $(x+y) * w \rightarrow z$

0		JMP 5
1	w	w
2	x	x
3	y	y
4	z	risultato
5	AX <-- x	LOAD @2
6	AX <-- AX+y	ADD @3
7	AX <-- AX * w	MUL @1
8	z <-- AX	STORE @4
9	end	HLT



Programma 02 bis

- Modificare l'esempio precedente calcolando la seguente espressione
- $X * (y + w) \rightarrow z$
- Di quanto differisce il programma?



Programma 03

- Modificare l'esempio precedente calcolando la seguente espressione
- $(x+y) * (w+u) \rightarrow z$



Programma 04

- Modificare l'esempio precedente calcolando la seguente espressione
- $X*y+u*w \rightarrow z$



Immagini

BitMap



La struttura dati della BITMAP

- Si ripeta il procedimento per manipolare i colori di una bitmap data.
- Si usi l'editor esadecimale XVI32 e la figura Bitmap.bmp