



# UNIVERSITÀ DEGLI STUDI DI MILANO

## CORSI DI LAUREA TRIENNALI IN INFORMATICA E SICUREZZA DEI SISTEMI E DELLE RETI INFORMATICHE (CREMA)

ARCHITETTURA DEGLI ELABORATORI (N.O. – codici F1A04- F680D-)

ARCHITETTURA DEGLI ELABORATORI II (N.O. – codice F1X1I-)

ARCHITETTURA DEGLI ELABORATORI I (V.O. – codice F6705-)

prof. Nello SCARABOTTOLO — anno accademico 2019/2020 — 4 febbraio 2020

PROVA PRATICA DI PROGRAMMAZIONE IN LINGUAGGIO ASSEMBLY LC-2 – tema **1**

### NOTE PER LO SVOLGIMENTO

I programmi **LC2Edit** e **Simulate** sono accessibili dal menu **Start/Tutti i programmi/Architettura**. L'elaborato deve essere **salvato sul desktop** della postazione di lavoro con la seguente denominazione: **cognome-numeromatricola.asm**

L'elaborato deve contenere soltanto il sottoprogramma da realizzare (denominato come indicato nelle specifiche) e le eventuali variabili o costanti di memoria necessarie; ogni ulteriore elemento (per es., istruzioni necessarie a collaudare il sottoprogramma, o pseudo-istruzioni di definizione di dove caricare in memoria il sottoprogramma) **deve essere rimosso dalla versione finale dell'elaborato**.

**È vietato consultare testi o appunti di qualsiasi genere, come pure navigare in rete, pena l'annullamento della prova.**

**ATTENZIONE: EVITARE ASSOLUTAMENTE DI CHIUDERE LA SESSIONE DI LAVORO (logoff)**

### SPECIFICHE DEL SOTTOPROGRAMMA DA REALIZZARE

Il candidato scriva un sottoprogramma denominato **SOMMA\_ARRAY** che riceve:

- nei registri **R0** e **R1** gli indirizzi dei primi elementi di due array **A0** e **A1** di numeri a 16 bit in complemento a due; i due array hanno uguale lunghezza e il valore zero è il "tappo" finale per entrambi;
- nel registro **R2** l'indirizzo di inizio di una zona di memoria libera destinata a contenere l'array **A2**.

Il sottoprogramma deve:

- assegnare a ogni elemento di posto **i** di **A2** il valore **A2(i) = A0(i) + A1(i)**;
- restituire in **R0** il numero di traboccamenti verificatisi, non importa se positivi o negativi.

Qualora per la realizzazione del sottoprogramma fosse necessario utilizzare altri registri della CPU, il sottoprogramma stesso deve restituire il controllo al programma chiamante senza che tali registri risultino alterati.

### ESEMPIO DI FUNZIONAMENTO DEL SOTTOPROGRAMMA

Input		A0		A1		A2	
R0	x3408	x3408	32760	x340D	8	x3412	-
R1	x340D	x3409	7	x340E	17		-
R2	x3412	x340A	-32760	x340F	-9		-
		x340B	8	x3410	-9		-
		x340C	0	x3411	0		-
Output		A0		A1		A2	
R0	2	x3408	32760	x340D	8	x3412	-32768
R1	-	x3409	7	x340E	17		24
R2	-	x340A	-32760	x340F	-9		32767
		x340B	8	x3410	-9		-1
		x340C	0	x3411	0		

### LINGUAGGIO ASSEMBLY LC-2

ADD	DR,SR1,SR2	LDI	DR,pgoffset9	SR	= Source Register
ADD	DR,SR,imm5	LDR	DR,BaseR,index6	DR	= Destination Register
AND	DR,SR1,SR2	LEA	DR,pgoffset9	BaseR	= Base Register
AND	DR,SR,imm5	NOT	DR,SR	imm5	= valore numerico, di 5 bit
BR	nzp,pgoffset9	RET		pgoffset9	= spiazamento nella pagina, di 9 bit
JMP	pgoffset9	RTI		index6	= indice rispetto a BaseR, di 6 bit
JMPR	BaseR,index6	ST	SR,pgoffset9	trapvect8	= vettore di interrupt, di 8 bit
JSR	pgoffset9	STI	SR,pgoffset9		
JSRR	BaseR,index6	STR	SR,BaseR,index6		
LD	DR,pgoffset9	TRAP	trapvect8		