



# 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 — 14 gennaio 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 (logout)**

### SPECIFICHE DEL SOTTOPROGRAMMA DA REALIZZARE

Il candidato scriva un sottoprogramma denominato **CONTA\_OVER** che riceve:

- nel registro **R0** l'indirizzo della prima cella di una zona di memoria contenente una sequenza di numeri **n(i)** a 16 bit in complemento a due;
- nel registro **R1** l'indirizzo della cella contenente l'ultimo numero della sequenza di cui al punto 1.
- nel registro **R2** un **numero positivo N** a 16 bit in complemento a due.

Il sottoprogramma deve sostituire nella sequenza ogni numero **n(i)** con il risultato della somma **n(i)+N** e restituire in **R2** il conteggio delle somme che hanno generato traboccamento positivo (**overflow**).

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.

### ESEMPI DI FUNZIONAMENTO DEL SOTTOPROGRAMMA

#### Input

R0	X3408	x3408	-56
R1	X340C	x3409	7
R2	32760	x340A	0
		x340B	8
		x340C	10

#### Input

R0	X3408	x3408	-56
R1	X340C	x3409	7
R2	2	x340A	0
		x340B	8
		x340C	10

#### Output

R0	-	x3408	32704
R1	-	x3409	32767
R2	2	x340A	32760
		x340B	-32768
		x340C	-32766

#### Output

R0	-	x3408	-54
R1	-	x3409	9
R2	0	x340A	2
		x340B	10
		x340C	12

### 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	= spiazzamento 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		