

## SPECIFICHE DEL SOTTOPROGRAMMA DA REALIZZARE

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

1. nel registro **R0** l'indirizzo della prima cella di una zona di memoria contenente una sequenza di stringhe di 16 bit ciascuna; la stringa costituita da tutti zeri è il terminatore della sequenza e non va considerata;
2. nel registro **R1** una stringa di 16 bit.

Il sottoprogramma deve sostituire a ogni stringa dell'array l'OR (somma logica) tra la stringa presente nell'array e la stringa ricevuta in **R1**. Si ricorda che per il teorema di De Morgan:

$$\overline{a \text{ OR } b} = \overline{a} \text{ AND } \overline{b}$$

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

R0	X3408	x3408	0000 1111 0000 1111
R1	1111 0000 1111 0000	x3409	0011 0011 0011 0011
		x340A	1100 1100 1100 1100
		x340B	0000 0000 0000 0000

### Output

R0		x3408	1111 1111 1111 1111
R1		x3409	1111 0011 1111 0011
		x340A	1111 1100 1111 1100
		x340B	0000 0000 0000 0000

## 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		