

<b>COGNOME:</b>		<b>NOME</b>		<b>A1</b>
<b>MATRICOLA:</b>				
<b>DOCENTE:</b>				

Domanda 1	Risultato
<p>Si convertano i seguenti numeri da decimale a binario in complemento a 2 su 8 bit:</p> <p><math>21_{10} = X_{ca2}</math></p> <p><math>-100_{10} = Y_{ca2}</math></p>	<p>X = _____</p> <p>Y = _____</p>
<p>Si riportino TUTTI i passaggi</p>	

Domanda 2
<p>Determinare se la seguente uguaglianza Booleana è vera: <math>(X \text{ AND } Y) \text{ OR } (Y \text{ AND } Z) \text{ OR } (\text{NOT } X \text{ AND } Z) = (X \text{ AND } Y) \text{ OR } (\text{NOT } X \text{ AND } Z)</math></p>

Domanda 3
<p>Descrivere brevemente il ruolo del program counter nell'esecuzione di un programma</p>

**Domanda 4 (PROGRAMMAZIONE)**

Nell'elaborazione digitale delle immagini, un kernel è una piccola matrice, solitamente di dimensione 3x3, utilizzata per applicare dei filtri a un'immagine quali sfocatura, messa a fuoco, estrazione dei contorni e altro ancora. Ciò si ottiene mediante l'operazione di convoluzione tra il kernel e l'immagine.

L'operazione di convoluzione prende in ingresso due matrici di numeri interi. La prima è l'immagine da filtrare, la seconda il kernel che definisce il tipo di filtro. Il filtro esamina ogni pixel dell'immagine e il risultato del filtraggio si ottiene in questo modo. Per prima cosa si "sovrappone" il kernel alla matrice immagine in modo che il centro del kernel sia in corrispondenza del pixel da elaborare. Quindi, il valore del pixel filtrato si ottiene come la somma dei prodotti di ciascun elemento della matrice kernel con il corrispondente pixel della matrice immagine sottostante. Per i pixel sul bordo dell'immagine, dal momento che non tutti gli 8 pixel vicini sono effettivamente contenuti nell'immagine, si assume che i pixel mancanti abbiano valore 0. Quindi, dato un pixel di coordinate (x,y), se le coordinate di uno dei suoi vicini cascano fuori dall'immagine si considera il valore 0 nella somma (si vedano gli esempi per chiarimenti).

Si scriva un programma C in grado di realizzare l'operazione di convoluzione tra un'immagine e un kernel. L'immagine è codificata in un file sotto forma di una matrice di dimensione 1024x768 pixel. Ogni pixel è rappresentato da un numero intero. I numeri su ogni riga sono separati da uno spazio. La matrice kernel è invece una matrice di dimensione 3x3 di numeri interi memorizzata su un secondo file. Anche in questo caso i numeri su una riga sono separati da uno spazio. Si assuma che il formato di tutti i file sia corretto.

Il programma riceve i nomi dei due file da linea di comando (il primo è il file dell'immagine, il secondo quello del filtro) e deve salvare l'immagine filtrata su un file chiamato `risultato.txt` con lo stesso formato del file di ingresso

Il seguente esempio mostra l'operazione di convoluzione per due pixel uno centrale e uno posizionato sul bordo nel caso di un'immagine di dimensione 10x7.

Matrice immagine

10	2	20	30	0	0	1	10	20	10
1	40	-1	27	33	41	444	98	900	79
78	6	54	87	0	67	0	67	87	78
67	78	87	95	98	61	98	67	98	67
98	67	89	63	98	65	89	67	89	654
	6	7	8	9	5	7	9	547	84
	234	54	6	867	789	54	34	54	67

Kernel

1	0	-1
0	0	0
-1	0	1

Pixel filtrato =  $87*1+0*0+67*-1+95*0+98*0+51*0+63*-1+98*0+65*1= 22$

Pixel filtrato =  $0*1+6*0+7*-1+0*0+234*0+54*0+0*-1+0*0+0*1= -7$

<b>COGNOME:</b>		<b>NOME</b>		<b>A2</b>
<b>MATRICOLA</b>				
<b>DOCENTE:</b>				

Domanda 1	Risultato
<p>Si convertano i seguenti numeri da decimale a binario in modulo e segno su 8 bit:</p> <p><math>24_{10} = X_{MS}</math></p> <p><math>-110_{10} = Y_{MS}</math></p>	<p>X = _____</p> <p>Y = _____</p>
<p>Si riportino TUTTI i passaggi</p>	

Domanda 2
<p>Determinare se la seguente uguaglianza Booleana è vera: <math>\text{NOT} ((X \text{ AND } Y) \text{ OR } (Y \text{ AND } Z) \text{ OR } (\text{NOT } X \text{ AND } Z)) = \text{NOT} ((X \text{ AND } Y) \text{ OR } (\text{NOT } X \text{ AND } Z))</math></p>

Domanda 3
<p>Descrivere brevemente cosa è la ALU in un microprocessore.</p>

**Domanda 4 (PROGRAMMAZIONE)**

Nell'elaborazione digitale delle immagini, un kernel è una piccola matrice, solitamente di dimensione 3x3, utilizzata per applicare dei filtri a un'immagine quali sfocatura, messa a fuoco, estrazione dei contorni e altro ancora. Ciò si ottiene mediante l'operazione di convoluzione tra il kernel e l'immagine.

L'operazione di convoluzione prende in ingresso due matrici di numeri interi. La prima è l'immagine da filtrare, la seconda il kernel che definisce il tipo di filtro. Il filtro esamina ogni pixel dell'immagine e il risultato del filtraggio si ottiene in questo modo. Per prima cosa si "sovrappone" il kernel alla matrice immagine in modo che il centro del kernel sia in corrispondenza del pixel da elaborare. Quindi, il valore del pixel filtrato si ottiene come la somma dei prodotti di ciascun elemento della matrice kernel con il corrispondente pixel della matrice immagine sottostante. Per i pixel sul bordo dell'immagine, dal momento che non tutti i 24 pixel vicini sono effettivamente contenuti nell'immagine, si assume che i pixel mancanti abbiano valore 255. Quindi, dato un pixel di coordinate (x,y), se le coordinate di uno dei suoi vicini cascano fuori dall'immagine si considera il valore 255 nella somma (si vedano gli esempi per chiarimenti).

Si scriva un programma C in grado di realizzare l'operazione di convoluzione tra un'immagine e un kernel. L'immagine è codificata in un file sotto forma di una matrice di dimensione 800x600 pixel. Ogni pixel è rappresentato da un numero intero. I numeri su ogni riga sono separati da uno spazio. La matrice kernel è invece una matrice di dimensione 3x3 di numeri interi memorizzata su un secondo file. Anche in questo caso i numeri su una riga sono separati da uno spazio. Si assuma che il formato di tutti i file sia corretto.

Il programma riceve i nomi dei due file da linea di comando (il primo è il file dell'immagine, il secondo quello del filtro) e deve salvare l'immagine filtrata su un file chiamato `risultato.txt` con lo stesso formato del file di ingresso

Il seguente esempio mostra l'operazione di convoluzione per due pixel uno centrale e uno posizionato sul bordo nel caso di un'immagine di dimensione 10x7.

Matrice immagine

10	2	20	30	0	0	1	10	20	10
1	40	-1	27	33	41	444	98	900	79
78	6	54	87	0	67	0	67	87	78
67	78	87	95	98	61	98	67	98	67
98	67	89	63	98	65	89	67	89	654
6	7	8	9	5	7	9	547	84	54
234	54	6	867	789	54	34	54	67	8

Kernel

1	0	-1
0	0	0
-1	0	1

Pixel filtrato =  $87 \cdot 1 + 0 \cdot 0 + 67 \cdot -1 + 95 \cdot 0 + 98 \cdot 0 + 51 \cdot 0 + 63 \cdot -1 + 98 \cdot 0 + 65 \cdot 1 = 22$

Pixel filtrato =  $255 \cdot 1 + 6 \cdot 0 + 7 \cdot -1 + 255 \cdot 0 + 234 \cdot 0 + 54 \cdot 0 + 255 \cdot -1 + 255 \cdot 0 + 255 \cdot 1 = 248$