NOME **COGNOME** **DATA**

ESERCIZI DA SVOLGERE A CASA

1) Convertire in binario in numeri -98 e -57 e -110 (da base DIECI a base DUE)

-57	
57:2	1
28:2	0
14:2	0
7:2	1
3:2	1
1	

57 = 111001

Per ottenere il negativo uso il **complemento a 2**: aggiungo il bit di segno, inverto tutti i bit e aggiungo 1

0111001 → **1000110 +**
1 =

1000111

RISULTATO: **1000111 (il primo a sinistra è il bit di segno)**

2) Convertire in decimale i seguenti numeri binari **con segno :** 100101 11011 010011

100101 è un numero NEGATIVO (il primo bit è 1). Devo fare il comlemento a 2.

100101 → 011010 +
1 =

011011 → Questo è l'opposto del numero di partenza, ovvero un numero positivo. Covertendo in decimale:
16 + 8 + 2 + 1 = 27

RISULTATO: il numreo binario con segno 100101 = - 27

3) Eseguire la somma dei seguenti numeri binari **con segno** 010100 + 111001

Il primo è un numero positivo, il secondo negativo. Il sommo così come sono

11
010100 +
111001 =

1001101

In rosso i bit di segno. Il bit dopo il bit di segno viene scartato

RISULTATO: **001101** è un numero positivo e si converte normalmente: **8 + 4 + 1 = 13**
Volendo fare una controprova si possono convertire i due numeri da sommare. Il primo (positivo) è **20**. Il secondo (negativo) è **-7**

1) Cos'è un pixel?

E' l'unità **minima** di cui è composta un'immagine digitale. Solitamente è quadrata (ma potrebbe essere rettangolare).
Il pixel può avere un solo colore (non posso dividere un pixel in due) è viene rappresentato, all'interno del computer, con un certo **numero di bit** (a seconda di **quanti colori** sono ammessi nell'immagine)

2) Cos'è la risoluzione? Come viene espressa?

La risoluzione, espressa in DPI (Dot Per Inch – punti per pollice) è il **numero di pixel** in cui suddivide un **centimetro dell'immagine**. Negli esempi del corso di Tecnologie Informatiche utilizziamo i cm al posto dei pollici.

3) Spiega in poche parole cos'è la codifica BITMAP

BITMAP è una tecnica per codificare in binario un'immagine.

- Divido l'immagine in pixel
- Ad ogni pixel associo un **certo numero di bit a seconda di quanti colori voglio assegnare** ad ogni pixel. Ad esempio:

1 bit per pixel: il pixel può essere **bianco o nero**
8 bit per pixel: posso fare **256 colori**. Di solito con 8 bit si preferisce fare **256 *livelli di grigio***
3 byte per pixel. Uso ogni byte per rappresentare 256 livelli di ognuo dei colori fondamentali RGB (in totale **16 milioni di colori**

4) Spiega cos'è RGB (fai anche un esempio)

E' una tecnica per codificare i colori. Utilizzo 1 **byte (8 bit) per ognuno dei tre colori** fondamentali (su schermo): **Rosso**, **Verde** e **Blu** (Red Green Blue).
Per ognuno dei colori posso fare 256 livelli (dal nero -0- al colore puro e brillante -255-).
Mischiando i 256 livelli di ognuno dei tre colori principali posso ottenere

256*256*256 = 16 milioni di colori

Per fare il **viola**, ad esempio assegno:
R=255 G=0 B=255 → in binario 11111111 00000000 11111111
Per fare i grigi, ad esempio, devo usare lo stesso valore per tutti e tre le componenti. Più è alto il valore più il **grigio** sarà chiaro. Ad esempio un grigio scuro sarà
R= 70 G= 70 B= 70 → in binario 01000110 01000110 01000110

5) Quanti sono i colori possibili se uso 8 bit per pixel? Di solito quando si usa questa codifica?

Con 8 bit posso codificare 256 colori (come nelle GIF) ma, solitamente, si preferisce codificare 256 **livelli di grigio**, per le immagini in bianco e nero (**scala di grigio**).

6) **Quanto bit mi servono per codificare 57 colori? Perché?**

Per codificare 57 colori ho bisogno di **6 bit.**
Infatti con 5 bit posso fare al massimo 2^5 = 32 codifiche (in questo caso colori)
Con 6 bit posso fare fin a 2^6 = 64 codifiche (colori)

7) Quanti bit servono per codificare in scala di grigio un immagine in formato full HD?

Il formato HD ha dimensioni (in pixel) 1280 x 720 px
Il numero di pixel totale è 1280 * 720 = **921.600 pixel**
Per codificare in scala di grigio completa servono 8 bit (1 byte) per poter fare 256 livelli di grigio
Quindi possiamo calcolare direttamente il numero di byte necessari assegnando **1 byte ad ogni pixel**
921.600 * 1 = 921.600 byte = (circa) **921 Kbyte**

8) Quanto spazio in memoria (byte) occupa un'immagine di 50x30cm in alta risoluzione e codificata con 90 colori?

Per capire quanti pixel ci sono nell'immagine devo utilizzare la risoluzione in **DPI** ovvero il **numero di pixel per ogni centimetro**
L'alta risoluzione prevede **300 DPI**, quindi avrò 300 pixel per ogni centimetro
Ora posso calcolare quanti pixel misura l'immagine
In orizzontale: 50cm x 300 dpi = 15000 pixel
In verticale: 30cm x 300 dpi = 9000 pixel
Il numero di pixel totali è 15000 * 9000 = **135.000.000 pixel** (135 milioni)

Per poter codificare 90 colori mi servono 7 bit.
Infatti 6 bit posso codificare 2^6 = 64 livelli (non bastano)
Con 7 bit posso codificare fino a 2^7 = 128 livelli (di colore)

Per trovare le dimensioni (in byte) dell'immagine moltiplico i bit usati per ogni pixel (7) per il numero di pixel:
135.000.000 * 7 = **945.000.000 bit**
Per trovare il numero di byte divido per 8
945.000.000 / 8 = **118.125.000 byte** = (circa) **118 Mbyte**

9) Quante immagini in formato BITMAP HD con 16 milioni di colori possono essere contenute in una scheda SD da 2GB?

Per prima cosa calcoliamo quanto pesa (in byte) un'immagine così come descritta dal problema.
L'immagine è in formato HD quindi 1280 x 720 pixel
I pixel totali sono quindi 1280*720 = 921600 px

Il numero di bit per ogni pixel è legato al numero di colori che vogliamo utilizzare. 16 Milioni è il numero di colori che possiamo realizzare con la codifica RGB che utilizza 3 byte per ogni pixel (un byte per ogni componente del colore: rosso, verde e blu)

Quindi con 3 byte per pixel posso trovare il numero totale di byte
921600 * 3 **= 2.764.800 byte** (circa 2,7 Mbyte)

Quante immagini grandi 2,7 MB possono essere contenute in una scheda da 2Gb
Dobbiamo dividere la dimensione della scheda per la dimensione delle foto. Ricordiamoci che
2G = 2000M
Quindi
2000 / 2,7 = circa 740 foto!

10) Una immagine fullHD alta 20cm quanto è larga?

L'immagine in fullHD misura 1920 x 1080 pixel
Sapendo l'altezza possiamo trovare la risoluzione in **DPI = PX/CM**
Utilizziamo la misura dell'altezza

Risoluzione = 1080 / 20 = 54 DPI

Ora possiamo utilizzare la risoluzione in DPI per trovare la larghezza dell'immagine: CM = PX/DPI
Larghezza:
1920 / 54 = 35,5 cm

11) Come funziona il codice ASCII?

Il codice ASCII è una tabella che associa ad ogni carattere un byte.
Utilizzando 1 solo byte posso codificare **2^8 = 256 caratteri.**
Quindi in questo codice ad ogni byte corrisponde un carattere, tra i quali anche simboli speciali, parentesi, punteggiatura etc)
Essendo questi caratteri troppo pochi per codificare le lettere di tutti gli alfabeti del mondo è stato introdotto un nuovo codice chiamato **UNICODE** che utilizza da 1 a 4 byte per ogni carattere.

12) Quale di queste parole può essere rappresentata in ASCII da questa sequenza di bit? Perché? (ATTENZIONE: non usare una tabella ASCII ma usare il ragionamento)

101100100011010010110010

Possibili risposta (scegliere quella giusta)

AMA – CERA – SEI – NOVE – ALT

Innanzi tutto dobbiamo dividere il codice in byte (8bit): 10110010 00110100 10110010
Sicuramente questo è un codice di una parola di 3 lettere (3byte). Escludiamo quindi CERA e novembre.
Ci accorgiamo che **il primo è il terzo byte sono uguali**! Quindi rappresentano lo **stesso carattere.**
L'unica risposta ammissibile è quindi **AMA**