

Fondamenti di Informatica - A.A. 2023-2024

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 20/01/2024



Cognome	Nome	Matricola o Cod. Persona
---------	------	--------------------------

INIZIARE LA SOLUZIONE DI OGNI
ESERCIZIO SU UNA PAGINA NUOVA

Quesito:	1	2	3	4	5	Totale
Valutazione massima (in /30):	4	7	6	7	6	30
Valutazione quesito in decimi (/10):						

RESTITUIRE COMPILATO ANCHE
NEL CASO IN CUI CI SI RITIRA

Istruzioni:

- gli esercizi devono essere risolti utilizzando il C ANSI 89, in linea con quanto fatto durante il corso;
- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del rosso.
- tempo a disposizione: 1h 45m

Stile del codice C:

- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è possibile utilizzare sottoprogrammi di libreria.

Sapevo già programmare:

☐ No, non è vero ☐ in C ☐ in C++/C# ☐ in Python ☐ in Java ☐ in PHP/Javascript ☐ in VB* ☐ in altro linguaggio

Quesito 1 [4 pts]

Dati i due valori $X = -ABBA_{16MS}$ e $Y = 10101010001000001_{2MS}$ effettuare la conversione in base 2, notazione complemento a 2 (2C2), di ognuno degli operandi sul numero minimo di bit necessari. Si effettuino quindi le operazioni $X+Y$ e $X-Y$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. **Mostrare i passaggi fatti e motivare la risposta relativa all'overflow.**

Riportare nello spazio sottostante la codifica di X_{2C2} , Y_{2C2} e i risultati finali delle operazioni (tutti i passaggi devono essere sui fogli di protocollo), utilizzando solo le caselle necessarie (**allineati a destra**) ed indicando se si è verificato overflow (segnare la casella corrispondente).

X_{2C2}	<input type="text"/>	Y_{2C2}	<input type="text"/>
$X+Y_{2C2}$	<input type="text"/>	$X-Y_{2C2}$	<input type="text"/>
	<input type="checkbox"/> OVF		<input type="checkbox"/> OVF

Quesito 2 [7 pts]

- (5 pts) Scrivere un sottoprogramma `minmaxchar` che riceve come parametro una stringa `vocabolo` costituita da caratteri alfabetici maiuscoli e minuscoli. Il sottoprogramma **trasmette** al chiamante i caratteri alfabetici più piccolo e più grandi in ordine alfabetico, indipendentemente dal fatto che siano minuscoli o maiuscoli. Alcuni esempi:
- ```
ingresso uscita
Edera a r
ZECCA A Z
Testi e T (oppure t)
\0 \0 \0
```
- (2 pts) Scrivere un programma che acquisisce da riga di comando una stringa e avvalendosi del sottoprogramma `minmaxchar` visualizza i caratteri alfabetici più piccolo e più grandi in essa contenuta.

### Quesito 3 [6 pti]

Per impostare una combinazione per un lucchetto numerico vengono indicate le seguenti regole:

- deve essere un numero di 6 cifre
- ci devono essere due cifre adiacenti uguali
- da sinistra verso destra le cifre non sono mai decrescenti: o crescono o rimangono uguali (come in 111123 oppure 135679)

A titolo di esempio, sono vere le seguenti affermazioni:

111111 rispetta le regole (doppio 1 e mai decrescente)  
223450 non rispetta le regole (cifre decrescenti 50)  
123789 non rispetta le regole (non ci sono due cifre adiacenti uguali)

Scrivere un sottoprogramma `verifica` che ricevuto in ingresso un valore intero positivo restituisce 1 se è una combinazione valida, 0 altrimenti.

### Quesito 4 [7 pti]

Per rappresentare una immagine vengono utilizzate delle sequenze di valori interi, ciascuno riferito ad un elemento dell'immagine. Ogni immagine è costituita da uno o più strati. Per esempio, data una immagine di dimensione  $3 \times 2$  pixel, la sequenza di valori 1 2 3 4 5 6 7 8 9 0 1 2 corrisponde a due strati come di seguito riportati:

strato 1: 1 2 3  
          4 5 6

strato 2: 7 8 9  
          0 1 2

Scrivere un sottoprogramma che ricevuto in ingresso un array monodimensionale di interi `valori` che contiene **tutti e soli** i valori che descrivono una immagine, le dimensioni dell'immagine in termini di `righe` e `colonne` ed il numero di `strati`, restituisce il numero di 1 presente nello strato che contiene meno 0. Per esempio, se si riceve l'informazione relativa all'immagine di seguito riportata, il sottoprogramma restituisce 4 (il numero di 1 presenti nello strato 3, che è quello che contiene meno 0):

strato 1: 2 2 0 2 2 2  
          2 1 0 2 0 2  
          2 2 2 0 2 2  
          2 2 2 2 2 2  
strato 2: 2 2 0 1 2 2  
          2 2 2 2 2 2  
          1 0 2 2 2 2  
          0 2 1 0 2 0  
strato 3: 2 2 0 2 2 2  
          2 2 1 2 2 2  
          1 2 2 2 2 2  
          2 1 2 2 1 2  
strato 4: 1 2 2 0 2 2  
          2 2 2 2 2 2  
          0 2 2 2 1 2  
          1 2 2 2 2 2  
strato 5: 1 2 0 2 0 2  
          2 2 2 0 2 2  
          2 2 2 2 0 2  
          2 2 2 2 2 2

Nel caso ci siano più strati con ugual numero minimo di 0, se ne scelga uno in modo arbitrario.

### Quesito 5 [6 pti]

(1 pto) Definire un nuovo tipo di dato per la gestione di elementi di una lista che rappresenta punti nello spazio piano, individuati da due coordinate intere.

(5 pti) Scrivere un sottoprogramma che ricevuta in ingresso una lista per la gestione di punti nello spazio piano (usando il tipo di dato prima definito) calcola e restituisce al chiamante la lunghezza del percorso che si ottiene collegando i punti presenti nella lista. Come distanza tra due punti si utilizzi la distanza di Manhattan o distanza del taxi, ossia la somma del valore assoluto delle differenze delle loro coordinate (quindi la distanza tra  $(x_1, y_1)$  e  $(x_2, y_2) = |x_2 - x_1| + |y_2 - y_1|$ ).

Esempio:

ingresso: (1, 3) -> (4, 3) -> (4, 9) -> (-4, 9) -> (-5, -3) -> (2, 1) -> |  
uscita: 41

