

Fondamenti di Informatica - A.A. 2020-2021

Scuola di Ingegneria Industriale e dell'Informazione
 Prof.ssa Cristiana Bolchini
 Appello del 14/01/2021



POLITECNICO
 MILANO 1863

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

Quesito:	1	2	3	4	Totale
Valutazione massima:	6	6	7	11	30
Valutazione in decimi (/10):					

Istruzioni:

- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- tempo a disposizione: 1h 15m

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.

INIZIARE LA SOLUZIONE DI OGNI ESERCIZIO SU UNA PAGINA NUOVA

RESTITUIRE COMPILATO ANCHE NEL CASO IN CUI CI SI RITIRA

Quesito 1 [6 pts]

(5 pts) Dati i due valori $X = -52_{10MS}$ e $Y = -52_{16MS}$ 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.

Indicare quale tra le 4 risposte è corretta.

A

X_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/>	ovf	Y_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>
$(X + Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/>	<input checked="" type="checkbox"/> X	$(X - Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>

B

X_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/>	ovf	Y_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>
$(X + Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/>	<input checked="" type="checkbox"/> X	$(X - Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>

C

X_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/>	ovf	Y_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>
$(X + Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/>	<input type="checkbox"/>	$(X - Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>

D

X_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="0"/>	ovf	Y_{2C2} :	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>
$(X + Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="0"/>	<input checked="" type="checkbox"/> X	$(X - Y)_{2C2}$:	<input type="text" value=""/> <input type="text" value=""/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="0"/>

(1 pts) Con riferimento alla notazione IEEE 754 per la rappresentazione dei numeri reali nel sistema binario, il valore 11.3 verrà rappresentato esattamente. Vero o falso?

Quesito 2 [6 pts]

Scrivere un sottoprogramma che ricevuto in ingresso un array di valori interi e qualsiasi altro parametro ritenuto strettamente necessario trasmette al chiamante i) il primo quartile q_1 , ii) la mediana (o secondo quartile, q_2), iii) il terzo quartile q_3 . I quartili si ottengono dividendo l'insieme di dati *ordinati* in 4 parti uguali ed esattamente: a) il primo quartile è il valore che lascia alla sua sinistra il 25% degli elementi; b) il secondo quartile coincide con la mediana dato che è quello che lascia alla sua sinistra il 50% dei dati, c) il terzo quartile è il valore che lascia il 75% degli elementi a sinistra e il 25% a destra.

Quesito 3 [7 pts]

(5 pts) Scrivere un sottoprogramma che riceve in ingresso una lista per la gestione dei numeri interi la compatta, facendo in modo che alla fine la lista non contenga valori replicati o nulli. Il programma chiamante dovrà ovviamente accedere al termine dell'esecuzione del sottoprogramma, alla lista compattata.

Se la lista iniziale è la seguente

1 → 3 → 5 → -2 → 3 → -1 → 0 → 2 → 4 → 1

al termine dell'esecuzione dovrà essere:

1 → 3 → 5 → -2 → -1 → 2 → 4

(1 pt) Definire un tipo di dato opportuno per gli elementi della lista.

(1 pt) Completare con la chiamata al sottoprogramma

```
list_t * head = NULL;
list_t * p;
/* chiamata */
...
/* visualizza */
for(p = head; p; p = p->next)
    printf("%d ->" p->info);
```

Si considerino già disponibili e non da sviluppare i sottoprogrammi seguenti:

```
/* inserisce in testa alla lista */
elem_t * push(elem_t *, int);
/* inserisce in coda alla lista */
elem_t * append(elem_t *, int);
/* inserisce un elemento nella lista in ordine crescente */
elem_t * insert_inc(elem_t *, int);
/* inserisce un elemento nella lista in ordine decrescente */
elem_t * insert_dec(elem_t *, int);
/* elimina dalla lista il primo elemento */
elem_t * pop(elem_t *);
/* elimina dalla lista tutti gli elementi con il valore indicato */
elem_t * delete(elem_t *, int);
/* restituisce il riferimento all'elemento nella lista che ha il valore indicato, se esiste, NULL altrimenti */
elem_t * exists(elem_t *, int);
/* restituisce il numero di elementi nella lista */
int length(elem_t *);
```

Quesito 4 [11 pts]

La cerniera è uno schema enigmistico che segue una formula del tipo $ZX / YZ = XY$, ossia date due stringhe s_1 e s_2 cerca una sottostringa Z presente in entrambe, in s_1 all'inizio, in s_2 alla fine, e **se esiste** crea una nuova stringa concatenando ciò che avanza da s_1 e s_2 . Per esempio, se $s_1 = \text{mare}$ e $s_2 = \text{tema}$, la sottostringa ma rispetta i vincoli e viene creata la stringa rete ; se $s_1 = \text{flauto}$ e $s_2 = \text{golfl}$, la sottostringa t rispetta i vincoli e viene creata la stringa lautogol . Non consideriamo valide soluzioni in cui Y non esista, ossia s_2 è una sottostringa di s_1 , come per esempio in $s_1 = \text{autoscuola}$ e $s_2 = \text{auto}$.

- (5 pts) Scrivere un sottoprogramma **cercacerniera** che riceve in ingresso due stringhe e restituisce l'indice della posizione in cui comincia in s_2 la sottostringa Z , se esiste, -1 se non esiste. Con questa informazione il chiamante potrà creare la stringa risultante. Con riferimento agli esempi precedenti, se $s_1 = \text{mare}$ e $s_2 = \text{tema}$ il sottoprogramma **cercacerniera** restituisce 2. Naturalmente non è importante in questo frangente che il vocabolo che risulterebbe dalla cerniera abbia senso in italiano.
- (4 pts) Scrivere un sottoprogramma **cerniera** che riceve in ingresso due stringhe s_1 e s_2 ed un valore intero *inizio*. Il sottoprogramma crea e restituisce una nuova stringa che si ottiene concatenando i caratteri di s_1 che seguono la sottostringa di s_2 a partire dalla posizione *inizio*, ai caratteri di s_2 che precedono *inizio*. Senz'altro il valore di *inizio* è compatibile con le dimensioni di s_1 e s_2 .
- (2 pts) Scrivere un programma che acquisisce da riga di comando due stringhe e calcola, se esiste, la cerniera visualizzandola (chiamando i sottoprogrammi di cui sono state indicate le funzionalità precedentemente). Nel caso in cui la cerniera non esista, il programma non visualizza nulla.