

# Fondamenti di Informatica - A.A. 2023-2024

Scuola di Ingegneria Industriale e dell'Informazione  
Prof.ssa Cristiana Bolchini  
Appello del 13/06/2024



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

Quesito:	1	2	3	4	5	Totale
Valutazione massima (in /30):	4	8	6	6	6	30
Valutazione quesito in decimi (/10):						

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, pur di scrivere con un tratto leggibile (calcare bene).
- scrivere solo negli spazi indicati, all'interno delle apposite cornici
- tempo a disposizione: 1h 40m

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.

RESTITUIRE COMPILATO ANCHE  
NEL CASO IN CUI SI RITIRA

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 pti]

Dati i due valori  $X = +5A3_{16MS}$  e  $Y = +353_{16MS}$  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. **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 fatti sopra), utilizzando solo le caselle necessarie (**allineati a destra**) ed indicando se si è verificato overflow (segnare la casella corrispondente).

$X_{2C2}$   
  
 $X+Y_{2C2}$

$Y_{2C2}$   
  
 $X-Y_{2C2}$

Quesito 2 [8 pt]

(6 pt) Scrivere un sottoprogramma `attacco` che riceve come parametro una stringa `seq` costituita da *vocaboli* separati da punteggiatura, spazi, numeri ... Il sottoprogramma verifica se tutti i vocaboli contenuti nella stringa iniziano con lo stesso carattere e in tal caso restituisce 1, altrimenti 0. Si considerano *vocaboli* le sequenze di caratteri alfabetici adiacenti. Caratteri minuscoli e maiuscoli analoghi (A e a) devono essere

	ingresso	uscita
	Che compilatore C ci consiglia?	1
considerati uguali. Alcuni esempi:	L'anno lunare	0
	Aiuto!	1
	Abbiamo 3 alternative	1
	Dimmi di si	0

(2 pti) Scrivere un programma che acquisisce da riga di comando una stringa e avvalendosi del sottoprogramma `attacco` visualizza la stringa se tutti i vocaboli in essa contenuti iniziano con lo stesso carattere.

Soluzione al quesito \_\_\_\_\_ (indicare il quesito di cui si sta scrivendo/continuando qua la soluzione)

**Quesito 3 [6 pt]**

Scrivere un sottoprogramma `cifreparidispari` che, ricevuto in ingresso un numero intero senz'altro positivo `num` trasmette al chiamante due valori interi costituiti il primo dalle cifre di `num` in posizione pari e il secondo da quelle in posizione dispari. Per esempio, se il sottoprogramma riceve in ingresso l'intero 274820012, calcola i valori 7801 e 24202 e li trasmette al chiamante. Nel caso in cui il valore in ingresso abbia una sola cifra, il valore costituito dalle cifre pari da trasmettere è 0.

**Quesito 4** [6 pt]

Scrivere un sottoprogramma che riceve in ingresso due array monodimensionali di valori reali `valori` e `pesi` e qualsiasi altro parametro ritenuto strettamente necessario. Il sottoprogramma cerca il *sottovettore* di elementi adiacenti di `valori` che massimizza il prodotto scalare con `pesi` e **trasmette** al chiamante l'indice dell'array `valori` da cui parte tale sottovettore ed il risultato ottenuto. Nel caso ci siano piú sottovettori che permettono di ottenere lo stesso valore massimo come prodotto scalare, il sottoprogramma trasmette l'indice del primo sottovettore trovato. Nel caso il prodotto scalare non possa esser calcolato il sottoprogramma trasmette al chiamante i valori `-1` e `0.0`.

ingresso	uscita
<code>valori = [1.0 -2.0 2.5 -5.0]</code> <code>pesi = [-3.0 5.5]</code>	<code>1</code> e <code>19.75</code>
<code>valori = [1.0 -2.0 2.5 -5.0]</code> <code>pesi = [-3.0 5.5 8.0 2.1 4.6]</code>	<code>-1</code> e <code>0.0</code>

sviluppo del primo caso:

$$\begin{array}{ccc}
 \begin{bmatrix} 1.0 & -2.0 & 2.5 & -5.0 \end{bmatrix} \times \begin{bmatrix} -3.0 \\ 5.5 \end{bmatrix} & \begin{bmatrix} 1.0 & -2.0 & 2.5 & -5.0 \end{bmatrix} \times \begin{bmatrix} -3.0 \\ 5.5 \end{bmatrix} & \begin{bmatrix} 1.0 & -2.0 & 2.5 & -5.0 \end{bmatrix} \times \begin{bmatrix} -3.0 \\ 5.5 \end{bmatrix} \\
 (1.0 \times -3.0) + (-2.0 \times 5.5) = -14.0 & & 
 \end{array}$$

**Quesito 5 [6 pt]**

In relazione ad alcuni esperimenti, si utilizzano file ASCII contenenti valori numerici interi. Si sta cercando il file che contiene meno valori per poterlo utilizzare. Scrivere un sottoprogramma che ricevuta in ingresso una lista di nomi di file restituisce al chiamante il **riferimento al nome** del file che si sta cercando. È utile organizzare la propria soluzione in sottoprogrammi. Nel caso in cui si verificano problemi di accesso ad un file della lista, lo si ignori (ma si non si vada in errore). Il tipo di dato per la lista che il sottoprogramma riceve in ingresso è il seguente:

```
typedef struct _slist{
    char * nomefile;
    struct _slist * next;
} slist_t;
```

Soluzione al quesito \_\_\_\_\_ (indicare il quesito di cui si sta scrivendo/continuando qua la soluzione)