

# Fondamenti di Informatica - A.A. 2024-2025

Scuola di Ingegneria Industriale e dell'Informazione  
Prof.ssa Cristiana Bolchini  
Appello del 04/06/2025



POLITECNICO  
MILANO 1863

Cognome

Nome

Cod. Persona

Quesito:	1	2	3	4	5	Totale
Valutazione massima (in /30):	5	5	6	7	7	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.
- è possibile organizzare il proprio codice in sottoprogrammi non esplicitamente richiesti

## Sapevo già programmare:

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

Sottoprogrammi per la gestione di liste concatenate semplici con campo `dato` intero: qualora utili, si considerino già disponibili (e quindi non da sviluppare).

```
/* inserisce in testa alla lista elemento con valore specificato */
listtype * push(listtype *, int);
/* inserisce in coda alla lista elemento con valore specificato */
listtype * append(listtype *, int);
/* inserisce ordinatamente in lista elemento con valore specificato, in senso crescente */
listtype * increasing(listtype *, int);
/* inserisce ordinatamente in lista elemento con valore specificato, in senso decrescente */
listtype * decreasing(listtype *, int);
/* elimina dalla lista il primo elemento */
listtype * pop(listtype *);
/* elimina dalla lista tutti gli elementi con il valore indicato */
listtype * delete(listtype *, int);
/* restituisce il riferimento all'elemento che ha il valore indicato, se esiste, NULL altrimenti */
listtype * find(listtype *, int);
/* restituisce il numero di elementi nella lista */
int length(listtype *);
/* elimina la lista */
listtype * emptylist(listtype *);
```

**Quesito 1 [5 pts]**

Dati i due valori  $X = -8A_{16MS}$  e  $Y = 101111000_{2MS}$  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 (senza passaggi/calcoli e motivazioni, l'esercizio non viene valutato).**

**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).

$X_{2C2}$ <input style="width: 100%; height: 15px; border: 1px solid black;" type="text"/>	$Y_{2C2}$ <input style="width: 100%; height: 15px; border: 1px solid black;" type="text"/>
$X+Y_{2C2}$ <input style="width: 100%; height: 15px; border: 1px solid black;" type="text"/> <input style="width: 20px; height: 15px; border: 1px solid black;" type="checkbox"/> OVF	$X-Y_{2C2}$ <input style="width: 100%; height: 15px; border: 1px solid black;" type="text"/> <input style="width: 20px; height: 15px; border: 1px solid black;" type="checkbox"/> OVF

**Quesito 2 [5 pts]**

Scrivere un sottoprogramma in C ANSI 89 che riceve in ingresso un array di valori reali  $v$ , un valore reale  $val$ , e qualsiasi altro parametro ritenuto strettamente necessario. Il sottoprogramma individua la sequenza di valori adiacenti di  $v$  tali per cui la loro somma è pari a  $val$ . Il sottoprogramma *trasmette* al chiamante l'indice del primo elemento della sequenza e la lunghezza (così il chiamante non deve rifare i calcoli). La sequenza potrebbe non esistere: in tal caso il sottoprogramma indica  $-1$  come indice del primo elemento, il valore della lunghezza non è a questo punto rilevante. Nel caso ci siano più sequenze, il sottoprogramma individua l'ultima trovata (quella con indice più alto).

**Esempio (in evidenza le due sequenze):**

```

Ingresso:          v      1.0 -3.0 2.0 2.0 -1.5 3.2 1.0 2.0 -1.0 -2.0
                   val  0.0
Uscita:           indice:6 lunghezza:4
  
```

**Quesito 3 [6 pts]**

Scrivere un programma che acquisisce due valori interi positivi e fino a quando non sono tali li richiede. Il programma visualizza 1 se il secondo numero *compare* nel primo, 0 altrimenti. Non si facciano assunzioni sulle dimensioni dei valori. Per esempio, se l'utente inserisce 324134 e 241 il programma visualizza 1. Nel caso in cui i valori in ingresso siano 32501 e 51 il programma visualizza 0.

**Quesito 4 [7 pts]**

Scrivere un sottoprogramma che riceve in ingresso una stringa contenente il nome di un file di testo ASCII, che ha estensione, ed un valore intero di una sola cifra  $n$ . Il sottoprogramma crea e restituisce una stringa che contiene il nome di un nuovo file, creato prendendo il nome del file in ingresso, nel seguente modo:

- nome del file (senza estensione),
- carattere .
- il valore in ingresso  $n$
- l'estensione.

Per esempio, se il sottoprogramma riceve in ingresso "esercizio.c" e 3, il sottoprogramma restituisce la stringa "esercizio.3.c". Il nome del file ha senz'altro una estensione (non è necessario fare verifiche). Non si deve modificare la stringa ricevuta in ingresso.

**Quesito 5 [7 pts]**

Scrivere un sottoprogramma in C ANSI 89 che **acquisisce** dall'utente una sequenza di valori interi di lunghezza a priori di lunghezza ignota, che si ritiene terminata quando l'utente inserisce il valore 0. Il sottoprogramma restituisce al chiamante una opportuna struttura dati che contiene i dati che si ottengono ordinando tutti i valori ricevuti in senso crescente e poi prendendo un valore si e l'altro no. Il sottoprogramma restituisce una sola variabile (del tipo opportuno) al chiamante. Si faccia un uso appropriato della memoria.

**Esempio:**

```

Ingresso:   sequenza ricevuta  3 -4 5 2 9 -5 12 3 -4 11 -10 4 0
                sequenza ordinata -10 -5 -4 -4 2 3 3 4 5 9 11 12
Uscita:    valori restituiti  -10 -4 2 3 5 11
  
```

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

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

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

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

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

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