

# Fondamenti di Informatica - A.A. 2018-2019

Scuola di Ingegneria Industriale e dell'Informazione  
Prof.ssa Cristiana Bolchini  
Appello del 09/09/2019



POLITECNICO  
MILANO 1863

Cognome

Nome

Matricola o Cod. Persona

Quesito:	1	2	3	4	5	6	Totale
Peso / Punti Max:	5	4	9	4	7	1	30
Valutazione in decimi (/10):							

## Istruzioni:

- 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 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.

### Quesito 1 [5 punti]

Dati i due numeri  $X = -57_{16MS}$  e  $Y = 111001110_{2C2}$  effettuare la conversione in base 2, notazione complemento a 2 (2C2), sul numero minimo di bit per ciascuno degli operandi. 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.

Riportare qua la codifica di  $X_{2C2}$ ,  $Y_{2C2}$  sul numero minimo di bit, e i risultati delle operazioni, utilizzando solo le caselle necessarie (allineati a destra) ed indicando se si è verificato overflow.

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

### Quesito 2 [4 punti]

Rappresentare il valore reale  $Z = 41540000$  (notazione IEEE754 espresso in base 16) in base 10, notazione modulo e segno. Mostrare i passaggi fatti e riportare qua sotto solo il valore finale risultante.

$Z_{10MS}$ :

### Quesito 3 [9 punti]

(7 punti) Scrivere un sottoprogramma `crop` che ricevuta in ingresso una stringa `frase` ed un carattere `ch` restituisce una nuova stringa che contiene i caratteri compresi tra la prima e la seconda occorrenza del carattere `ch`, incluso il carattere `ch`. Per esempio, se il sottoprogramma riceve in ingresso `informatica` e `i`, il sottoprogramma restituisce la stringa `informati`. Nel caso in cui la stringa `frase` non contenga due occorrenze del carattere `ch`, restituisce `NULL`.

(2 punti) Scrivere un programma che acquisisce da riga di comando una stringa ed un carattere e chiama il sottoprogramma `crop` prima descritto e visualizza il risultato dell'elaborazione, quindi termina. Un paio di esecuzioni, da riga di comando, di esempio sono:

```
./ritagliastringa informatica a
atica
./ritagliastringa collaudo x
(null)
```

[nota: quando si esegue una istruzione `printf("%s", variabile)` e la variabile vale `NULL`, a video si vede `(null)`].

**Quesito 4 [4 punti]**

Si scriva un sottoprogramma che ricevuto in ingresso il nome di un file restituisce 1 se il file è vuoto, 0 se non è vuoto, -1 se non è possibile accedervi (per qualsiasi motivo).

**Quesito 5 [7 punti]**

(6 punti) Scrivere un sottoprogramma `extract` che riceve in ingresso una lista per la gestione dei numeri interi, e un intero `start` che vale senz'altro 0 o 1 (non è necessario gestire il caso in cui non sia così). La lista *codifica* un'informazione binaria: il valore del primo elemento indica quante volte consecutive compare il bit `start`, il secondo elemento indica quante volte compare il complemento di `start`, il terzo elemento quante volte compare il bit `start` e così fino alla fine. Il sottoprogramma scompatta tale informazione e restituisce **una nuova lista** i cui elementi contengono ciascuno 0 o 1.

Non è necessario definire due tipi diversi di dato, poichè il contenuto dell'elemento della lista è comunque sempre un intero.

Per esempio, se la lista in ingresso è quella di seguito riportata e `start` è 1:

$$3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 2 \rightarrow 1 \rightarrow |$$

il sottoprogramma restituisce la lista seguente

$$1 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow |$$

(1 punto) Definire il tipo di lista necessario per la realizzazione del sottoprogramma.

Si considerino già disponibili (e quindi non da sviluppare) i sottoprogrammi seguenti, validi per **qualsiasi** tipo di lista che gestisca "almeno" un campo intero:

```
/* inserisce in testa alla lista */
listtype * push(listtype *, int);
/* inserisce in coda alla lista */
listtype * append(listtype *, int);
/* inserisce ordinatamente in lista, in base al valore crescente */
listtype * increasing(listtype *, int);
/* inserisce ordinatamente in lista, in base al valore 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 nella lista che ha il valore indicato, se esiste, NULL altrimenti */
listtype * find(list_t *, int);
/* restituisce il numero di elementi nella lista */
int length(list_t *);
/* elimina la lista */
listtype * emptylist(list_t *);
```

**Quesito 6 [1 punto]**

Si considerino i 4 stralci di codice qua riportati: per ciascuno di essi si vuole calcolare la quantità di memoria allocata per le variabili dichiarate ed utilizzate (quindi la memoria allocata nel momento in cui si arriva all'ultima istruzione del codice riportato). Si utilizzi l'operatore `sizeof`. Per i primi due stralci, il calcolo è già stato fatto, riportare il risultato nell'apposito spazio il calcolo per gli ultimi due stralci.

```
int a;
...
scanf("%d", &a);
```

```
int v[NUM], i;
...
for(i = 0; i < NUM; i++)
    scanf("%d", &v[i]);
```

```
int * p, i, ndati;
...
if(p = (int *)malloc(ndati*sizeof(int))){
    for (i = 0; i < ndati; i++)
        scanf("%d", p+i);
    ...
}
```

```
typedef struct _s {
    int val;
    struct _s * next;
} t_listi;
...
t_listi * head = NULL;
int i, n, ndati;
...
ndati = 0;
scanf("%d", &n);
while(n != STOP){
    head = append(head, n);
    ndati++;
    scanf("%d", &n);
}
```

quantità di memoria complessiva allocata:

`sizeof(int)`

`NUM * sizeof(int) +`  
`sizeof(int)`

.

.