



Politecnico di Milano

Corso di Laurea in Ingegneria Civile

A.A. 2018/19

**Informatica**  
**M. Lattuada – A. Miele**  
**10 Luglio 2019**

Cognome e nome

Firma

Matr. o CodPersona

Domanda	1	2	3	4	5	
Punteggio max	4	7	6	6	4	TOT
Punteggio						

La durata della prova è di 2 ore. Non è consentito consultare libri o appunti, non è consentito l'uso di calcolatrici o cellulari.

**Domanda 1 – 4 punti**

Codificare i numeri  $A=23$  e  $B=-51$  in base 2 complemento a 2 su 10 bit. Eseguire poi l'operazione  $B-A$  su 10 bit indicando se si è verificato o meno overflow. **Riportare e motivare tutti i passaggi.**

**Domanda 2 – 7 punti**

a) Scrivere un sottoprogramma `analizza_numero` che riceve come parametri due numeri interi `n` e `val`. Se il numero `n` contiene almeno due cifre, il sottoprogramma estrae le prime due cifre da destra e verifica se tale nuovo numero è multiplo di `val`. In caso affermativo il sottoprogramma restituisce 1 altrimenti, se il numero composto dalle prime due cifre non è multiplo di `val` oppure il numero `n` è composto da una sola cifra, ritorna 0.

b) Scrivere un programma che apre un file di testo `testo.txt` contenente una serie di lunghezza indefinita (zero o più) di numeri interi. Il sottoprogramma analizza ciascun numero mediante il sottoprogramma `analizza_numero` specificando la costante 15 come secondo parametro, e stampa a video ciascuna coppia tale per cui il primo valore soddisfa le condizioni valutate da `analizza_numero` mentre il secondo no. Ad esempio, se il file contiene la seguente sequenza:

845    945    460    461    7    15    15    16    0    9

Il programma stamperà a video:

460    461  
15    16

**Domanda 3 – 6 punti**

Scrivere un sottoprogramma che riceve come parametro una matrice di interi `m` di dimensioni  $N \times N$ , dove  $N$  è una costante definita con una direttiva di `#define`. Il sottoprogramma verifica se a) gli elementi di ciascuna riga sono disposti in ordine crescente e b) almeno 2 righe hanno somma pari. Il sottoprogramma restituirà 1 se entrambe le condizioni sono rispettate altrimenti 0.

Per esempio, se  $N=5$  e  $m$  contiene

```
1 2 3 4 5
1 2 4 5 5
2 2 3 4 5
1 1 1 1 1
1 1 2 4 6
```

Il sottoprogramma restituirà 1

Se invece  $m$  contiene

```
1 2 3 4 5
5 1 2 3 4
4 5 1 2 3
3 4 5 1 2
2 3 4 5 1
```

Il sottoprogramma restituirà 0

#### Domanda 4 – 6 punti

Date le seguenti strutture dati che rappresentano le corse delle linee autobus di una città:

```
#define N_CORSE 100
```

```
#define MAX_STR 100
```

```
typedef struct{
    char tipo_autobus[MAX_STR+1];
    int numero_corse;
    int numero_passengeri;
    int posti_totali;
} t_corsa;
```

```
typedef struct{
    t_corsa corse[N_CORSE];
    int n_corse;
    char partenza[MAX_STR+1];
    char arrivo[MAX_STR+1];
    int numero_linea;
} t_linea;
```

a) Realizzare un sottoprogramma che riceve come parametri un array di linee `lb`, la sua dimensione `dim` e restituisce un intero come risultato. Per ciascuna corsa di ciascuna linea, il sottoprogramma calcola la percentuale di riempimento (quanti dei posti totali sono occupati), quindi calcola la media della percentuale di riempimento per ciascuna linea, ed infine identifica qual è la linea con la più bassa percentuale media di riempimento restituendo il numero della linea.

b) Scrivere uno stralcio di `main` in cui viene dichiarato un array di 10 elementi di tipo `t_linea` e qualsiasi altra variabile eventualmente necessaria. **OMETTENDO** la parte di acquisizione dei dati, il programma esegue la chiamata al sottoprogramma precedentemente definito e visualizza il risultato.

#### Domanda 5 – 4 punti

Completare il seguente programma Fortran in modo tale che, una volta acquisiti due array di 10 interi `a` e `b`, ne calcoli il prodotto elemento per elemento memorizzando il risultato in `prodotto`. In programma in seguito somma gli elementi di `prodotto`. Se il risultato della somma è positivo, il programma ne stampa il valore, altrimenti stampa il suo opposto. Dichiarare le ulteriori eventuali variabili ritenute necessarie.

```
PROGRAM cosafa
    IMPLICIT NONE
    INTEGER, PARAMETER :: DIM = 10
    INTEGER, DIMENSION(1:DIM) :: a, b, prodotto
    ! AGGIUNGERE VARIABILI

    READ(*,*) a
    READ(*,*) b

    ! COMPLETARE CODICE

    STOP
END
```