



Politecnico di Milano

Corso di Laurea in Ingegneria Civile

A.A. 2018/19

Informatica
M. Lattuada – A. Miele
20 Febbraio 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

a) Dati i numeri in binario complemento a due $A=00110101$ e $B=01110010$, eseguire l'operazione di sottrazione $A-B$ indicando e motivando se si è verificato o meno overflow. **Riportare e motivare tutti i passaggi.**

b) Disegnare lo schema della macchina di Von Neumann ed indicare il nome di ciascun componente.

Domanda 2 – 7 punti

a) Scrivere un sottoprogramma `confrontastringhe` che riceve come parametro due stringhe `str1` e `str2`. Il sottoprogramma stampa a video i caratteri uguali che si trovano nella stessa posizione in entrambe le stringhe e restituisce il loro conteggio.

b) Scrivere un programma che apre un file di testo `testo.txt` contenente una serie di lunghezza indefinita (zero o più) di parole ciascuna lunga al massimo 15 caratteri. Il programma legge ciascuna parola contenuta nel file e la confronta con la sua precedente mediante l'invocazione del sottoprogramma `confrontastringhe` (il primo confronto viene eseguito fra la prima e la seconda parola). Il programma infine stampa a video la somma complessiva di tutti i caratteri comuni tra ciascuna coppia di parole consecutive. Ad esempio, se il file contiene il seguente testo:

```
Gatto Gattini 120 10 alce dice bica informatica 34tre tre tr
```

Il programma stamperà a video: `Gatt 1 ce ic tred` infine stamperà il valore 11.

Domanda 3 – 6 punti

Scrivere un programma che chiede all'utente i dati di una matrice 5×5 di interi. Il programma conta e visualizza il numero di righe della matrice che soddisfano le seguenti proprietà:

- Ci sono almeno due elementi dispari (o il cui modulo è dispari nel caso di numero negativo).
- La somma degli elementi in posizione con indice pari è maggiore della corrispondente somma della riga precedente (per la prima riga si esegua il confronto con il valore 0).
- Non ci sono più di due elementi multipli di 3 (o il cui modulo è multiplo di 3 nel caso di numero negativo).

Per esempio acquisita la matrice:

```
-2 -4 6 -1 5
-3 3 4 -1 4
-7 -8 16 -1 -3
-1 -3 -6 -3 -3
-2 -4 6 -2 5
```

Il programma stamperà a video il valore “2” poiché soltanto la prima e la terza riga soddisfano tutte le proprietà richieste.

Domanda 4 – 6 punti

Date le seguenti strutture dati che rappresentano un’edicola:

```
#define N_RIVISTE 1000
#define MAX_STR 100
```

```
typedef struct{
    char titolo[MAX_STR+1];
    float prezzo;
    int num_uscite;
    int pagine_per_uscita;
} t_rivista;
```

```
typedef struct{
    t_rivista riv [N_RIVISTE];
    int n_riviste;
    char nome[MAX_STR+1];
    char citta[MAX_STR+1];
} t_edicola;
```

a) Realizzare un sottoprogramma che riceve come parametri un array di edicole `edi`, la sua dimensione `dim` e un array `max_ed` di tipo `char` in cui verrà salvato il risultato. Per ciascuna edicola, il sottoprogramma calcola il numero medio di pagine di tutte le riviste vendute, considerando di ciascuna rivista tutte le relative uscite. Il sottoprogramma identifica l’edicola avente il valore medio più alto e ne salva il nome nel parametro `max_ed`.

b) Scrivere uno stralcio di `main` in cui viene dichiarato un array di 10 elementi di tipo `t_edicola` 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

Il seguente programma acquisisce la matricola ed i punteggi ottenuti nei 3 esercizi di un esame da parte di 10 studenti ed analizza i risultati. Tali informazioni sono organizzate in una matrice di interi con una riga per ogni studente in cui la prima colonna memorizza la matricola, le colonne successive i punteggi dei singoli esercizi e l’ultima colonna il punteggio totale. Completare il programma come specificato sotto.

```
PROGRAM risultatiesame
    IMPLICIT NONE
    INTEGER, DIMENSION(1:10, 1:5) :: risultati
    INTEGER :: i, esercizioPeggior
    REAL, DIMENSION(1:3) :: mediaEsercizi
    READ(*,*) (risultati(i,1:4), i=1,10)
    DO i = 1,10
        risultati(i,5) = !Da completare punto a)
    END DO
    !Da completare punto b)
    WHERE (risultati(:,5)<18) !Da commentare punto c)
        risultati(:,5) = 0
    END WHERE
    !Da completare punto d)
END PROGRAM
```

- Completare il primo ciclo del programma con il calcolo del voto di ciascun studente (la somma del punteggio ottenuto nei tre esercizi).
- Completare il codice calcolando per ciascun esercizio il punteggio medio ottenuto e stampando a video il punteggio medio dell’esercizio andato peggio.
- Descrivere qual è l’effetto del costrutto `WHERE`.
- Stampare a video matricola e voto finale di ciascuno studente.