

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Appello del 24/1/2022		COGNOME E NOME						
	Fila	Colonna	MATRICOLA						
<div style="text-align: right;"> Spazio riservato ai docenti <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table> </div>									

- Il presente plico contiene 3 esercizi e 2 domande e **deve essere debitamente compilato con cognome e nome, e numero di matricola.**
- Il tempo a disposizione è di **1 ora e 30** minuti.
- Non separate questi fogli. Scrivete la soluzione solo sui fogli distribuiti, utilizzando il retro delle pagine in caso di necessità. Cancellate le parti di brutta (o ripudiate) con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- È possibile scrivere a matita (e non occorre ricalcare al momento della consegna!).
- **Per chi sta facendo l'esame cartaceo è vietato utilizzare calcolatrici, telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- **Qualsiasi tentativo di comunicare con altri studenti comporta l'espulsione dall'aula.**
- È possibile ritirarsi senza penalità.
- **Non è possibile lasciare l'aula conservando il tema della prova in corso.**
- **Per il superamento dell'esame è necessario dimostrare sufficienti competenze sia in C sia in Matlab, e quindi saper impostare correttamente esercizi in entrambi i linguaggi.**
- **MOLTO IMPORTANTE: risposte poco leggibili** (scritte molto piccolo, con calligrafia poco comprensibile, o molto disordinate) **non saranno considerate nella valutazione.**

Esercizio 1 (10 punti)

Gestione di un concorso musicale in linguaggio C.

Al Concorso Musicale Eurovisione hanno partecipato 50 nazioni. I voti del pubblico e della giuria tecnica di ogni nazione sono salvati rispettivamente nelle matrici VP e VG, in cui ogni riga corrisponde a un paese votante e ogni colonna a un paese che riceve il voto, e il valore di ogni cella corrisponde a un valore numerico tra 0 e 10, con 0 che corrisponde a “non voto”.

Considerando le matrici VP e VG dichiarate come segue e interamente popolate, rispondere alle successive domande.

```
#define N_PART 50
typedef int matriceVoti [N_PART][N_PART];

matriceVoti VP, VG;
```

Domanda 1.1 (punti 2)

Scrivere un frammento di codice in linguaggio C per verificare che, per ciascuna nazione, sia il pubblico che la giuria tecnica non abbiano votato per la loro stessa nazione.

Domanda 1.2 (punti 4)

Scrivere un frammento di codice in linguaggio C per verificare che la giuria tecnica di ogni nazione abbia inserito esattamente 10 voti non nulli da 1 a 10, tutti differenti tra loro.

Domanda 1.3 (punti 2)

Scrivere un frammento di codice in linguaggio C per memorizzare un vettore di punteggi totali (somma dei punteggi del pubblico e della giuria tecnica) per ogni nazione, e successivamente scrivere a schermo l'indice della nazione vincitrice (con il più alto punteggio totale) e il relativo punteggio totale.

Domanda 1.4 (punti 2)

Utilizzando il vettore calcolato al punto precedente 1.3, scrivere un frammento di codice in linguaggio C per verificare la presenza di nazioni che non hanno ricevuto nessun voto (hanno totalizzato zero punti). Inserire gli indici di tali nazioni in un altro vettore senza lasciare buchi, e scrivere a schermo quante nazioni non hanno ricevuto nessun voto.

Soluzione

```
#define N_PART 50
typedef int matriceVoti[N_PART][N_PART];
matriceVoti VP, VG;
```

Domanda 1.1

```
int i;
int trovato;

trovato = 0;
for (i = 0; i < N_PART && !trovato; i++) {
    if (VP[i][i] != 0 || VG[i][i] != 0) {
        trovato = 1;
    }
}
if (!trovato) {
    printf("Nessun paese si e' autovotato\n");
}
```

Domanda 1.2

```
int non_nulli, trovato;
int i, j, k;
```

```
// ciclo per scorrere i paesi votanti (righe di VG)
for (i = 0; i < N_PART; i++) {
    trovato = 0; // flag per inconsistenza di voti su una riga
    non_nulli = 0; // contatore dei voti non nulli sulla riga
    // ciclo per scorrere i voti assegnati (colonne di VG)
    for (j = 0; j < N_PART && !trovato; j++) {
        if (VG[i][j] != 0) {
            non_nulli++; // incremento contatore voti non nulli
            if (VG[i][j] > 10 || VG[i][j] < 0) {
                trovato = 1; // voti minori di zero o maggiori di 10 sono inconsistenti
            } else {
                // confronto il voto con i successivi (devono essere tutti diversi)
                for (k = j + 1; k < N_PART; k++) {
                    if (VG[i][j] == VG[i][k]) {
                        trovato = 1;
                    }
                }
            }
        }
    }
}
// i voti sono validi se non ho trovato inconsistenze e i voti non nulli sono 10
if (!trovato && non_nulli == 10) {
    printf("I voti della nazione %d", i + 1);
    printf(" sono validi\n");
}
}
```

Domanda 1.3

```
int punteggi[N_PART];
int somma_punti;
int i, j, max, i_max;

for (j = 0; j < N_PART; j++) {
    somma_punti = 0;
    for (i = 0; i < N_PART; i++) {
        somma_punti = somma_punti + VP[i][j] + VG[i][j];
    }
    punteggi[j] = somma_punti;
}

max = punteggi[0];
i_max = 0;
for (i = 1; i < N_PART; i++) {
    if (punteggi[i] > max) {
        max = punteggi[i];
        i_max = i;
    }
}
printf("La nazione vincitrice e' la numero %d", i_max + 1);
printf(" con punteggio %d\n", max);
```

Domanda 1.4

```
int zero_punti[N_PART];
int i, indice;

indice = 0;
for (i = 0; i < N_PART; i++) {
    if (punteggi[i] == 0) {
        zero_punti[indice] = i;
        indice++;
    }
}
printf("Il numero delle nazioni senza voti e' %d", indice);
```

Esercizio 2 (10 punti)

Alla conferenza sull'ambiente POLIZIOTTO26 sono state verificate le emissioni di CO₂ negli ultimi due anni dei paesi partecipanti. Tali valori (numeri positivi) sono stati registrati in un file *emissioni.mat*, più precisamente in due vettori *E20* ed *E21*, contenenti rispettivamente le emissioni di CO₂ dei paesi partecipanti durante gli anni 2020 e 2021. Durante la conferenza sono stati definiti i livelli minimi per considerare un paese virtuoso o canaglia:

- i paesi virtuosi sono quelli che hanno abbassato le loro emissioni di CO₂ almeno del 5% tra il 2020 e il 2021;
- i paesi canaglia sono quelli che hanno incrementato le loro emissioni di CO₂ almeno del 2% tra il 2020 e il 2021.

Domanda 2.1 (punti 3)

Si sviluppi una funzione **analisi** in linguaggio Matlab che, dati due vettori corrispondenti alle emissioni di CO₂ dell'anno precedente e dell'anno in corso per i paesi partecipanti alla conferenza, restituisca un vettore con le variazioni percentuali degli stati partecipanti, un vettore logico che indichi se ognuno dei paesi è virtuoso (0 non virtuoso e 1 virtuoso) e un vettore logico se ognuno dei paesi è canaglia (0 non canaglia e 1 canaglia).

Domanda 2.2 (punti 1)

Si scriva uno script in linguaggio Matlab che esegua le seguenti operazioni:

- 1) legga dal file *emissioni.mat* i vettori *E20* e *E21*.
- 2) crei due vettori logici contenenti uno i paesi canaglia e l'altro i paesi virtuosi e li visualizzi a schermo.
- 3) Salvi in un file *risultati* i vettori risultanti (evitando di salvare tutto il resto del workspace).

Domanda 2.3 (punti 3)

- 4) visualizzi a schermo la percentuale di incremento/decremento globale, ovvero la percentuale di incremento/decremento delle emissioni cumulate su tutti i paesi partecipanti.
- 5) visualizzi a schermo gli indici, relativi ai vettori *E20* ed *E21*, dei paesi che stanno avendo il massimo incremento e il massimo decremento della percentuale di CO₂ (se ve ne fossero più di uno a pari merito, se ne estragga almeno uno di essi).

Domanda 2.4 (punti 3)

- 6) disegni un grafico di una linea che identifica sull'asse delle ascisse i differenti paesi e sull'asse delle ordinate i valori delle variazioni percentuali di CO₂. Si evidenzino con un asterisco rosso il punto con la massima percentuale di incremento di CO₂, e con un pallino giallo quello con la massima percentuale di decremento. Si aggiungano titolo e etichette agli assi del grafico.

Suggerimento: esiste una soluzione all'esercizio che non richiede di utilizzare cicli

Soluzione

% Domanda 2.1

```
function [perc, virtuosi, canaglia] = analisi(e1, e2)
```

% funzione che calcola le percentuali di incremento di CO2 e valuta se i paesi sono virtuosi o canaglia

% input: e1: vettore delle emissioni per l'anno passato

% e2: vettore delle emissioni per l'anno corrente

% output: perc: vettore contenente la percentuale delle variazioni paese per paese

% virtuosi: vettore logico che specifica se ognuno dei paesi è virtuoso o non
(riduzione delle emissioni di CO2 di almeno 5%)

% canaglia: vettore logico che specifica se ognuno dei paesi è canaglia o non
(incremento delle emissioni di CO2 di almeno 2%)

```
perc = (e2 - e1) ./ e1 * 100;
```

```
virtuosi = perc <= -5;
```

```
canaglia = perc >= 2;
```

```
end
```

% Domanda 2.2

```
clear;
```

```
clc;
```

```
close all;
```

```
load emissioni;
```

```
[perc, virtuosi, canaglia] = analisi(E20, E21)
```

```
save risultati virtuosi canaglia;
```

% Domanda 2.3

```
perc_glob = (sum(E21) - sum(E20)) ./ sum(E20) * 100
```

```
migliore = find(perc == min(perc), 1)
```

```
peggiore = find(perc == max(perc), 1)
```

% Domanda 2.4

```
figure();  
plot(perc);  
hold on;  
plot(peggiore, max(perc), 'r*');  
plot(migliore, min(perc), 'yo');  
title('Aumenti percentuali');  
xlabel('Paesi');  
ylabel('Percentuali');
```

Esercizio 3 (6 punti)

Su un cellulare Nokia 3310 i numeri erano rappresentati in complemento a due (CP2) con 8 bit, e i caratteri erano rappresentati con 8 bit in base alla tabella ASCII.

Domanda 3.1 (punti 2)

Sapendo che la memoria dedicata ai messaggi era 10 KByte e che un messaggio conteneva al massimo 100 caratteri, quanti messaggi poteva immagazzinare in memoria prima di riempirsi?

Domanda 3.2 (punti 3)

Su questo dispositivo, che risultato davano le seguenti operazioni: $41 + 38$, e $-120 - 57$. Motivare adeguatamente le risposte con i dovuti calcoli.

Domanda 3.3 (punti 1)

Su questo dispositivo, che risultato dava l'operazione 'b' - 'f'?

Soluzione

1) Se i caratteri sono rappresentati in 8 bit = 1 Byte, la memoria del cellulare può contenere 10 kByte / 100 Byte = $10 \cdot 1024 / 100 = 102$ messaggi da 100 caratteri.

2) 41 in CP2 a 8 bit -> 00101001

38 in CP2 a 8 bit -> 00100110

somma: 01001111 -> 79

-120 in CP2 a 8 bit -> 10001000

-57 in CP2 a 8 bit -> 11000111

somma: 01001111 -> 79, overflow dato che -177 non può essere in CP2 con 8 bit

3) 'b' è il simbolo con valore ASCII = 98 -> 98 in CP2 a 8 bit -> 01100010

'f' è il simbolo con valore ASCII = 102 -> -102 in CP2 a 8 bit -> 10011010

somma: -4 in CP2 a 8 bit -> 11111100

NOTA: In realtà non serve sapere a memoria la tabella ASCII per rispondere a questa domanda. Basta ricordarsi che le lettere nella codifica ASCII sono rappresentate in sequenza crescente; quindi, la differenza tra la lettera 'b' ed 'f' è -4 che codificato in complemento a 2 su 8 bit è 11111100

Domanda 1 (3 punti)

Si consideri il seguente frammento di codice:

```
...  
int a;  
scanf("%s", a);  
int vec[a];  
vec[50] = 15;  
...
```

Quali sono i rischi collegati a questo codice? Suggestire quali possono essere le modifiche che evitino tali rischi.

Soluzione

- la scanf legge un'intero la la format string è %, occorre correggere il codice con scanf("%d",&a);
- l'istruzione vec[50] = 15 causa un'accesso oltre la dimensione dell'array se l'utente inserisce un numero inferiore a 51. Nota: 50 non basta perché in un'array da 50 elementi gli indici validi vanno da 0 a 49.
- questo codice causa stack overflow se l'utente inserisce un numero molto grande (o negativo!)
- entrambi i problemi possono essere risolti controllando la correttezza del valore di a dopo la scanf o chiedendo all'utente di inserire un'altro numero finché il numero inserito è accettabile

Domanda 2 (3 punti)

Si confronti il seguente frammento di codice C

```
if(((a ^ c) && !a) || (a && !b && !c)) printf("Vero!\n");
```

con il seguente frammento di codice Matlab

```
if (~a & c) | (a & ((~b & ~c) | (b & c))) disp("Vero!"); end
```

In cui a, b, c sono variabili booleane. Dire se le due espressioni sono equivalenti, o alternativamente fornire una combinazione dei valori a, b c per cui il comportamento è diverso. E' richiesto di motivare i passaggi logici che portano alla vostra conclusione.

Soluzione

Computando la tabella della verità si scopre che per a=1 b=1 c=1 la condizione in C è falsa, mentre quella in Matlab è vera. L'operatore ^ in C è lo xor (slide 20 algebra booleana, tavola di verità a slide 10).

abc	$(a \oplus c)\bar{a} + a\bar{b}\bar{c}$	$\bar{a}c + a(\bar{b}\bar{c} + bc)$
000	0 0 0 0 1 0 0 1 1	1 0 0 0 0 1 1 0 0
001	0 1 1 1 1 1 0 1 0	1 1 1 1 0 0 1 0 0 1
010	0 0 0 0 1 0 0 0 1	1 0 0 0 0 0 1 0 1 0
011	0 1 1 1 1 1 0 0 0	1 1 1 1 0 0 0 1 1 1
100	1 1 0 0 0 1 1 1 1	0 0 1 1 1 1 1 0 0
101	1 0 1 0 0 0 1 1 0	0 1 0 1 0 1 0 0 1
110	1 1 0 0 0 0 1 0 1	0 0 0 1 0 0 1 0 1 0
111	1 0 1 0 0 0 1 0 0	0 1 1 1 1 0 0 1 1 1