

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Appello del 15/07/2022	COGNOME E NOME					
		MATRICOLA					
		Spazio riservato ai docenti <table border="1" style="width: 100%; height: 20px;"> <tr> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> </table>					

- 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 45 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!).
- **Qualsiasi tentativo di comunicare con altri studenti, comporta l'annullamento dell'esame.**
- 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)

I risultati di un campionato di calcio sono rappresentati in un archivio con la seguente struttura:

```
#define N_PARTITE 38
#define N_SQUADRE 20

typedef struct{
    int goal_fatti[N_PARTITE];
    int goal_subiti[N_PARTITE];
    char[50] nome_squadra;
} team;

team squadre[N_SQUADRE];
```

In tale struttura, l'*i*-esimo elemento del vettore `goal_fatti` (`goal_subiti`) corrisponde al numero di goal realizzati (subiti) nella *i*-esima partita dalla squadra `nome_squadra`.

- 1) **(3 punti)** Si scriva un frammento di codice in linguaggio C che controlli che il valore di ciascun campo di ciascuno dei due vettori `goal_fatti` e `goal_subiti` sia maggiore o uguale a zero, e stampi a schermo i nomi delle squadre per cui ciò non avviene, oppure stampi "Tutto corretto".
- 2) **(3 punti)** Per ciascuna squadra, si stampi a schermo il numero massimo di `goal_subiti` e l'indice corrispondente alla partita in cui questo è avvenuto. Se è avvenuto in più partite, si stampi l'indice corrispondente a una di queste.
- 3) **(4 punti)** Si supponga che a ciascuna squadra in ciascuna partita vengano attribuiti 3 punti se `goal_fatti > goal_subiti`, 1 punto se `goal_fatti = goal_subiti` e 0 punti se `goal_fatti < goal_subiti`. Scrivere un frammento di codice che calcoli il punteggio totale di ciascuna squadra e stampi a video il nome della squadra che ha totalizzato il punteggio maggiore. In caso più squadre abbiano ottenuto tale punteggio maggiore, stampare il nome di tutte tali squadre.

SOLUZIONE

Punto 1

```
int corretto = 1;
int s, p;

for (s = 0; s < N_SQUADRE; s++)
    for (p = 0; p < N_PARTITE; p++)
        if (!(squadre[s].goal_fatti[p] >= 0 && squadre[s].goal_subiti[p] >= 0)) {
            printf("%s\n", squadre[s].nome_squadra);
            corretto = 0;
        }
if (corretto)
    printf("\nTutto corretto");
```

Punto 2

```
int s, p, max, maxp;

for (s = 0; s < N_SQUADRE; s++) {
    max = squadre[s].goal_subiti[0];
    maxp = 0;
    for (p = 1; p < N_PARTITE; p++)
        if (squadre[s].goal_subiti[p] > max) {
            max = squadre[s].goal_subiti[p];
            maxp = p;
        }
    printf("La squadra %s ha subito %d goal la partita numero %d\n",
           squadre[s].nome_squadra, max, maxp+1);
}
```

Punto 3

```
int s, p, max, tot[N_SQUADRE];

max = 0;

for (s = 0; s < N_SQUADRE; s++) {
    tot[s] = 0;

    for (p = 0; p < N_PARTITE; p++) {
        if (squadre[s].goal_fatti[p] > squadre[s].goal_subiti[p])
            tot[s] += 3;
        else if (squadre[s].goal_fatti[p] == squadre[s].goal_subiti[p])
            tot[s] += 1;
    }

    if (tot[s] > max)
        max = tot[s];
}

printf("Il punteggio maggiore è stato di %d punti, ed è stato totalizzato dalle
       seguenti squadre:\n", max);
for (s = 0; s < N_SQUADRE; s++) {
    if (tot[s] == max)
        printf("%s\n", squadre[s].nome_squadra);
}
```

Esercizio 2 (10 punti)

L'holter pressorio è un esame diagnostico non invasivo e indolore che consente di misurare, a intervalli temporali regolari, la pressione sanguigna arteriosa di un soggetto durante un'intera giornata, e di registrarne i valori.

Si supponga che, per essere poi analizzati, i dati vengano memorizzati in un file MATLAB "misure.mat" contenente una matrice di questo tipo:

1	2	3	4	5	6	...
110	121	-1	115	140	120	...
68	71	-1	82	82	75	...

Dove la prima riga rappresenta il numero (id) dell'intervallo temporale, la seconda riga la pressione massima (o sistolica) e la terza riga la pressione minima (o diastolica). I valori -1 indicano che la misurazione non è andata a buon fine.

Si scriva uno script in linguaggio MATLAB che:

- 1) carichi il file "misure.mat" in una opportuna variabile MATLAB (**1 punto**);
- 2) calcoli i valori medi di pressione massima e minima, escludendo i valori errati (-1) (**2 punti**);
- 3) sostituisca nella variabile creata al punto 1) i valori uguali a -1 con i corrispondenti valori medi calcolati al punto 2) (**2 punti**);
- 4) calcoli e memorizzi in due opportune variabili *min_massima* e *max_minima* gli id di tutti gli intervalli di tempo in cui si è registrata rispettivamente la minima pressione massima e la massima pressione minima (**2 punti**);
- 5) disegni un grafico che abbia sulle ascisse gli intervalli di tempo, sulle ordinate i valori di pressione sanguigna e che rappresenti due linee, una relativa ai valori della pressione massima e una relativa ai valori della pressione minima misurati in ogni intervallo di tempo. Inoltre, si aggiungano al grafico delle rette verticali in concomitanza degli intervalli di tempo *min_massima* e *max_minima*, e infine si aggiungano due rette orizzontali tratteggiate corrispondenti rispettivamente al valore medio della pressione massima e della pressione minima (**3 punti**).

SOLUZIONE

1)

```
misure = load('misure.mat');
```

2)

```
index_high = misure(2,:) != -1;
index_low = misure(3,:) != -1;
high_mean = mean(misure(2, index_high), 2);
low_mean = mean(misure(3, index_low), 2);
```

3)

```
index_high = misure(2,:) == -1;
index_low = misure(3,:) == -1;
misure(2, index_high) = high_mean;
misure(3, index_low) = low_mean;
```

4)

```
min_massima = misure(1, misure(2,:) == min(misure(2,:)) )
max_minima = misure(1, misure(3,:) == max(misure(3,:)) )
```

5)

```
figure();
hold on
plot(misure(1,:), misure(2,:), 'r')
plot(misure(1,:), misure(3,:), 'b')
xline(min_massima, 'black')
xline(max_minima, 'black')
yline(low_mean, 'black--')
yline(high_mean, 'black--')
plt.legend('Massima', 'Minima')
hold off
```

Esercizio 3 (6 punti)

A) (3 punti) Si consideri il seguente script in linguaggio MATLAB. Si dica quante e quali linee del codice corrispondono all'interazione con una periferica (si indichino i numeri di riga riportati di fianco al testo). Per fare questo, si ipotizzi che la funzione *calcolaCosto* utilizzata dallo script non interagisca con alcuna periferica.

```
1) energiaProdotta = [1:1000];
2) energiaConsumata = [0.5:1000];
3) prezzoEnergia = input('Inserire il prezzo energia: ');

4) x = 1:length(energiaProdotta);

5) ylabel('Energia prodotta');
6) xlabel('Minuto');
7) title('Produzione energia');
8) plot(x, energiaProdotta);

9) indici = find(energiaProdotta >= energiaConsumata);
10) esubero = energiaProdotta(indici);

11) [spesaTotale, media] = calcolaCosto(energiaProdotta, energiaConsumata,
    prezzoEnergia);

12) disp(['Spesa totale per l''energia elettrica: ', num2str(spesaTotale)]);
13) disp(['I punti di esubero sono: ', num2str(esubero)]);
14) disp(num2str(media));
15) save dati;
```

B) (3 punti) Si supponga che su un calcolatore monoprocesso ci sia un processo pronto (P2), due processi in attesa (P3 e P4), e uno in esecuzione (P1). Si supponga inoltre che P1 esegua il codice dello script riportato sopra. Si spieghi che cosa succede quando l'unità di elaborazione esegue la prima istruzione di P1 relativa all'interazione con una periferica.

SOLUZIONE

Punto A)

Si hanno 9 interazioni con periferiche nei punti 3, 5, 6, 7, 8, 12, 13, 14, 15 (anche le righe 5, 6, 7, causando un aggiornamento immediato dello schermo, interagiscono con periferiche).

Punto B)

Quando l'unità di elaborazione esegue la prima istruzione di interazione con una periferica, si verifica un'interruzione interna che viene gestita dal sistema operativo (supervisor call). Il sistema operativo sposta P1 in stato di attesa e manda in esecuzione l'unico processo pronto P2. Nella nuova situazione non ci sarà nessun processo in stato di pronto, un processo in stato di esecuzione (P2) e tre processi in stato di attesa (P3, P4 e P1).

Domanda 1 (3 punti)

Dato il seguente frammento di codice:

```
#include <stdio.h>

int main() {
    int i;
    for (i=1; i>0; i++)
        printf("Iterazione del ciclo: %d\n", i);
    printf("Terminato\n");
    return 0;
}
```

Spiegare cosa succede e perché.

Domanda 2 (3 punti)

Dato un array di caratteri in cui i caratteri ammissibili sono solo quelli che rappresentano i numeri da 0 a 9. Data (non si deve scrivere) una funzione in C che, preso in ingresso un array di questi caratteri, restituisce il valore intero equivalente, ad esempio, dato '5' '0' '1' restituisce l'intero 501. Si dica se i numeri restituiti dalla funzione possono essere negativi. Giustificare opportunamente la risposta.