

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Appello del 12/2/2024		COGNOME E NOME
	Fila	Colonna	MATRICOLA

- 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'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.**

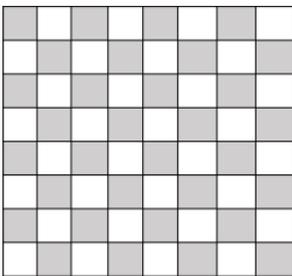
Esercizio 1 (10 punti)

Si considerino le seguenti strutture dati in linguaggio C che rappresentano una casella in una scacchiera usata per giocare a dama (le definizioni sono mostrate su due colonne per motivi di spazio):

```
typedef enum {falso, vero} Booleano;  
typedef enum {bianco, nero} Colore;  
typedef struct {  
    Colore c;  
    Booleano damone;  
} Pedina;  
  
typedef struct {  
    Colore c;  
    Booleano occupata;  
    Pedina p;  
} Casella;  
typedef Casella Scacchiera[8][8];
```

Le strutture rappresentano il fatto che una casella può essere di colore bianco o nero, può essere occupata oppure no e, se occupata, è associata a una pedina del gioco della dama. Una pedina è caratterizzata da un colore, bianco o nero, e dal fatto di essere un damone o no (i damoni nel gioco della dama hanno maggiori possibilità di movimento rispetto alle pedine normali). Si risponda alle seguenti domande utilizzando il linguaggio C.

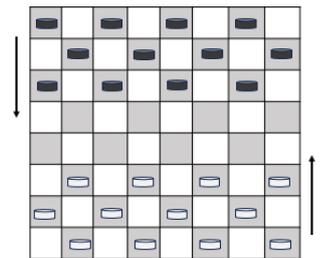
Domanda 1.1 (punti 3)



Si dichiara una variabile di tipo Scacchiera e la si inizializzi in modo tale che le caselle della scacchiera siano di colore alternato come nella figura a sinistra e non contengano pedine.

Domanda 1.2 (punti 3)

A partire dalla scacchiera vuota definita al punto precedente, si inseriscano le pedine della Dama nella configurazione iniziale mostrata nella figura a destra.

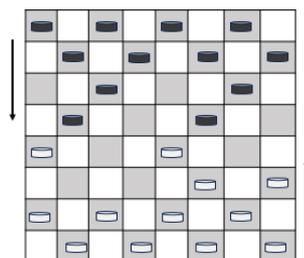


Per semplicità, si scriva solo il codice per il riempimento della parte superiore della scacchiera con le pedine nere.

Domanda 1.3 (punti 4)

Si scriva un frammento di codice che aiuti un utente nello sviluppo del gioco **relativo alle pedine nere**. In particolare, il codice deve chiedere all'utente qual è la posizione sulla scacchiera della **pedina nera** che vuole muovere, **controllare che i dati acquisiti siano corretti**, e, se il movimento è possibile in base alla regola indicata qui sotto, modifica lo stato della scacchiera di conseguenza, conferma all'utente di aver eseguito il movimento e qual è la nuova posizione della pedina. In caso contrario, il codice stampa un messaggio di errore per l'utente.

Regola per il movimento di una pedina: una pedina si muove sempre in diagonale sulle caselle nere libere, avanzando di una casella alla volta e soltanto nella direzione di gioco indicata dalle frecce (quindi, verso il basso per le pedine nere). Quando una pedina raggiunge una delle caselle dell'ultima riga viene promossa e diventa damone. **Per semplicità, si considerino esclusivamente i movimenti sulle diagonali che, dall'alto, vanno verso sinistra.** Per esempio, nell'immagine seguente le uniche pedine nere che possono fare una mossa valida verso sinistra sono quelle sulla riga 2, colonna 2 e riga 2, colonna 6. Non si consideri la possibile mossa di mangiata delle pedine bianche.



Soluzione

//1.1

```
Scacchiera s;
int i, j,
// le variabili che seguono servono per gli altri punti dell'esercizio
int start, r, c, tr, tc;
for (i=0; i<8; i++)
    for (j=0; j<8; j++)
    {
        s[i][j].occupata = falso;
        if (i % 2 == 0)
            if(j % 2 == 0)
                s[i][j].c = nero;
            else
                s[i][j].c = bianco;
        else if(j % 2 == 1)
            s[i][j].c = nero;
        else
            s[i][j].c = bianco;
    }
```

//1.2

```
// riempimento della parte superiore della scacchiera
for (i=0; i<3; i++)
{
    if (i % 2 == 0)
        start = 0;
    else start = 1;
    /*start serve per identificare la posizione da cui inserire le
    caselle nere */

    for (j=start; j<8; j=j+2)
    {
        s[i][j].occupata = vero;
        s[i][j].p.c = nero;
        s[i][j].p.damone = falso;
    }
}
```

//Soluzione alternativa

```
for(i=0; i<3; i++)
    for(j=0; j<8; j++)
        if(s[i][j].c==nero)
        {
            s[i][j].occupata = vero;
            s[i][j].p.c = nero;
            s[i][j].p.damone = falso;
        }
```

//1.3

```
r = 8; c = 8;
while(!(r>=0 && r<8) || !(c>=0 && c<8))
{
    printf("Inserisci il numero di riga e colonna del pezzo: ");
    scanf("%d %d", &r, &c);
}
if(s[r][c].occupata && s[r][c].p.c == nero)
{
    if(r+1<8 && c-1>=0 && !s[r+1][c-1].occupata)
    {
        s[r+1][c-1].p = s[r][c].p;
```

```
s[r+1][c-1].occupata = vero;
if(r+1 == 7)
    s[r+1][c-1].p.damone = vero;
s[r][c].occupata = falso;
printf("Movimento: pedina in (%d, %d)", r+1, c-1);
}
else printf("La pedina non puo` muoversi\n");
}
else printf("La posizione selezionata non contiene una pedina nera\n");
```

Esercizio 2 (10 punti)

Si consideri un sistema per monitorare gli accessi dei veicoli nel centro di una città (si pensi all'Area C di Milano). Il sistema, che è attivo ogni giorno dalle 7.00 alle 19.00, identifica i veicoli che entrano nell'area controllata attraverso dei varchi in cui è installato un sistema di riconoscimento delle targhe dei veicoli. Ogni volta che un nuovo veicolo passa attraverso il varco, il sistema memorizza l'ingresso dell'auto tenendo traccia del numero di targa e dell'orario del passaggio. Si risponda alle seguenti domande utilizzando il linguaggio Matlab.

Domanda 2.1 (punti 3)

Si scriva una funzione **calcolaNumeroAccessiOrari** che, data la matrice degli accessi giornalieri, restituisce un array di 12 valori, uno per ogni ora tra le 7.00 e le 19.00, che rappresentano il numero di accessi avvenuti nell'ora corrispondente. La matrice degli accessi giornalieri è organizzata in modo che in ogni riga contenga i dati di un accesso, cioè, nella prima colonna il numero di targa dell'auto che è entrata nell'area (per semplicità, consideriamo le targhe come valori puramente numerici) e nella seconda colonna un orario di ingresso **in minuti, a partire dalle 7.00 di ogni giorno**. La funzione, quindi, analizza la matrice degli accessi giornalieri e calcola il numero di accessi per ogni ora tra le 7.00 e le 19.00 della giornata considerata. Nello sviluppare la funzione, si tenga presente che gli elementi nella matrice non sono ordinati temporalmente.

Domanda 2.2 (punti 3)

Si scriva la funzione **calcolaPicco** che, dato l'array con il numero di accessi per ora, restituisce l'ora o le ore in cui il numero di accessi ha raggiunto il picco, il numero di accessi medio nella giornata e l'ora o le ore con il minor numero di accessi.

Domanda 2.3 (punti 4)

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

- Acquisisca da un file binario **accessiAreaC** due matrici di accessi giornalieri **giorno1** e **giorno2**, relativi a due diversi giorni della settimana.
- Verifichi che tutti i dati siano coerenti in termini di struttura delle matrici (devono essere matrici costituite da due colonne e un numero qualsiasi di righe) ed elimini i dati non corretti, in cui, cioè, l'orario di ingresso non corrisponde a un numero di minuti compreso tra 0 e il numero di minuti che intercorrono tra le 7 e le 19.
- Per ognuna delle matrici recuperate dal file, calcoli, tramite la funzione **calcolaNumeroAccessiOrari**, il numero di accessi per ciascuna ora della giornata e poi, tramite la funzione **calcolaPicco**, l'ora di picco.
- Crei un plot che mostri, con linee diverse, il numero degli accessi per ora al centro città nelle giornate considerate.

Soluzione

2.1

```
function [nAccessiOrari] = calcolaNumeroAccessiOrari(accessi)

    nAccessiOrari = zeros(1,12);
    orari = accessi(:, 2)';
    for ii=1:12
        nAccessiOrari(ii) = sum(orari<=ii*60 & orari>(ii-1)*60);
    end
end
```

2.2

```
function [oraPicco, mediaAccessi, oraMinimo] = calcolaPicco(nAccessi)
    oraPicco = find(nAccessi==max(nAccessi))+6;
    oraMinimo = find(nAccessi==min(nAccessi))+6;
    mediaAccessi = mean(nAccessi);
end
```

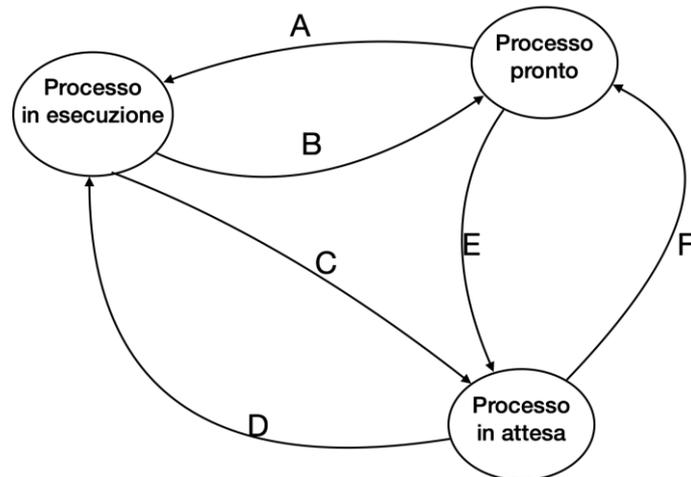
2.3

```
load accessiAreaC giorno1 giorno2

if(size(giorno1, 2) == 2 && size(giorno2, 2) == 2)
    indexes = giorno1(:, 2) < 0 | giorno1(:, 2) > 12*60;
    giorno1(indexes, :) = [];
    indexes = giorno2(:, 2) < 0 | giorno2(:, 2) > 12*60;
    giorno2(indexes, :) = [];
    nA_g1 = calcolaNumeroAccessiOrari(giorno1);
    picco_g1 = calcolaPicco(nA_g1);
    nA_g2 = calcolaNumeroAccessiOrari(giorno2);
    picco_g2 = calcolaPicco(nA_g2);
    plot(nA_g1, 'g');
    hold on
    plot(nA_g2, 'b');
    hold off
end
```

Esercizio 3 (6 punti)

Si consideri il seguente diagramma che vuole rappresentare gli stati di un processo e i relativi passaggi di stato. Per ogni freccia da A ad F, si dica se corrisponde a un passaggio di stato corretto per un processo. Si motivi il perché della presenza o assenza di ognuno dei passaggi di stato. **Risposte senza motivazione non verranno considerate nella valutazione.**



Domanda 1 (3 punti)

Si consideri la seguente funzione in linguaggio Matlab:

```
1. function G = cut(M)
2. if (length(M) <= 5)
3.     G = M;
4.     disp(G(5,12));
5.     G(4:5,2:3) = [];
6. end
```

Si considerino le seguenti due chiamate alla funzione:

```
x = cut([1 2 5 6 7 8; 3 4 5 6 7 8]);
y = cut([1, 2, 3; 4 5 6; 7 8 9; 6 7 8; 7 6 5]);
```

Si spieghi il comportamento della funzione nei due casi e si dica se vengono rilevati errori nell'esecuzione e i motivi associati a ciascuno degli errori che si verificano.

Soluzione

Nella prima chiamata la condizione dell'if è falsa e la funzione termina senza aver prodotto un valore di ritorno nella variabile G. Questo viene segnalato come un errore dall'interprete.

Nella seconda chiamata la condizione `length(M) <= 5` è vera, questo significa che tutte le dimensioni di M sono inferiori a 5. Di conseguenza, l'elemento di posizione 5, 12 in G e quindi in M non può esistere (riga 4). Anche la riga 5 contiene un errore perché si cerca di eliminare due valori in ciascuna delle righe 4 e 5 lasciando inalterato il numero di valori contenuti nelle altre righe. Quest'asimmetria non è possibile e quindi l'interprete segnala un errore.

Domanda 2 (3 punti)

Per risolvere uno stesso problema si hanno a disposizione due differenti programmi, A e B. Il programma A richiede una grande quantità di spazio in memoria per essere eseguito, ma viene eseguito in un tempo molto breve. Il programma B, invece, ha una richiesta di spazio in memoria ridotto, ma la sua esecuzione richiede un tempo molto lungo. Si risponda alle seguenti domande:

- Si supponga che il problema da risolvere sia il monitoraggio in tempo reale dei parametri di un reattore nucleare (Flusso Neutronico, Livelli di Radioattività, Pressione) per segnalare eventuali anomalie. Le anomalie devono essere segnalate entro pochi millisecondi. Quale dei due programmi è da preferirsi? Motivare la risposta.
- Si supponga ora che il problema da risolvere sia la risoluzione di calcoli scientifici intensivi (ad esempio simulazioni aerodinamiche per macchine sportive ad alte prestazioni). Quale dei due programmi è da preferirsi? Motivare la risposta.