

Nome _____ Cognome _____ Matricola _____ Corso _____ Aula _____ Posizione nell'aula _____ →	<table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th colspan="7">cattedra</th> </tr> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>...</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>A1</td> <td>B1</td> <td>C1</td> <td>D1</td> <td>E1</td> <td>...</td> </tr> <tr> <td>2</td> <td>A2</td> <td>B2</td> <td>C2</td> <td>D2</td> <td>E2</td> <td>...</td> </tr> <tr> <td>3</td> <td>A3</td> <td>B3</td> <td>C3</td> <td>D3</td> <td>E3</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> </tbody> </table>	cattedra								A	B	C	D	E	...	1	A1	B1	C1	D1	E1	...	2	A2	B2	C2	D2	E2	...	3	A3	B3	C3	D3	E3
cattedra																																											
	A	B	C	D	E	...																																					
1	A1	B1	C1	D1	E1	...																																					
2	A2	B2	C2	D2	E2	...																																					
3	A3	B3	C3	D3	E3	...																																					
...																																					

L'emettitore di biglietti a banda magnetica dedicato all'utenza oraria e posto all'ingresso di House of Cars ha un pulsante per la richiesta del biglietto e una stampante per la sua emissione. Inoltre, ha un pulsante di assistenza, per utenti portatori di handicap che necessitano di assistenza al parcheggio. L'assistenza deve essere richiesta dopo aver richiesto il biglietto e prima di ritirarlo.

Si consideri il caso d'uso **Ingresso al parcheggio**, avviato dalla richiesta di biglietto e specificato da:
Precondizione: Esiste un posto libero. **Postcondizione:** Posto auto individuato; biglietto ritirato; colonna segnalata; sbarre colonna aperte; se assistenza richiesta, addetto avvisato.

Per la progettazione architetturale del sottosistema **Ingresso**, che realizza il caso d'uso **Ingresso al parcheggio**, sono state individuate le seguenti componenti:

Interfaccia Utente	Realizza l'interfaccia utente presso l'emettitore di biglietti
Interfaccia Addetto	Realizza l'interfaccia dell'addetto
Gestione Ingresso	Realizza la business logic del sottosistema per la ricerca del posto e l'eventuale invio di richiesta assistenza all'addetto
DB Posti	Mantiene le informazioni sui box del parcheggio
Gestione Devices	Realizza le funzionalità di segnalazione colonna e apertura sbarre.

Domanda 1. a) Si dia una vista C&C dell'architettura del sottosistema **Ingresso**, decorando i connettori con l'indicazione del ruolo delle componenti coinvolte. b) Si specifichi l'interfaccia che *Gestione Ingresso* offre/richiede per comunicare con *Interfaccia Utente*

Domanda 2. Si fornisca un diagramma di sequenza che mostra le interazioni cui partecipano gli attori e le componenti individuate per realizzare il caso d'uso **Ingresso al parcheggio**.

Domanda 3. Si definisca la struttura interna della componente **Gestione Ingresso**, disegnando un diagramma di struttura composita. Si includa anche una parte col ruolo di creatore della componente e si scriva lo pseudocodice del metodo `creaGestioneIngresso(...)`.

Domanda 4. Il metodo

```
public boolean isMCD(int a, int b, int mcd) throws IllegalArgumentException
restituisce true se mcd è il massimo comune divisore di a e b, false altrimenti e lancia un'eccezione se uno dei parametri è non positivo.
```

a) Specificare alcuni casi input secondo il criterio di copertura delle classi di equivalenza e dei valori di frontiera

a	b	mcd	Motivazione

b) Specificare alcuni casi input per minimizzi le combinazioni possibili, tenendo conto degli errori

a	b	mcd

c) Proporre un insieme di casi di input basato sul catalogo dettato dalla propria esperienza, spiegandone la motivazione.

a	b	mcd	Motivazione

Domanda 5. Si consideri la seguente implementazione

```

1) public boolean isMCD(int a, int b, int mcd) {
2)   if (a<=0 || b<=0)
3)       throw new IllegalParameter();
4)   int min = a<b ? a : b;
5)   if ( (a % mcd != 0) | (b % mcd != 0) )
6)       return false;
7)   for (int i = mcd + 1; i < min; i++) {
8)       if ( (a % i == 0) && (b % i == 0) )
9)           return false;
10)  }
11)  return true;
12) }
```

a) Fornire una batteria di test che garantisca la copertura dei comandi

a	b	mcd	output

b) Fornire una batteria di test che garantisca la copertura delle condizioni e delle decisioni

a	b	mcd	output

c) Individuare un mutante e un caso di test che lo uccide che permettano di evidenziare un difetto nel programma