

**GestAir**  
**Software per la gestione**  
**dell'aeroporto di**  
**Bari-Palese**

**Andrea De Rinaldis**  
**A.A. 2016-17**

# Indice:

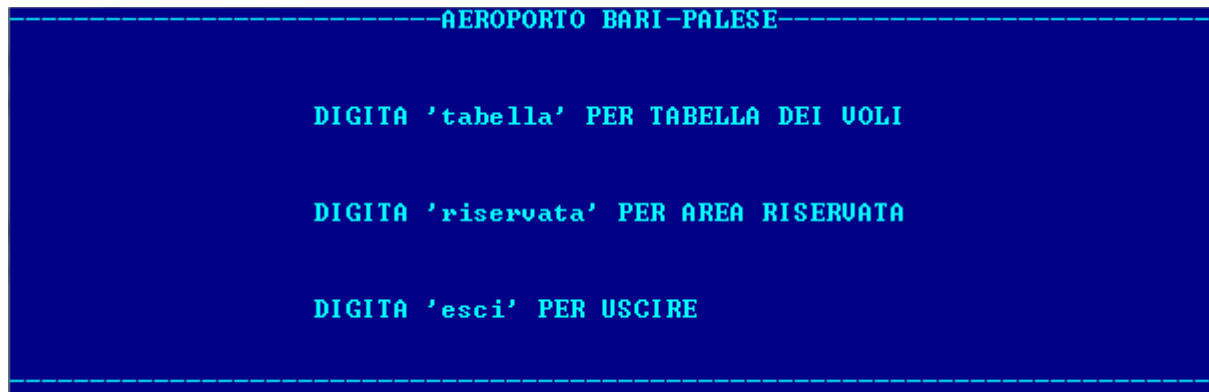
<b><u>Analisi</u></b> .....	<b>1</b>
- <u>Descrizione del sistema</u> .....	2
- <u>Requisiti Funzionali</u> .....	4
- <u>Casi d'uso</u> .....	5
- <u>Strumenti di sviluppo</u> .....	13
<b><u>Progettazione</u></b> .....	<b>14</b>
- <u>Progettazione dei tipi di dato e delle strutture dati</u> .....	15
- <u>Progettazione delle librerie e delle funzioni</u> .....	19
- <u>Dipendenza tra funzioni</u> .....	21
- <u>Pseudocodice</u> .....	24
<b><u>Codifica</u></b> .....	<b>51</b>
- <u>Documentazione Doxygen</u> .....	52
<b><u>Testing</u></b> .....	<b>53</b>
- <u>Definizione del piano di test</u> .....	54
- <u>Esiti del piano di test</u> .....	62
- <u>Pianificazione di miglioramenti</u> .....	63
<b><u>Dichiarazione di unicità del progetto</u></b>	

# **ANALISI DEL PROBLEMA**

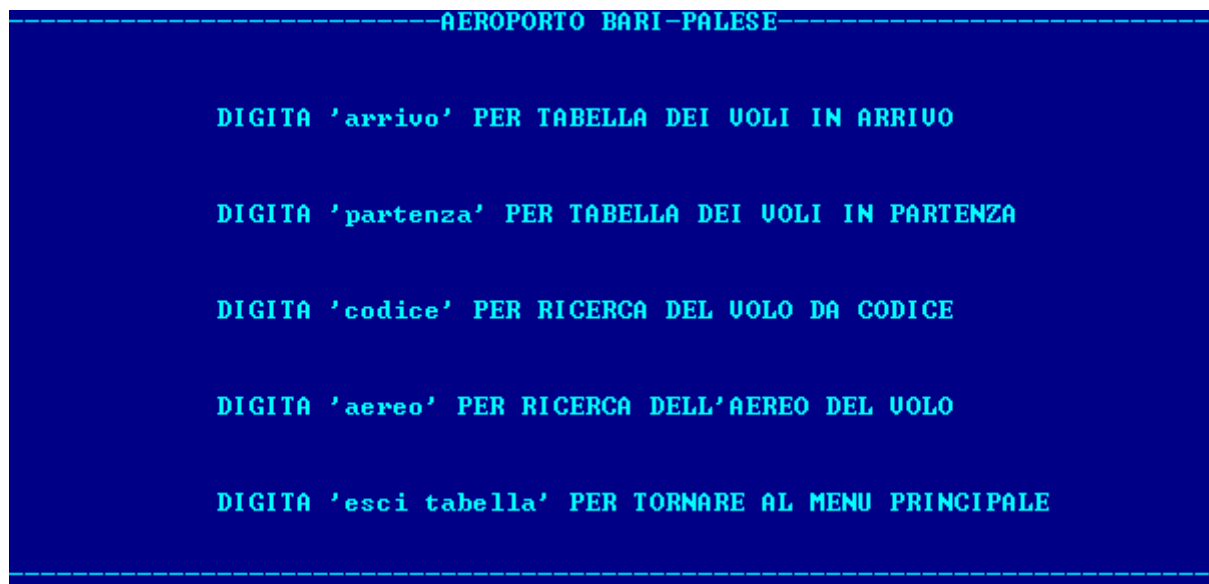
## - DESCRIZIONE DEL SISTEMA

Il sistema permette la gestione dei voli in transito dall'aeroporto di Bari-Palese.

In un primo momento, l'utente si troverà dinanzi ad un menù testuale dove potrà decidere cosa fare.



Se l'utente inserisse il termine *tabella*, potrà visualizzare i voli presenti nel database in arrivo ed in partenza dall'aeroporto, potrà stampare un volo partendo dal suo codice e potrà stampare l'aereo utilizzato dal volo partendo dal suo codice.



Inoltre, se l'utente è in possesso della password (fornita dall'amministratore del sistema), è permessa una gestione più approfondita dei database dell'aeroporto: infatti, egli potrà modificare il gate di approdo, cancellare, autorizzare ed inserire in emergenza i vari voli presenti in lista.<sup>3</sup>

-----AEROPORTO BARI-PALESE-----

DIGITA 'cancella' PER ELIMINAZIONE VOLO IN PARTENZA

DIGITA 'gate' PER CAMBIO DI GATE DEL VOLO

DIGITA 'emergenza' PER GESTIONE DELLE EMERGENZE

DIGITA 'autorizza' PER AUTORIZZAZIONE DEI VOLI

DIGITA 'esci riservata' PER TORNARE AL MENU PRINCIPALE

Il limite di tentativi di immissione della password è di tre errori, dopodichè il sistema bloccherà l'accesso all'area riservata per la sessione in corso.

LA CHIAVE D'ACCESSO E' COMPOSTA DA SOLE CIFRE. L'IMMISSIONE DI UN  
CARATTERE COMPORTA IL BLOCCO PER L'AREA RISERVATA PER TALE SESSIONE.  
INSERISCI LA CHIAVE D'ACCESSO <3 TENTATIVI>.  
INSERIMENTO:

Alla fine dell'uso del programma, verrà cancellato il database per motivi di riservatezza e verrà creato un file di log con tutti i dati dei cambiamenti avvenuti durante la sessione.

GRAZIE PER AVER UTILIZZATO QUESTO PROGRAMMA!  
UN FILE DI LOG E' STATO CREATO NELLA CARTELLA log  
IL FILE voli.csv VERRA' ELIMINATO PER MOTIVI DI SICUREZZA.

## - REQUISITI FUNZIONALI

Di seguito sono elencate le funzionalità implementate nel software:

<b>CODICE</b>	<b>NOME</b>	<b>DESCRIZIONE</b>
<b>C1</b>	<b>Caricamento dei database</b>	Il programma deve caricare i diversi database presenti nelle directory. Nel particolare, caricherà i file contenenti i dati sui voli, sugli aerei dei voli, sui modelli di aerei e sugli aeroporti.
<b>C2</b>	<b>Caricamento della password</b>	Il programma deve caricare il valore della password, che è stato fornito dell'amministratore del sistema tramite un file apposito.
<b>M1</b>	<b>Visualizzazione del menù principale</b>	Il programma deve visualizzare il menù principale dove vengono rappresentate le possibili opzioni d'uso.
<b>M2</b>	<b>Visualizzazione del menù <i>tabella</i></b>	Il programma deve visualizzare il menù per la rappresentazione delle tabelle dei voli dove vengono rappresentate le possibili opzioni d'uso.
<b>M3</b>	<b>Visualizzazione del menù <i>riservata</i></b>	Il programma deve visualizzare il menù per la rappresentazione dell'area riservata dove vengono rappresentate le possibili opzioni d'uso.
<b>T1</b>	<b>Rappresentazione della tabella dei voli in partenza</b>	Il programma deve visualizzare la tabella di tutti i voli (ordinati in ordine decrescente per la data d'arrivo) in partenza dall'aeroporto di Bari-Palese.
<b>T2</b>	<b>Rappresentazione della tabella dei voli in uscita</b>	Il programma deve visualizzare la tabella di tutti i voli (ordinati in ordine decrescente per la data d'arrivo) in arrivo all'aeroporto di Bari-Palese.
<b>T3</b>	<b>Rappresentazione della tabella del volo ricercato per codice</b>	Il programma deve visualizzare la tabella del volo ricercato tramite l'inserimento del suo codice identificativo.
<b>T4</b>	<b>Rappresentazione della tabella dell'aereo ricercato per codice</b>	Il programma deve visualizzare la tabella dell'aereo del volo ricercato tramite l'inserimento del suo codice identificativo
<b>P1</b>	<b>Acquisizione della password</b>	Il programma deve acquisire da tastiera la password per l'area riservata.
<b>R1</b>	<b>Cancellazione del volo</b>	Il programma deve cancellare il volo ricercato tramite l'inserimento del suo codice identificativo.
<b>R2</b>	<b>Modifica del gate del volo</b>	Il programma deve modificare il gate del volo tramite l'inserimento del codice identificativo di quest'ultimo.
<b>R3</b>	<b>Gestione dei voli in emergenza</b>	Il programma deve aggiungere un volo in partenza tra quelli in emergenza e farlo tornare a Bari-Palese, tramite l'inserimento del suo codice identificativo.
<b>R4</b>	<b>Autorizzazione dei voli</b>	Il programma deve autorizzare un volo in partenza o arrivo tramite l'inserimento del suo codice identificativo.
<b>L1</b>	<b>Creazione del file di log</b>	Il programma deve cancellare il file dei voli e creare il file di log

## - CASI D'USO

Di seguito sono elencati i casi d'uso delle varie funzioni:

<b>CODICE</b>	C1	
<b>PRE-CONDIZIONI</b>	La funzione viene sempre effettuata.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	I vari dati sono stati caricati dai file .
	<b>Fallimento</b>	Mostra un messaggio di errore e chiude il programma.
<b>EVENTO INnescante</b>	Nessuno.	
<b>SCENARIO DI BASE</b>	Il sistema carica i dati relativi ai voli, agli aerei dei voli, ai modelli degli aerei ed agli aeroporti, continuando l'esecuzione sulla funzione C2.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema rileva l'assenza di uno dei file o il loro danneggiamento, mostra un messaggio di errore relativo al primo file trovato inutilizzabile (o non trovato) e chiude il programma.	

<b>CODICE</b>	C2	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata a patto che la funzione C1 vada a buon fine.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	La password è stata caricata dal file
	<b>Fallimento</b>	Mostra un messaggio di errore e chiude il programma.
<b>EVENTO INnescante</b>	La funzione C1 deve essere andata a buon fine.	
<b>SCENARIO DI BASE</b>	Il sistema carica la password dal file e continua l'esecuzione entrando nella schermata del menù di base.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema rileva l'assenza del file o il suo danneggiamento, mostra un messaggio di errore relativo al file inutilizzato (o non trovato) e chiude il programma.	

<b>CODICE</b>	M1	
<b>PRE-CONDIZIONI</b>	Corretta riuscita delle funzioni C1 e C2.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema mostra uno tra i due menù più avanzati del programma o esce dal programma a seconda della scelta effettuata.
	<b>Fallimento</b>	Il sistema continua a mostrare il menù del programma fino a che non viene inserito l'opzione corretta.
<b>EVENTO INnescante</b>	Le funzioni C1 e C2 devono essere andate entrambe a buon fine.	
<b>SCENARIO DI BASE</b>	Il sistema mostra il menù di base del programma con le possibili opzioni. Se ciò che inserisce l'utente è incluso tra le opzioni allora carica un altro menù.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema rileva che la scelta dell'utente non è corretta e mostra nuovamente lo stesso menu, fino a che non viene inserita una scelta inclusa tra le opzioni.	

<b>CODICE</b>	M2	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù iniziale si scrive la parola <i>tabella</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema entra nella visualizzazione di una tipologia di tabella o esce dal menù <i>tabella</i> a seconda della scelta effettuata.
	<b>Fallimento</b>	Il sistema continua a mostrare il menù <i>tabella</i> fino a che non viene inserita l'opzione corretta.
<b>EVENTO INnescante</b>	L'inserimento della stringa <i>tabella</i> in M1.	
<b>SCENARIO DI BASE</b>	Il sistema mostra il menu <i>tabella</i> per permettere all'utente di scegliere in quale tra le diverse categorie entrare.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema continua a mostrare lo stesso menù <i>tabella</i> fino a che non viene inserita una corretta scelta in base alle opzioni presenti.	



<b>CODICE</b>	M3	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù iniziale si scrive la parola <i>riservata</i> e, contemporaneamente, la password inserita è corretta.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema entra in una delle categorie dell'area riservata o esce a seconda della scelta effettuata.
	<b>Fallimento</b>	Il sistema continua a mostrare il menu riservata fino a che non viene inserita l'opzione corretta.
<b>EVENTO INNESCANTE</b>	L'inserimento della stringa <i>riservata</i> in M1(per la chiamata del menù) e l'inserimento della password corretta in P1(per lo sblocco dell'area riservata)	
<b>SCENARIO DI BASE</b>	Il sistema mostra il menu <i>riservata</i> per permettere all'utente di scegliere in quale tra le diverse categorie entrare.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema continua a mostrare lo stesso menù <i>riservata</i> fino a che non viene inserita una corretta scelta in base alle opzioni presenti.	

<b>CODICE</b>	T1	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menu <i>tabella</i> viene inserita la parola <i>arrivo</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema stampa a video i voli in ordine decrescente di arrivo, uno alla volta. Se si inserisce la parola <i>prossimo</i> si continua con la stampa di altri voli, se, invece, si inserisce la parola <i>esci tabella</i> si esce dalla visualizzazione dei voli in arrivo
	<b>Fallimento</b>	Il sistema continua a mostrare lo stesso volo finchè la scelta inserita non sarà uguale ad una delle due opzioni
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>arrivo</i> in M2.	
<b>SCENARIO DI BASE</b>	Il sistema mostra all'utente i diversi voli che sono in arrivo all'aeroporto di Bari-Palese.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema mostra all'utente sempre lo stesso volo se la scelta effettuata non è corretta.	

<b>CODICE</b>	T2	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù <i>tabella</i> viene inserita la parola <i>partenza</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema stampa a video i voli in ordine decrescente di arrivo, uno alla volta. Se si inserisce la parola <i>prossimo</i> si continua con la stampa di altri voli, se, invece, si inserisce la parola <i>esci tabella</i> si esce dalla visualizzazione dei voli in arrivo
	<b>Fallimento</b>	Il sistema continua a mostrare lo stesso volo finchè la scelta inserita non sarà uguale ad una delle due opzioni
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>arrivo</i> in M2.	
<b>SCENARIO DI BASE</b>	Il sistema mostra all'utente i diversi voli che sono in arrivo all'aeroporto di Bari-Palese.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema mostra all'utente sempre lo stesso volo se la scelta effettuata non è corretta.	

<b>CODICE</b>	T3	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù <i>tabella</i> viene inserita la parola <i>codice</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice da ricercare inserito dall'utente e quello dei diversi voli presenti nel database. Se il volo viene trovato allora si stampano le informazioni relative ad esso.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice immesso nell'utente e quello relativo di tutti i voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>codice</i> in M2.	
<b>SCENARIO DI BASE</b>	Il sistema mostra all'utente il volo con il codice identificativo corrispondente a quello immesso da tastiera.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito e il codice identificativo di ciascun volo.	

<b>CODICE</b>	T4	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù <i>tabella</i> viene inserita la parola <i>aereo</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice da ricercare inserito dall'utente e quello dei diversi voli presenti nel database. Se il volo viene trovato allora si stampano le informazioni relative al veicolo utilizzato per il volo.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice immesso nell'utente e quello relativo di tutti i voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>aereo</i> in M2.	
<b>SCENARIO DI BASE</b>	Il sistema mostra all'utente l'aereo utilizzato nel volo con il codice identificativo corrispondente a quello immesso da tastiera.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito e il codice identificativo di ciascun volo.	

<b>CODICE</b>	P1	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù di base viene inserita la parola <i>riservata</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema richiede di inserire la password dell'area riservata fino ad un massimo di tre tentativi utili. Se la password è corretta allora viene permesso all'utente di accedere all'area riservata.
	<b>Fallimento</b>	Se la password non è corretta, viene chiesto nuovamente all'utente di reinserirla oppure, se i tre tentativi sono ormai scaduti, blocca l'accesso all'area riservata per la sessione in corso.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>riservata</i> in M1.	
<b>SCENARIO DI BASE</b>	Il sistema, per accedere all'area riservata, mette a disposizione tre tentativi per immettere una password corretta. Se l'immissione è andata a buon fine allora all'utente è permesso entrare nell'area riservata.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema blocca l'area riservata per la sessione in corso.	

<b>CODICE</b>	R1	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù di base viene inserita la parola <i>cancella</i> e la password inserita è corretta.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice del volo inserito e quello di ciascun volo in partenza. Se la corrispondenza esiste allora cancella il volo.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice inserito dall'utente e quello dei diversi voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>cancella</i> in M3.	
<b>SCENARIO DI BASE</b>	Il sistema mostra tutti i voli in partenza da Bari-Palese. L'utente inserirà il codice del volo da cancellare e, se esiste una corrispondenza tra esso e il codice di un volo, allora potrà cancellarlo.	
<b>SCENARIO ALTERNATIVO</b>	<p>Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito dall'utente e il codice identificativo di ciascun volo.</p> <p>Il sistema mostra un messaggio di errore se il volo è già stato cancellato.</p>	

<b>CODICE</b>	R2	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù di base viene inserita la parola <i>gate</i> e la password inserita è corretta.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice del volo inserito e quello di ciascun volo. Se la corrispondenza esiste allora modifica il gate del volo.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice inserito dall'utente e quello dei diversi voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>gate</i> in M3.	
<b>SCENARIO DI BASE</b>	Il sistema mostra tutti i voli in transito da Bari-Palese. L'utente inserirà il codice del volo il cui gate deve essere modificato e, se esiste una corrispondenza tra esso e il codice di un volo, allora potrà modificarlo.	
<b>SCENARIO ALTERNATIVO</b>	<p>Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito dall'utente e il codice identificativo di ciascun volo.</p> <p>Il sistema mostra un messaggio di errore se il gate inserito non è valido oppure è lo stesso.</p>	

<b>CODICE</b>	R3	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù di base viene inserita la parola <i>emergenza</i> e la password inserita è corretta.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice del volo inserito e quello di ciascun volo in arrivo. Se la corrispondenza esiste allora metti in emergenza il volo.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice inserito dall'utente e quello dei diversi voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>emergenza</i> in M3.	
<b>SCENARIO DI BASE</b>	Il sistema permette all'utente l'inserimento del codice del volo in partenza da inserire in emergenza. Se esiste una corrispondenza tra il codice inserito da tastiera e il codice del volo, il volo viene inserito in emergenza spostandolo tra i voli in arrivo e ottenendo il tempo che intercorreva tra la partenza e il possibile arrivo all'aeroporto.	
<b>SCENARIO ALTERNATIVO</b>	<p>Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito dall'utente e il codice identificativo di ciascun volo.</p> <p>Il sistema mostra un messaggio di errore se il volo è già in arrivo.</p> <p>Il sistema mostra un messaggio di errore se il volo è stato cancellato.</p>	

<b>CODICE</b>	R4	
<b>PRE-CONDIZIONI</b>	La funzione viene effettuata quando nel menù di base viene inserita la parola <i>autorizza</i> e la password inserita è corretta.	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema ricerca una corrispondenza tra il codice del volo inserito e quello di ciascun volo in transito. Se la corrispondenza esiste allora autorizza il volo.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non aver trovato una corrispondenza tra il codice inserito dall'utente e quello dei diversi voli.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>autorizza</i> in M3.	
<b>SCENARIO DI BASE</b>	Il sistema permette all'utente l'inserimento del codice del volo in partenza da inserire tra i voli autorizzati. Se esiste una corrispondenza tra il codice inserito da tastiera e il codice del volo, il volo sarà inserito tra quelli autorizzati, calcolando un possibile ritardo o anticipo del volo e stampando a video la lista di alcuni voli ordinati in ordine crescente in base all'ora di arrivo.	

<b>SCENARIO ALTERNATIVO</b>	<p>Il sistema mostra un messaggio di errore se non è stata trovata una corrispondenza tra il codice inserito dall'utente e il codice identificativo di ciascun volo.</p> <p>Il sistema mostra un messaggio di errore se il volo è già stato autorizzato.</p> <p>Il sistema mostra un messaggio di errore se il volo è stato cancellato.</p>
-----------------------------	---

<b>CODICE</b>	L1	
<b>PRE-CONDIZIONI</b>	Il file deve essere chiuso tramite l'inserimento della parola <i>esci</i> .	
<b>POST-CONDIZIONI</b>	<b>Successo</b>	Il sistema cancella il database dei voli e crea un file di log in cui saranno presenti tutte le modifiche che sono state effettuate durante la sessione.
	<b>Fallimento</b>	Il sistema mostra un messaggio di errore per non esser riuscito a creare il file di log.
<b>EVENTO INNESCANTE</b>	L'inserimento della parola <i>esci</i> in M1.	
<b>SCENARIO DI BASE</b>	Il sistema, prima della chiusura del programma, cancella il database dei voli (per motivi di riservatezza) e crea un file di log nel quale saranno annotati tutti i diversi cambiamenti che sono stati effettuati all'interno del programma nel corso della sessione.	
<b>SCENARIO ALTERNATIVO</b>	Il sistema mostra un messaggio di errore nel quale afferma di non esser riuscito a creare il file di log.	

## **- STRUMENTI DI SVILUPPO**

### Sistema di sviluppo

Il sistema è stato realizzato utilizzando un notebook Olivetti Olibook S1400 con processore Intel Core i3 da 2.13 Ghz, con 2GB di RAM.

### Linguaggio di programmazione

Il linguaggio di programmazione utilizzato è il C.

### IDE e plugin

L'ambiente di sviluppo utilizzato è stato Eclipse, corredato dai plugin CUnit (per la gestione del test delle funzioni) e Doxygen (per la documentazione del codice).

### Requisiti minimi

Per eseguire il sistema è consigliato avere un elaboratore con almeno queste caratteristiche:

- Sistemi: Windows 7 o superiori / Linux
- Processore: Intel Pentium 4 o superiori
- RAM: 256 MB o superiori

### Guide e siti esterni

I siti [cplusplus](#) e [Stack Overflow](#) sono stati particolarmente utili nelle situazioni complicate e in presenza di piccoli accorgimenti all'interno del programma.

# **PROGETTAZIONE**



## - PROGETTAZIONE DEI TIPI DI DATO E DELLE STRUTTURE DATI

Di seguito, sono mostrati i tipi di dati utilizzate nell'ambito del progetto:

NOME	TIPO	DESCRIZIONE	VALORI / CAMPI
string	Array di char	Astrazione dell'array di caratteri definito per formalizzare le stringhe	char
_AEROMOBILE	Struct	Tipo di dato definito per descrivere le componenti identificative di un aereo	modello (string) targa (string)
_AEROPORTO	Struct	Tipo di dato definito per descrivere le componenti fondamentali di un aeroporto	codice (string) nome (string)
_DATA	Struct	Tipo di dato definito per descrivere le componenti che costituiscono il formato della data	minuto (int) ora (int) giorno (int) mese (int) anno (int) data_estesa (double)
_MODELLO	Struct	Tipo di dato definito per descrivere le componenti fondamentali di un modello di aereo	azienda (string) modello (string) propulsore (char) numero_motori (int) numero_massimo_passeggeri (int)
_VOLO	Struct	Tipo di dato definito per descrivere le componenti fondamentali di un volo	codice_id (string) nome_compagnia (string) targa (string) partenza_arrivo (char) codice_iata (string) partenza (_DATA) arrivo (_DATA) effettiva (_DATA) numero_passeggeri (int) gate (string) cancellato (char) emergenza (char) autorizzato (char)

Di seguito, sono mostrati le variabili, le costanti e le strutture dati fondamentali nell'ambito del progetto:

<b>NOME</b>	<b>TIPO</b>	<b>DESCRIZIONE</b>	<b>VALORI / CAMPI</b>
DIMENSIONE_NOME_AZIENDA_COSTRUTTRICE	Costante intera	Costante per la definizione della dimensione del nome dell'azienda costruttrice degli aerei.	20
DIMENSIONE_NOME_MODELLO_AEREO	Costante intera	Costante per la definizione della dimensione del nome del modello dell'aereo.	20
NUMERO_LETTERE_TARGA	Costante intera	Costante per la definizione della dimensione della targa dell'aereo	6
DIMENSIONE_CODICE_VOLO	Costante intera	Costante per la definizione della dimensione del codice di volo.	6
DIMENSIONE_NOME_COMPAGNIA_AEREA	Costante intera	Costante per la definizione della dimensione del nome della compagnia aerea	20
DIMENSIONE_NOME_GATE	Costante intera	Costante per la definizione della dimensione del nome del gate.	2
DIMENSIONE_CODICE_IATA	Costante intera	Costante per la definizione della dimensione del codice IATA di un volo.	3
MAX_NUMERO_MODELLI	Costante intera	Costante per la definizione del massimo numero di modelli di aerei.	200
MAX_NUMERO_AEROMOBILI	Costante intera	Costante per la definizione del massimo numero di aeromobili.	200
MAX_NUMERO_VOLI	Costante intera	Costante per la definizione del massimo numero di voli.	200
MAX_NUMERO_AEROPORTI	Costante intera	Costante per la definizione del massimo numero di aeroporti di arrivo-partenza.	200
DIMENSIONE_NOME_AEROPORTO	Costante intera	Costante per la definizione della dimensione del nome dell'aeroporto.	25

DIMENSIONE_STRINGA_MENU	Costante intera	Costante per la definizione della dimensione della stringa del menu.	30
modello	Array di _MODELLO	Struttura dati per la memorizzazione dei dati del modello di aereo.	azienda (string) modello (string) propulsore (char) numero_motori (int) numero_massimo_passeggeri (int)
aereo	Array di _AEROMOBILE	Struttura dati per la memorizzazione dei dati dell'aereo.	modello (string) targa (string)
volo	Array di _VOLO	Struttura dati per la memorizzazione dei dati del volo.	codice_id (string) nome_compagnia (string) targa (string) partenza_arrivo (char) codice_iata (string) partenza (_DATA) arrivo (_DATA) effettiva (_DATA) numero_passeggeri (int) gate (string) cancellato (char) emergenza (char) autorizzato (char)
aeroporto	Array di _AEROPORTO	Struttura dati per la memorizzazione dei dati dell'aeroporto.	codice (string) nome (string)
numero_modelli	Intero	Variabile per la memorizzazione del numero di modelli salvati	int
numero_aerei	Intero	Variabile per la memorizzazione del numero di aerei salvati	int
numero_voli	Intero	Variabile per la memorizzazione del numero di voli salvati	int
numero_aeroporti	Intero	Variabile per la memorizzazione del numero di aeroporti salvati	int
password	Intero esteso	Variabile per la memorizzazione della password da file	long int

password_inserita	Intero esteso	Variabile per la memorizzazione della password da tastiera	long int
tentativi	Intero	Variabile di conteggio del numero di tentativi rimasti per l'immissione della password	int
scelta_menu	Stringa	Variabile per il controllo della scelta del menu immessa da tastiera	string
voli.csv	File	File contenente i dati relativi ai voli	--
modelli.csv	File	File contenente i dati relativi ai modelli	--
aeromobili.csv	File	File contenente i dati relativi agli aerei	--
aeroporti.csv	File	File contenente i dati relativi agli aeroporti	--
log_{orario_sistema}.csv	File	File contenente i dati relativi alle modifiche effettuate durante la sessione	--
key.bin	File	File contenente i dati relativi alla password	--

## - PROGETTAZIONE DELLE LIBRERIE E DELLE FUNZIONI

Le librerie utilizzate nel progetto sono le seguenti:

- main.h
- gestione\_aeroporto.h
- gestione\_dati.h
- funzioni\_varie.h
- test.h

In main.h non sono inclusi prototipi di funzione.

In essa vi sono le dichiarazioni delle costanti e delle astrazioni dei nuovi tipi creati.

Essa è stata creata al fine di gestire le strutture dati più utilizzate all'interno del progetto in modo da renderle accessibili ai diversi moduli di quest'ultimo.

In gestione\_aeroporto.h sono inclusi i seguenti prototipi delle funzioni:

- stampa\_lista\_ordinata\_voli\_in\_partenza ()
- stampa\_lista\_ordinata\_voli\_in\_arrivo ()
- mergesort\_data\_per\_arrivo ()
- merge\_data\_per\_arrivo ()
- mergesort\_data\_per\_partenza ()
- merge\_data\_per\_partenza ()
- riconoscimento\_aeroporto ()
- stampa\_volo\_ricerca\_per\_codice ()
- stampa\_aereo\_ricerca\_per\_codice ()
- cancellazione\_volo ()
- modifica\_gate ()
- gestione\_emergenze ()
- autorizzazione\_voli ()

Essa è stata creata al fine di gestire tutte le funzioni che prevedono la gestione e la manipolazione dei dati relativi ai voli ed alle altre strutture dati del programma.

In gestione\_dati.h sono inclusi i seguenti prototipi delle funzioni:

- caricamento\_info\_modelli\_aerei ()
- caricamento\_info\_aeromobili ()
- caricamento\_info\_voli ()
- caricamento\_info\_aeroporti ()
- acquisizione\_password ()
- creazione\_log ()

Essa è stata creata al fine di gestire tutte le funzioni che prevedono il caricamento dei vari dati da file esterni per la memorizzazione nelle varie strutture dati del programma.

In funzioni\_varie.h sono inclusi i seguenti prototipi delle funzioni:

- settaggio\_schermo ()
- stampa\_menu\_iniziale ()
- converti\_minuscolo ()
- converti\_maiuscolo ()
- stampa\_menu\_tabella\_voli ()
- stampa\_menu\_area\_riservata ()
- cambio\_data\_avanti ()

- cambio\_data\_indietro ( )

Essa è stata creata al fine di gestire tutte le funzioni che non sono previste negli altri file header e non dipendono da un particolare criterio di categorizzazione comune.

In test.h sono inclusi i seguenti prototipi delle funzioni:

- init\_suite ( )
- clean\_suite ( )
- test\_cambio\_data\_avanti ( )
- test\_cambio\_data\_indietro ( )
- test\_caricamento\_info\_modelli\_aerei ( )
- test\_caricamento\_info\_aeromobili ( )
- test\_caricamento\_info\_voli ( )
- test\_caricamento\_info\_aeroporti ( )
- test\_acquisizione\_password ( )
- test\_mergesort\_data\_per\_arrivo ( )
- test\_mergesort\_data\_per\_partenza ( )
- test\_riconoscimento\_aeroporto ( )

Essa è stata creata al fine di gestire tutte le funzioni che vengono usate per la gestione dei test dei vari metodi.

## - DIPENDENZA TRA FUNZIONI

Di seguito, sono mostrate le dipendenze tra le diverse funzioni:

- stampa\_lista\_ordinata\_voli\_in\_partenza ( ) : Dipendente dalle funzioni:
  - mergesort\_data\_per\_arrivo ( )
  - converti\_minuscolo ( )
- stampa\_lista\_ordinata\_voli\_in\_arrivo ( ) : Dipendente dalle funzioni:
  - mergesort\_data\_per\_arrivo ( )
  - converti\_minuscolo ( )
- mergesort\_data\_per\_arrivo ( ) : Dipendente dalla funzione merge\_data\_per\_arrivo ( )
- merge\_data\_per\_arrivo ( ) : Non dipendente da altre funzioni
- mergesort\_data\_per\_partenza ( ) : Dipendente dalla funzione merge\_data\_per\_partenza ( )
- merge\_data\_per\_partenza ( ) : Non dipendente da altre funzioni
- riconoscimento\_aeroporto ( ) : Non dipendente da altre funzioni
- stampa\_volo\_ricerca\_per\_codice ( ) : Dipendente dalle funzioni:
  - converti\_minuscolo ( )
  - converti\_maiuscolo ( )
  - riconoscimento\_aeroporto ( )
- stampa\_aereo\_ricerca\_per\_codice ( ) : Dipendente dalle funzioni:
  - converti\_minuscolo ( )
  - converti\_maiuscolo ( )
- cancellazione\_volo ( ) : Dipendente dalle funzioni:
  - mergesort\_data\_per\_partenza ( )
  - riconoscimento\_aeroporto ( )
  - converti\_minuscolo ( )
- modifica\_gate ( ) : Dipendente dalle funzioni:
  - mergesort\_data\_per\_partenza ( )
  - riconoscimento\_aeroporto ( )
  - converti\_maiuscolo ( )
  - converti\_minuscolo ( )
- gestione\_emergenze ( ) : Dipendente dalle funzioni:
  - mergesort\_data\_per\_partenza ( )
  - cambio\_data\_avanti ( )
  - converti\_maiuscolo ( )
  - converti\_minuscolo ( )

- autorizzazione\_voli ( ) : Dipendente dalle funzioni:
  - cambio\_data\_avanti ( )
  - cambio\_data\_indietro ( )
  - converti\_maiuscolo ( )
  - converti\_minuscolo ( )
- caricamento\_info\_modelli\_aerei ( ) : Non dipendente da altre funzioni
- caricamento\_info\_aeromobili ( ) : Non dipendente da altre funzioni
- caricamento\_info\_voli ( ) : Non dipendente da altre funzioni
- caricamento\_info\_aeroporti ( ) : Non dipendente da altre funzioni
- acquisizione\_password ( ) : Non dipendente da altre funzioni
- creazione\_log ( ) : Non dipendente da altre funzioni
- settaggio\_schermo ( ) : Non dipendente da altre funzioni
- stampa\_menu\_iniziale ( ) : Non dipendente da altre funzioni
- converti\_minuscolo ( ) : Non dipendente da altre funzioni
- converti\_maiuscolo ( ) : Non dipendente da altre funzioni
- stampa\_menu\_tabella\_voli ( ) : Non dipendente da altre funzioni
- stampa\_menu\_area\_riservata ( ) : Non dipendente da altre funzioni
- cambio\_data\_avanti ( ) : Non dipendente da altre funzioni
- cambio\_data\_indietro ( ) : Non dipendente da altre funzioni
- init\_suite ( ) : Non dipendente da altre funzioni
- clean\_suite ( ) : Non dipendente da altre funzioni
- test\_cambio\_data\_avanti ( ) : Dipendente dalle funzioni:
  - cambio\_data\_avanti ( )
- test\_cambio\_data\_indietro ( ) : Dipendente dalle funzioni:
  - cambio\_data\_indietro ( )
- test\_caricamento\_info\_modelli\_aerei ( ) : Dipendente dalle funzioni:
  - caricamento\_info\_modelli\_aerei ( )
- test\_caricamento\_info\_aeromobili ( ) : Dipendente dalle funzioni:
  - caricamento\_info\_aeromobili ( )



- test\_caricamento\_info\_voli ( ) : Dipendente dalle funzioni:  
caricamento\_info\_voli ( )
- test\_caricamento\_info\_aeroporti ( ) : Dipendente dalle funzioni:  
caricamento\_info\_aeroporti ( )
- test\_acquisizione\_password ( ) : Dipendente dalle funzioni:  
acquisizione\_password ( )
- test\_mergesort\_data\_per\_arrivo ( ) : Dipendente dalle funzioni:  
mergesort\_data\_per\_arrivo ( )
- test\_mergesort\_data\_per\_partenza ( ) : Dipendente dalle funzioni:  
caricamento\_info\_aeromobili ( )
- test\_riconoscimento\_aeroporto ( ) : Dipendente dalle funzioni:  
riconoscimento\_aeroporto ( )
- main ( ) : Dipendente dalle funzioni:  
caricamento\_info\_modelli\_aerei ( )  
caricamento\_info\_aeromobili ( )  
caricamento\_info\_voli ( )  
caricamento\_info\_aeroporti ( )  
acquisizione\_password ( )  
converti\_minuscolo ( )  
stampa\_menu\_tabella\_voli ( )  
stampa\_lista\_ordinata\_voli\_in\_arrivo ( )  
stampa\_lista\_ordinata\_voli\_in\_partenza ( )  
stampa\_volo\_ricerca\_per\_codice ( )  
stampa\_aereo\_ricerca\_per\_codice ( )  
stampa\_menu\_area\_riservata ( )  
cancellazione\_volo ( )  
modifica\_gate ( )  
gestione\_emergenze ( )  
autorizzazione\_voli ( )  
creazione\_log ( )  
init\_suite ( )  
clean\_suite ( )  
test\_cambio\_data\_avanti ( )  
test\_cambio\_data\_indietro ( )  
test\_caricamento\_info\_modelli\_aerei ( )  
test\_caricamento\_info\_aeromobili ( )  
test\_caricamento\_info\_voli ( )  
test\_caricamento\_info\_aeroporti ( )  
test\_acquisizione\_password ( )  
test\_mergesort\_data\_per\_arrivo ( )  
test\_mergesort\_data\_per\_partenza ( )  
test\_riconoscimento\_aeroporto ( )

## - PSEUDOCODICE

Di seguito, sono elencate tutte le funzioni presenti all'interno del progetto.

Funzione converti\_minuscolo ( ):

```
BEGIN converti_maiuscolo
  FOR I = 0 WHILE stringa_da_convertire[i] != '\0' STEP 1 DO
    IF stringa_da_convertire[i] >= 'A' && stringa_da_convertire[i] <= 'Z' THEN
      stringa_da_convertire[i] = stringa_da_convertire[i] + 32
    END-IF
  END-FOR
  RETURN stringa_da_convertire
END converti_maiuscolo
```

Funzione converti\_maiuscolo ( )

```
BEGIN converti_maiuscolo
  FOR I = 0 WHILE stringa_da_convertire[i] != '\0' STEP 1 DO
    IF stringa_da_convertire[i] >= 'a' && stringa_da_convertire[i] <= 'z' THEN
      stringa_da_convertire[i] = stringa_da_convertire[i] - 32
    END-IF
  END-FOR
  RETURN stringa_da_convertire
END converti_maiuscolo
```

Funzione cambio\_data\_avanti ( )

```
BEGIN cambio_data_avanti
  SELECT mese
    CASE 1, 3, 5, 7, 8, 10 OR 12:
      IF giorno > 31 THEN
        IF mese == 12 THEN
          giorno = 1
          mese = 1
          anno = anno + 1
        ELSE
          giorno = 1
          mese = mese + 1
        END-IF
      END-IF
    BREAK
  END-CASE
  CASE 4, 6, 9 OR 11:
    IF giorno > 30 THEN
      mese = mese + 1
      giorno = 1
    END-IF
    BREAK
  END-CASE
  CASE 2:
    IF (anno - 2000) % 4 THEN
      IF giorno > 29 THEN
        mese = mese + 1
        giorno = 1
      END-IF
    ELSE
      IF giorno > 28 THEN
        mese = mese + 1
        giorno = 1
      END-IF
    END-IF
    BREAK
  END-CASE
  CASE default:
    BREAK
  END-CASE
END-SELECT
END cambio_data_avanti
```

Funzione cambio\_data\_indietro ( )

```
BEGIN cambio_data_indietro
  SELECT mese
    CASE 11, 9, 8, 6, 4, 2 OR 1:
      IF giorno < 1 THEN
        IF mese == 1 THEN
          giorno = 31
          mese = 12
          anno = anno - 1
        ELSE
          giorno = 31
          mese = mese - 1
        END-IF
      END-IF
      BREAK
    END-CASE
    CASE 12, 10, 7 OR 5:
      IF giorno < 1 THEN
        mese = mese - 1
        giorno = 30
      END-IF
      BREAK
    END-CASE
    CASE 3:
      IF giorno < 1 THEN
        IF (anno - 2000) % 4 THEN
          mese = mese - 1
          giorno = 29
        ELSE
          mese = mese - 1
          giorno = 28
        END-IF
      END-IF
      BREAK;
    END-CASE
    CASE default:
      BREAK;
    END-CASE
  END-SELECT
END cambio_data_indietro
```

Funzione stampa\_lista\_voli\_ordinati\_in\_partenza ( )

```
BEGIN stampa_lista_voli_ordinati_in_partenza
i = 0
mergesort_data_per_arrivo (volo, 0, numero_voli - 1)
WHILE scelta != "esci tabella" DO
    IF volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' THEN
        PRINT dati delle informazioni dell'aereo del volo
    IF volo[i].cancellato == 'n' || volo[i].cancellato == 'N' THEN
        PRINT dati delle informazioni sugli orari del volo
    IF volo[i].autorizzato == 's' || volo[i].autorizzato == 'S' THEN
        PRINT messaggio di volo autorizzato
    ELSE
        PRINT messaggio di volo non autorizzato
    END-IF
    ELSE
        PRINT messaggio di volo cancellato
    END-IF
    SCAN scelta
    scelta = converti_minuscolo (scelta)
    IF scelta == "prossimo" THEN
        i = i + 1
    END-IF
    ELSE
        i = i + 1
    END-IF
    IF i == numero_voli THEN
        i = 0
    END-IF
END-WHILE
END stampa_lista_voli_ordinati_in_partenza
```

Funzione stampa\_lista\_voli\_ordinati\_in\_arrivo ( )

```
BEGIN stampa_lista_voli_ordinati_in_arrivo
i = 0
mergesort_data_per_arrivo (volo, 0, numero_voli - 1)
WHILE scelta != "esci tabella" DO
    IF volo[i].partenza_arrivo == 'a' || volo[i].partenza_arrivo == 'A' THEN
        PRINT dati delle informazioni dell'aereo del volo
    IF volo[i].emergenza == 's' || volo[i].emergenza == 'S' THEN
        PRINT comunicazione di emergenza
    END-IF
    IF volo[i].autorizzato == 's' || volo[i].autorizzato == 'S' THEN
        PRINT comunicazione di autorizzazione
    END-IF
    IF volo[i].autorizzato == 'n' || volo[i].autorizzato == 'N' THEN
        PRINT comunicazione di non autorizzazione
    END-IF
    SCAN scelta
    scelta = converti_minuscolo (scelta)
    IF scelta == "prossimo" THEN
        i = i + 1
    END-IF
ELSE
    i = i + 1
END-IF
IF i == numero_voli THEN
    i = 0
END-IF
END-WHILE
END stampa_lista_voli_ordinati_in_arrivo
```

Funzione mergesort\_data\_per\_arrivo ( )

```
BEGIN mergesort_data_per_arrivo
IF sinistra < destra THEN
    centro = (sinistra + destra) / 2
    mergesort_data_per_arrivo (volo, sinistra, centro)
    mergesort_data_per_arrivo (volo, centro + 1, destra)
    merge_data_per_arrivo (volo, sinistra, centro, destra)
END_IF
END mergesort_data_per_arrivo
```

Funzione merge\_data\_per\_arrivo ( )

```
BEGIN merge_data_per_arrivo
i = sinistra
j = centro + 1
k = 0
WHILE i <= centro && j <= destra DO
    IF volo[i].arrivo.data_estesa > volo[j].arrivo.data_estesa THEN
        appoggio[k] = volo[i]
        i = i + 1
    ELSE
        appoggio[k] = volo[j]
        j = j + 1
    END-IF
    k = k + 1
END-WHILE
WHILE i <= centro DO
    appoggio[k] = volo[i]
    i = i + 1
    k = k + 1
END-WHILE
WHILE j <= destra DO
    appoggio[k] = volo[j]
    j = j + 1
    k = k + 1
END-WHILE
FOR k = sinistra TO destra STEP 1 DO
    volo[k] = appoggio [k - sinistra]
END-FOR
END merge_data_per_arrivo
```

Funzione mergesort\_data\_per\_partenza ( )

```
BEGIN mergesort_data_per_partenza
IF sinistra < destra THEN
    centro = (sinistra + destra) / 2
    mergesort_data_per_partenza (volo, sinistra, centro)
    mergesort_data_per_partenza (volo, centro + 1, destra)
    merge_data_per_partenza (volo, sinistra, centro, destra)
END_IF
END mergesort_data_per_partenza
```

Funzione merge\_data\_per\_arrivo ( )

```
BEGIN merge_data_per_arrivo
i = sinistra
j = centro + 1
k = 0
WHILE i <= centro && j <= destra DO
    IF volo[i].partenza.data_estesa > volo[j].partenza.data_estesa THEN
        appoggio[k] = volo[i]
        i = i + 1
    ELSE
        appoggio[k] = volo[j]
        j = j + 1
    END-IF
    k = k + 1
END-WHILE
WHILE i <= centro DO
    appoggio[k] = volo[i]
    i = i + 1
    k = k + 1
END-WHILE
WHILE j <= destra DO
    appoggio[k] = volo[j]
    j = j + 1
    k = k + 1
END-WHILE
FOR k = sinistra TO destra STEP 1 DO
    volo[k] = appoggio [k - sinistra]
END-FOR
END merge_data_per_arrivo
```

Funzione riconoscimento\_aeroporto ( )

```
BEGIN riconoscimento_aeroporto
FOR i = 0 TO numero_aeroporti STEP 1 DO
    WHILE codice_da_cercare == aeroporto[i].codice DO
        nome_aeroporto = aeroporto[i].nome
        i = i + 1
    END-WHILE
END-FOR
RETURN nome_aeroporto
END riconoscimento_aeroporto
```



Funzione stampa\_volo\_ricerca\_per\_codice ( )

```
BEGIN stampa_volo_ricerca_per_codice
WHILE scelta == "si" DO
    trovato = 0
    SCAN codice_da_cercare
    codice_da_cercare = converti_maiuscolo (codice_da_cercare)
    WHILE scelta != "si" || scelta != "no" DO
        FOR i = 0 TO numero_voli STEP 1 DO
            IF volo[i].codice_id == codice_da_cercare THEN
                PRINT informazioni sul volo
                trovato = 1
            END-IF
        END-FOR
        IF trovato == 0 THEN
            PRINT messaggio di non ritrovamento
        END-IF
        SCAN scelta
    END-WHILE
END-WHILE
END stampa_volo_ricerca_per_codice
```

Funzione stampa\_aereo\_ricerca\_per\_codice ( )

```
BEGIN stampa_aereo_ricerca_per_codice
WHILE scelta == "si" DO
    WHILE scelta != "si" || scelta != "no" DO
        WHILE i < numero_voli && trovato != 1 DO
            IF volo[i].codice_id == codice_da_cercare THEN
                targa = volo[i].targa
                WHILE j < numero_aerei && trovato != 1 DO
                    IF aereo[j].targa == targa THEN
                        model = aereo[j].targa
                        WHILE k < numero_modelli && trovato != 1 DO
                            IF model == modello[k].modello THEN
                                PRINT dati dell'aereo trovato
                                trovato = 1
                            END-IF
                            k = k + 1
                        END-WHILE
                    END-IF
                    j = j + 1
                END-WHILE
            END-IF
            i = i + 1
        END-WHILE
        IF trovato == 0 THEN
            PRINT messaggio di non ritrovamento
        END-IF
    END-WHILE
END stampa_aereo_ricerca_per_codice
```

```

SCAN scelta
scelta = converti_minuscolo (scelta)
END_WHILE
END-WHILE
END stampa_aereo_ricerca_per_codice

```

Funzione cancellazione\_voli ( )

```

BEGIN cancellazione_voli
trovato = 0
i = 0
scelta_corretta = 0
mergesort_data_per_partenza (volo, 0, numero_voli - 1)
WHILE scelta_menu != "avanti" && scelta_menu != "cancella" && scelta_menu != "esci cancella"
    WHILE scelta_corretta != 1 DO
        voli_stampati = 0
        scorrimento_indice = 0
        WHILE i < numero_voli && voli_stampati < MAX_VOLI_STAMPATI DO
            IF volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' THEN
                destinazione = riconoscimento_aeroporto (volo[i].codice_iata, aeroporto,
                    numero_aeroporti)
                IF volo[i].cancellato == 'n' && volo[i].cancellato == 'N' THEN
                    PRINT valori del volo
                ELSE
                    PRINT messaggio di volo cancellato
                END-IF
                voli_stampati = voli_stampati + 1
            END-IF
            i = i + 1
            scorrimento_indice = scorrimento_indice + 1
        END-WHILE
        SCAN scelta_menu
        scelta_menu = converti_minuscolo (scelta_menu)
        IF scelta_menu == "cancella" THEN
            SCAN volo_da_cancellare
            volo_da_cancellare = converti_maiuscolo (volo_da_cancellare)
            FOR i = 0 TO numero_voli STEP 1 DO
                IF volo[k].codice == volo_da_cancellare && volo[k].cancellato == 'n' THEN
                    volo[k].numero_passeggeri = -1
                    volo[k].partenza.giorno = -1
                    volo[k].partenza.mese = -1
                    volo[k].partenza.anno = -1
                    volo[k].partenza.ora = -1
                    volo[k].partenza.minuto = -1
                    volo[k].arrivo.giorno = -1
                    volo[k].arrivo.mese = -1
                    volo[k].arrivo.anno = -1
                    volo[k].arrivo.ora = -1
                    volo[k].arrivo.minuto = -1
                END-IF
            END-FOR
        END-IF
    END-WHILE
END

```

```

        volo[k].cancellato= -1
        volo[k].emergenza = -1
        volo[k].autorizzato = -1
        trovato = 1
    END-IF
END-FOR
IF trovato == 0 THEN
    PRINT messaggio di errore
END-IF
i = i - scorrimento_indice
ELSE IF scelta_menu != "avanti" || scelta_menu != "cancella" || scelta_menu != "esci
    cancella" THEN
        scelta_corretta = -1
    ELSE
        scelta_corretta = 1
    END-IF
    IF scelta_corretta == -1 THEN
        i = i - scorrimento_indice
    END-IF
END-WHILE
IF i == numero_voli THEN
    i = 0
END-IF
END-WHILE
END cancellazione_voli

```

Funzione modifica\_gate ( )

```
BEGIN modifica_gate
i = 0
trovato = 0
scelta_corretta = 0
mergesort_data_per_partenza (volo, 0, numero_voli - 1)
WHILE scelta_menu != "esci gate" DO
    WHILE scelta_corretta != 1 && inserimento_corretto != 1 && trovato == 0 DO
        voli_stampati = 0
        scorrimento_indice = 0
        inserimento_corretto = 0
        WHILE i < numero_voli && voli_stampati < MAX_VOLI_STAMPATI DO
            prov_dest = riconoscimento_aeroporto (volo[i].codice_iata, aeroporto,
            numero_aeroporti)
            IF volo_cancellato == 'n' THEN
                PRINT informazioni sul volo
            ELSE IF volo_cancellato == 's' THEN
                PRINT messaggio di errore
            END-IF
            IF volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' THEN
                PRINT informazioni su volo in partenza
            ELSE IF volo[i].partenza_arrivo == 'a' || volo[i].partenza_arrivo == 'A' THEN
                PRINT informazioni sul volo in arrivo
            END-IF
            voli_stampati = voli_stampati + 1
            i = i + 1
            scorrimento_indice = scorrimento_indice + 1
        END-WHILE
        SCAN scelta_menu
        scelta_menu = converti_minuscolo (scelta_menu)
        IF scelta_menu == "cambia" THEN
            SCAN volo_da_modificare
            volo_da_modificare = converti_maiuscolo (volo_da_modificare)
            FOR k = 0 TO numero_voli STEP 1 DO
                IF volo[k].codice_id == volo_da_modificare && volo[k].cancellato == 'n' THEN
                    trovato = 1
                    SCAN nuovo_gate
                    nuovo_gate = converti_maiuscolo (nuovo_gate)
                    IF nuovo_gate == "A1" || nuovo_gate == "A2" || nuovo_gate == "A3" ||
                    nuovo_gate == "A4" || nuovo_gate == "A5" THEN
                        IF nuovo_gate == volo[k].gate THEN
                            PRINT messaggio di errore
                        ELSE
                            PRINT messaggio di corretto cambiamento
                            volo[k].gate = nuovo_gate
                            inserimento_corretto = 1
                        END-IF
                    END-IF
                END-IF
            END-IF
        END-IF
    END-IF
END-WHILE
```

```

ELSE IF nuovo_gate != "A1" && nuovo_gate != "A2" &&
    nuovo_gate != "A3" && nuovo_gate == "A4" && nuovo_gate == "A5"
    PRINT messaggio di errore
END-IF
END-IF
END-FOR
IF trovato == 0 THEN
    PRINT messaggio di errore
END-IF
i = i - scorrimento_indice
ELSE IF scelta != "avanti" && scelta != "cambia" && scelta != "esci gate" THEN
    scelta_corretta = -1
ELSE
    scelta_corretta = 1
END_IF
IF scelta_corretta == -1 THEN
    i = i - scorrimento_indice
END-IF
END-WHILE
IF i == numero_voli THEN
    i = 0
END-IF
END-WHILE
END modifica_gate

```

Funzione gestione\_emergenze ( )

```
BEGIN gestione_emergenze ( )
tempo_emergenza = 0
WHILE scelta == "sì" DO
    SCAN volo_emergenza
    volo_emergenza = converti_maiuscolo (volo_emergenza)
    trovato = 0
    i = 0
    WHILE i < numero_voli && trovato != 1 DO
        IF volo[i].codice_id == volo_emergenza && volo[i].cancellato == 'n' THEN
            IF volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' THEN
                WHILE tempo_emergenza < 0 || tempo_emergenza > 60 DO
                    SCAN tempo_emergenza
                    IF tempo_emergenza < 1 THEN
                        PRINT messaggio di errore
                    END-IF
                    IF tempo_emergenza > 60 THEN
                        PRINT messaggio di errore
                    END-IF
                END-WHILE
                volo[i].arrivo.anno = volo[i].partenza.anno
                volo[i].arrivo.mese = volo[i].partenza.mese
                volo[i].arrivo.giorno = volo[i].partenza.giorno
                volo[i].arrivo.ora = volo[i].partenza.ora
                volo[i].arrivo.minuto = volo[i].partenza.minuto + tempo_emergenza
                volo[i].partenza_arrivo = 'a'
                volo[i].emergenza = 's'
                trovato = 1
                IF volo[i].arrivo.minuto > 60 THEN
                    volo[i].arrivo.minuto = volo[i].arrivo.minuto - 60
                    volo[i].arrivo.ora = volo[i].arrivo.ora + 1
                    IF volo[i].arrivo.ora > 24 THEN
                        volo[i].arrivo.ora = volo[i].arrivo.ora - 24
                        volo[i].arrivo.giorno = volo[i].arrivo.giorno + 1
                    END-IF
                    cambio_data_avanti (volo[i].arrivo.anno, volo[i].arrivo.mese,
                        volo[i].arrivo.giorno)
                END-IF
                PRINT informazioni sui voli
            ELSE IF volo[i].partenza_arrivo == 'a' || volo[i].partenza_arrivo == 'A' THEN
                trovato = -1
            END-IF
        END-IF
        IF volo[i].codice_id == volo_emergenza && volo[i].cancellato == 's' THEN
            PRINT messaggio di errore
        END-IF
        i = i + 1
    END-WHILE
```

```

WHILE scelta != "si" && scelta != "no" DO
  SELECT trovato
  CASE -1
    PRINT mostra messaggio di errore
    BREAK
  END-CASE
  CASE 0
    PRINT mostra messaggio di errore
    BREAK
  END-CASE
  CASE 1
    PRINT mostra messaggio di riuscita
    BREAK
  END-CASE
  DEFAULT
    BREAK
  END-CASE
END-SELECT
SCAN scelta
scelta = converti_minuscolo (scelta)
END-WHILE
END-WHILE
END gestione_emergenze

```

Funzione autorizzazione\_voli ( )

```
BEGIN autorizzazione_voli
k = 0
ritardo_casuale = 0
voli_stampati = 0
WHILE scelta == "si" DO
    i = 0
    trovato = 0
    SCAN volo_autorizzato
    volo_autorizzato = converti_maiuscolo (volo_autorizzato)
    WHILE i < numero_voli && trovato == 0 DO
        IF volo[i].codice_id == volo_autorizzato THEN
            trovato == 1
        ELSE
            i = i + 1
        END-IF
        IF volo[i].autorizzato == 's' || volo[i].autorizzato == 'S' THEN
            trovato = - 1
            IF volo[i].cancellato == 's' || volo[i].cancellato == 'S' THEN
                PRINT messaggio di errore
            ELSE
                PRINT messaggio di errore
            END-IF
        END-IF
        IF volo[i].cancellato == 's' || volo[i].cancellato == 'S' && trovato == 1 THEN
            trovato = -1
            PRINT messaggio di errore
        END-IF
    END-WHILE
    IF trovato == 1 THEN
        ritardo_casuale = -10 + rand () % 20
        volo[i].autorizzato = 's'
        IF volo[i].partenza_arrivo == 'a' || volo[i].partenza_arrivo == 'A' THEN
            volo[i].effettiva = volo[i].arrivo
            volo[i].effettiva.minuto = volo[i].effettiva.minuto + ritardo_casuale
            IF volo[i].effettiva.minuto <= 0 THEN
                volo[i].effettiva.ora = volo[i].effettiva.ora - 1
                volo[i].effettiva.minuto = volo[i].effettiva.minuto + 59
                IF volo[i].effettiva.ora <= 0 THEN
                    volo[i].effettiva.ora = 23
                    volo[i].effettiva.giorno = volo[i].effettiva.giorno + 1
                END-IF
            ELSE IF volo[i].effettiva.minuto >= 60 THEN
                volo[i].effettiva.minuto = volo[i].effettiva.minuto - 60
                volo[i].effettiva.ora = volo[i].effettiva.ora + 1
                IF volo[i].effettiva.ora >= 24 THEN
                    volo[i].effettiva.ora = 0
                    volo[i].effettiva.giorno = volo[i].effettiva.giorno + 1
                END-IF
            END-IF
        END-IF
    END-IF
END-IF
```



```

END-IF
IF ritardo_casuale > 0 THEN
    cambio_data_avanti (volo[i].effettiva.anno, volo[i].effettiva.mese,
        volo[i].effettiva.giorno)
ELSE IF ritardo_casuale < 0 THEN
    cambio_data_indietro (volo[i].effettiva.anno, volo[i].effettiva.mese,
        volo[i].effettiva.giorno)
END-IF
PRINT informazioni sul volo
ELSE IF volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' THEN
    volo[i].effettiva = volo[i].partenza
    volo[i].effettiva.minuto = volo[i].effettiva.minuto + ritardo_casuale
    IF volo[i].effettiva.minuto <= 0 THEN
        volo[i].effettiva.ora = volo[i].effettiva.ora - 1
        volo[i].effettiva.minuto = volo[i].effettiva.minuto + 59
        IF volo[i].effettiva.ora <= 0 THEN
            volo[i].effettiva.ora = 23
            volo[i].effettiva.giorno = volo[i].effettiva.giorno + 1
        END-IF
    ELSE IF volo[i].effettiva.minuto >= 60 THEN
        volo[i].effettiva.minuto = volo[i].effettiva.minuto - 60
        volo[i].effettiva.ora = volo[i].effettiva.ora + 1
        IF volo[i].effettiva.ora >= 24 THEN
            volo[i].effettiva.ora = 0
            volo[i].effettiva.giorno = volo[i].effettiva.giorno + 1
        END-IF
    END-IF
    PRINT informazioni sul volo
ELSE IF trovato == 0 THEN
    PRINT messaggio di errore
END-IF
WHILE scelta != "si" && scelta != "no" DO
    k = 0
    WHILE voli_stampati < MAX_VOLI_STAMPATI && k < numero_voli DO
        IF volo[i].autorizzato == 's' || volo[i].autorizzato == 'S' &&
            volo[i].autorizzato == 'n' || volo[i].autorizzato == 'N' THEN
            PRINT informazioni sul volo
            voli_stampati = voli_stampati + 1
        END-IF
    END-WHILE
    SCAN scelta
    scelta = converti_minuscolo (scelta)
END-WHILE

```

```
END-WHILE
END autorizzazione_volo
```

Funzione caricamento\_info\_modelli\_aerei ( )

```
BEGIN caricamento_info_modelli_aerei
IF fp = fopen("db/modelli.csv","r") != NULL THEN
    i = 0
    WHILE !feof DO
        FILE-SCAN appoggio
        modello[i].azienda = strtok (appoggio, ",")
        modello[i].modello = strtok (appoggio, ",")
        modello[i].propulsore = strtok (appoggio, ",")
        modello[i].numero_motori = strtok (appoggio, ",")
        modello[i].numero_massimo_passeggeri = strtok (appoggio, ",")
        i = i + 1
    END-WHILE
    numero_modelli = i
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
END caricamento_info_modelli_aerei
```

Funzione caricamento\_info\_aeromobili ( )

```
BEGIN caricamento_info_aeromobili
IF fp = fopen("db/aeromobili.csv","r") != NULL THEN
    i = 0
    WHILE !feof DO
        FILE-SCAN appoggio
        aereo[i].targa = strtok (appoggio, ",")
        aereo[i].modello = strtok (appoggio, ",")
        i = i + 1
    END-WHILE
    numero_aerei = i
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
END caricamento_info_aeromobili
```

Funzione caricamento\_info\_voli ( )

```
BEGIN caricamento_info_voli
IF fp = fopen("gest/voli.csv","r") != NULL THEN
    i = 0
    WHILE !feof DO
        FILE-SCAN appoggio
        volo[i].codice_id = strtok (appoggio, ",")
        volo[i].nome_compagnia = strtok (appoggio, ",")
        volo[i].targa = strtok (appoggio, ",")
        volo[i].partenza_arrivo = strtok (appoggio, ",")
        volo[i].codice_iata = strtok (appoggio, ",")
        volo[i].partenza.giorno = strtok (appoggio, "/")
        volo[i].partenza.mese = strtok (appoggio, "/")
        volo[i].partenza.anno = strtok (appoggio, ",")
        volo[i].partenza.ora = strtok (appoggio, ":")
        volo[i].partenza.minuto = strtok (appoggio, ",")
        volo[i].arrivo.giorno = strtok (appoggio, "/")
        volo[i].arrivo.mese = strtok (appoggio, "/")
        volo[i].arrivo.anno = strtok (appoggio, ",")
        volo[i].arrivo.ora = strtok (appoggio, ":")
        volo[i].arrivo.minuto = strtok (appoggio, ",")
        volo[i].numero_passeggeri = strtok (appoggio, ",")
        volo[i].gate = strtok (appoggio, ",")
        volo[i].cancellato = 'n'
        volo[i].autorizzato = 'n'
        volo[i].emergenza = 'n'
        volo[i].arrivo.data_estesa = (volo[i].arrivo.giorno * 0.0001) +
            (volo[i].arrivo.mese * 0.000001) + (volo[i].arrivo.anno * 0.000000001) +
            (volo[i].arrivo.ora * 0.00000000001) + (volo[i].arrivo.minuto * 0.0000000000001)
        volo[i].partenza.data_estesa = (volo[i].partenza.giorno * 0.0001) +
            (volo[i].partenza.mese * 0.000001) + (volo[i].partenza.anno * 0.000000001) +
            (volo[i].partenza.ora * 0.00000000001) + (volo[i].partenza.minuto * 0.0000000000001)
        i = i + 1
    END-WHILE
    numero_voli = i
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
END caricamento_info_voli
```

Funzione caricamento\_info\_aeroporti ( )

```
BEGIN caricamento_info_aeroporti
IF fp = fopen("db/aeroporti.csv","r") != NULL THEN
    i = 0
    WHILE !feof DO
        FILE-SCAN appoggio
        aereoporto[i].codice = strtok (appoggio, ",")
        aereoporto[i].nome = strtok (appoggio, ";")
        i = i + 1
    END-WHILE
    numero_aeroporti = i
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
END caricamento_info_aeroporti
```

Funzione acquisizione\_password ( )

```
BEGIN acquisizione_password
password = 0
IF fp = fopen("db/key.bin","r+b") != NULL THEN
    FILE-SCAN password
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
RETURN password
END acquisizione_password
```

Funzione creazione\_log ( )

```
BEGIN creazione_log
tempo_in_secondi = time(NULL)
tempo = localtime (tempo_in_secondi)
percorso_file = "gest/log/log_{data del sistema}.csv"
IF fp = fopen (percorso_file, "w") != NULL THEN
    WHILE i < numero_voli DO
        FILE-PRINT voli in base alla loro modifica
        i = i + 1
    END-WHILE
ELSE
    PRINT messaggio di errore
END-IF
fclose (fp)
END creazione_log
```

Funzione init\_suite ( )

```
BEGIN init_suite
RETURN 0
END init_suite
```

Funzione clean\_suite ( )

```
BEGIN clean_suite
RETURN 0
END clean_suite
```

Funzione test\_cambio\_data\_avanti ( )

```
BEGIN test_cambio_data_avanti
anno = 2017
mese = 2
giorno = 29
cambio_data_avanti (anno, mese, giorno)
CU_ASSERT_NOT_EQUAL (anno, 0)
CU_ASSERT_TRUE (anno >= 2017)
CU_ASSERT_NOT_EQUAL (mese, 0)
CU_ASSERT_TRUE (mese >= 1)
CU_ASSERT_TRUE (mese <= 12)
CU_ASSERT_NOT_EQUAL (giorno, 0)
CU_ASSERT_TRUE (giorno >= 1)
CU_ASSERT_TRUE (giorno <= 31)
IF mese == 2 THEN
    CU_ASSERT_TRUE (giorno <= 29)
END-IF
END test_cambio_data_avanti
```

Funzione test\_cambio\_data\_indietro ( )

```
BEGIN test_cambio_data_indietro
anno = 2017
mese = 2
giorno = 29
cambio_data_avanti (anno, mese, giorno)
CU_ASSERT_NOT_EQUAL (anno, 0)
CU_ASSERT_TRUE (anno >= 2017)
CU_ASSERT_NOT_EQUAL (mese, 0)
CU_ASSERT_TRUE (mese >= 1)
CU_ASSERT_TRUE (mese <= 12)
CU_ASSERT_NOT_EQUAL (giorno, 0)
CU_ASSERT_TRUE (giorno >= 1)
CU_ASSERT_TRUE (giorno <= 31)
IF mese == 2 THEN
    CU_ASSERT_TRUE (giorno <= 29)
END-IF
END test_cambio_data_indietro
```

Funzione test\_caricamento\_info\_modelli ( )

```
BEGIN test_caricamento_info_modelli
numero_modelli = 0
caricamento_info_modelli (modello, numero_modello)
IF CU_ASSERT_TRUE (numero_modelli > 0) THEN
    FOR i = 0 TO numero_modelli STEP 1 DO
        CU_ASSERT_NOT_EQUAL (modello[i].azienda, '\0')
        CU_ASSERT_NOT_EQUAL (modello[i].modello, '\0')
        CU_ASSERT_TRUE (modello[i].propulsore == 'g' || modello[i].propulsore == 'e')
        CU_ASSERT_TRUE (modello[i].numero_propulsori > 0)
        CU_ASSERT_TRUE (modello[i].numero_massimo_passeggeri > 0)
    END-FOR
END-IF
END test_caricamento_info_modelli
```

Funzione test\_caricamento\_info\_aeromobili ( )

```
BEGIN test_caricamento_info_aeromobili
numero_aerei = 0
caricamento_info_aeromobili (aereo, numero_aerei)
IF CU_ASSERT_TRUE (numero_aerei > 0) THEN
    FOR i = 0 TO numero_aerei STEP 1 DO
        CU_ASSERT_NOT_EQUAL (aereo[i].modello, '\0')
        CU_ASSERT_NOT_EQUAL (aereo[i].targa, '\0')
    END-FOR
END-IF
END test_caricamento_info_aeromobili
```

Funzione test\_caricamento\_info\_aeroporti ( )

```
BEGIN test_caricamento_info_aeroporti
numero_aeroporti = 0
caricamento_info_aeroporti (aeroporto, numero_aeroporti)
IF CU_ASSERT_TRUE (numero_aeroporti > 0) THEN
    FOR i = 0 TO numero_aeroporti STEP 1 DO
        CU_ASSERT_NOT_EQUAL (aeroporto[i].codice, '\0')
        CU_ASSERT_NOT_EQUAL (aeroporto[i].nome, '\0')
    END-FOR
END-IF
END test_caricamento_info_aeroporti
```

Funzione test\_caricamento\_info\_voli ( )

```
BEGIN test_caricamento_info_voli
numero_voli = 0
caricamento_info_voli (volo, numero_voli)
IF CU_ASSERT_TRUE (numero_voli > 0) THEN
    FOR i = 0 TO numero_voli STEP 1 DO
        CU_ASSERT_NOT_EQUAL (volo[i].codice_id, '\0')
        CU_ASSERT_NOT_EQUAL (volo[i].nome_compagnia, '\0')
        CU_ASSERT_NOT_EQUAL (volo[i].targa, '\0')
        CU_ASSERT_TRUE (volo[i].partenza_arrivo == 'p' || volo[i].partenza_arrivo == 'P' ||
            volo[i].partenza_arrivo == 'a' || volo[i].partenza_arrivo == 'A')
        CU_ASSERT_NOT_EQUAL (volo[i].codice_iata, '\0')
        CU_ASSERT_TRUE (volo[i].partenza.minuto >= 0 && volo[i].partenza.minuto < 60)
        CU_ASSERT_TRUE (volo[i].partenza.ora >= 0 && volo[i].partenza.ora < 24)
        CU_ASSERT_TRUE (volo[i].partenza.mese >= 1 && volo[i].partenza.minuto <= 12)
        CU_ASSERT_TRUE (volo[i].partenza.anno >= 2017)
        CU_ASSERT_NOT_EQUAL (volo[i].partenza.data_estesa, 0)
        CU_ASSERT_TRUE (volo[i].arrivo.minuto >= 0 && volo[i].arrivo.minuto < 60)
        CU_ASSERT_TRUE (volo[i].arrivo.ora >= 0 && volo[i].arrivo.ora < 24)
        CU_ASSERT_TRUE (volo[i].arrivo.mese >= 1 && volo[i].arrivo.minuto <= 12)
        CU_ASSERT_TRUE (volo[i].arrivo.anno >= 2017)
        CU_ASSERT_NOT_EQUAL (volo[i].arrivo.data_estesa, 0)
        CU_ASSERT_TRUE (volo[i].numero_passeggeri > 0)
        CU_ASSERT_NOT_EQUAL (volo[i].gate, '\0')
        CU_ASSERT_TRUE (volo[i].cancellato == 'n' || volo[i].cancellato == 'N')
        CU_ASSERT_TRUE (volo[i].emergenza == 'n' || volo[i].emergenza == 'N')
        CU_ASSERT_TRUE (volo[i].autorizzato == 'n' || volo[i].autorizzato == 'N')
    END-FOR
END-IF
END test_caricamento_info_voli
```

Funzione test\_acquisizione\_password ( )

```
BEGIN test_acquisizione_password
test_password = 0
test_password = acquisizione_password ( )
CU_ASSERT_TRUE (test_password > 0)
END test_acquisizione_password
```



Funzione test\_mergesort\_data\_per\_arrivo ( )

```
BEGIN test_mergesort_data_per_arrivo
NUMERO_VALORI_TEST = 5
test_volo[0].arrivo.data_estesa = 3
test_volo[1].arrivo.data_estesa = 1
test_volo[2].arrivo.data_estesa = 4
test_volo[3].arrivo.data_estesa = 5
test_volo[4].arrivo.data_estesa = 2
mergesort_data_per_arrivo (test_volo, 0, NUMERO_VALORI_TEST - 1)
CU_ASSERT_TRUE (test_volo[0].arrivo.data_estesa > test_volo[1].arrivo.data_estesa)
CU_ASSERT_TRUE (test_volo[1].arrivo.data_estesa > test_volo[2].arrivo.data_estesa)
CU_ASSERT_TRUE (test_volo[2].arrivo.data_estesa > test_volo[3].arrivo.data_estesa)
CU_ASSERT_TRUE (test_volo[3].arrivo.data_estesa > test_volo[4].arrivo.data_estesa)
END test_mergesort_data_per_arrivo
```

Funzione test\_mergesort\_data\_per\_partenza ( )

```
BEGIN test_mergesort_data_per_partenza
NUMERO_VALORI_TEST = 5
test_volo[0].partenza.data_estesa = 3
test_volo[1].partenza.data_estesa = 1
test_volo[2].partenza.data_estesa = 4
test_volo[3].partenza.data_estesa = 5
test_volo[4].partenza.data_estesa = 2
mergesort_data_per_partenza (test_volo, 0, NUMERO_VALORI_TEST - 1)
CU_ASSERT_TRUE (test_volo[0].partenza.data_estesa > test_volo[1].partenza.data_estesa)
CU_ASSERT_TRUE (test_volo[1].partenza.data_estesa > test_volo[2].partenza.data_estesa)
CU_ASSERT_TRUE (test_volo[2].partenza.data_estesa > test_volo[3].partenza.data_estesa)
CU_ASSERT_TRUE (test_volo[3].partenza.data_estesa > test_volo[4].partenza.data_estesa)
END test_mergesort_data_per_partenza
```

Funzione test\_riconoscimento\_aeroporto ( )

```
BEGIN test_riconoscimento_aeroporto
NUMERO_VALORI_TEST = 5
test_codice_da_cercare = "FCO"
aeroporto[0].nome = "Milano Linate"
aeroporto[1].nome = "Londra Gatwick"
aeroporto[2].nome = "Roma Fiumicino"
aeroporto[3].nome = "Riga"
aeroporto[4].nome = "Zurigo"
aeroporto[0].codice = "LIN"
aeroporto[1].codice = "LGK"
aeroporto[2].codice = "FCO"
aeroporto[3].codice = "RIX"
aeroporto[4].codice = "ZRH"
riconoscimento_aeroporto (test_codice_da_cercare, aeroporto, numero_valori_test)
CU_ASSERT_STRING_EQUAL (test_aeroporto, "Roma Fiumicino")
END test_mergesort_data_per_arrivo
```

Funzione main ( )

```
BEGIN main
IF CUE_SUCCESS != CU_initialize_registry ( ) THEN
    RETURN CU_get_error( )
END-IF
CU_pSuite Suite_gestione_dati = CU_add_suite ("Suite di gestione_dati.c", init_suite, clean_suite)
CU_pSuite Suite_funzioni_varie = CU_add_suite ("Suite di funzioni_varie.c", init_suite,
clean_suite)
CU_pSuite Suite_gestione_aeroporto = CU_add_suite ("Suite di gestione_aeroporto.c", init_suite,
clean_suite)
CU_add_test (Suite_gestione_dati, "Test di caricamento_info_modelli()",
test_caricamento_info_modelli_aerei)
CU_add_test (Suite_gestione_dati, "Test di caricamento_info_aeromobili()",
test_caricamento_info_aeromobili)
CU_add_test (Suite_gestione_dati, "Test di caricamento_info_voli()", test_caricamento_info_voli)
CU_add_test (Suite_gestione_dati, "Test di caricamento_info_aeroporti()",
test_caricamento_info_aeroporti)
CU_add_test (Suite_gestione_dati, "Test di acquisizione_password()", test_acquisizione_password)
CU_add_test (Suite_funzioni_varie, "Test di cambio_data_indietro()", test_cambio_data_indietro)
CU_add_test (Suite_funzioni_varie, "Test di cambio_data_avanti()", test_cambio_data_avanti)
CU_add_test (Suite_gestione_aeroporto, "Test di mergesort_data_per_arrivo()",
test_mergesort_data_per_arrivo)
CU_add_test (Suite_gestione_aeroporto, "Test di mergesort_data_per_partenza()",
test_mergesort_data_per_partenza)
CU_add_test (Suite_gestione_aeroporto, "Test di riconoscimento_aeroporto()",
test_riconoscimento_aeroporto)
CU_basic_set_mode(CU_BRM_VERBOSE)
CU_basic_run_tests()
numero_modelli = 0
```

```

numero_aerei = 0
numero_voli = 0
numero_aeroporti = 0
password = 0
password_inserita = 0
tentativi = 3
srand(time(NULL))
settaggio_schermo ( )
caricamento_info_modelli_aerei (modello, numero_modelli)
caricamento_info_aeromobili (aereo, numero_aerei)
caricamento_info_voli (volo, numero_voli)
caricamento_info_aeroporti (aeroporto, numero_aeroporto)
password = acquisizione_password ( )
WHILE scelta_menu != "esci" DO
    stampa_menu_iniziale ( )
    SCAN scelta_menu
    scelta_menu = converti_minuscolo (scelta_menu)
    IF scelta_menu == "tabella" THEN
        WHILE scelta_menu != "esci tabella" DO
            stampa_menu_tabella_voli ( )
            SCAN scelta_menu
            scelta_menu = converti_minuscolo (scelta_menu)
            IF scelta_menu == "arrivo" THEN
                stampa_lista_ordinata_voli_in_arrivo (volo, numero_voli, aeroporto,
                    numero_aeroporti)
            ELSE IF scelta_menu == "partenza" THEN
                stampa_lista_ordinata_voli_in_partenza (volo, numero_voli, aeroporto,
                    numero_aeroporti)
            ELSE IF scelta_menu == "codice" THEN
                stampa_volo_ricerca_per_codice (volo, numero_voli, aeroporto,
                    numero_aeroporti)
            ELSE IF scelta_menu == "aereo" THEN
                stampa_aereo_ricerca_per_codice (volo, numero_voli, aeroporto,
                    numero_aeroporti, aereo, numero_aerei, modello, numero_modelli)
            ELSE IF scelta_menu == "esci tabella" THEN
                BREAK;
            END-IF
        END-WHILE
    ELSE IF scelta_menu == "tabella" THEN
        riuscita = 0
        WHILE tentativi > 0 && riuscita == 0 DO
            SCAN password_inserita
            IF password_inserita == password THEN
                WHILE scelta_menu != "esci riservata" DO
                    riuscita = 1
                    stampa_menu_area_riservata ( )
                    SCAN scelta_menu
                    scelta_menu = converti_minuscolo (scelta_menu)

```

```

IF scelta_menu == "cancella" THEN
    cancellazione_volo (volo, numero_voli, aeroporto, numero_aeroporti)
ELSE IF scelta_menu == "gate" THEN
    modifica_gate (volo, numero_voli, aeroporto, numero_aeroporti)
ELSE IF scelta_menu == "emergenza" THEN
    gestione_emergenze (volo, numero_voli)
ELSE IF scelta_menu == "autorizza" THEN
    autorizzazione_voli (volo, numero_voli)
ELSE IF scelta_menu == "esci riservata" THEN
    BREAK
END-IF
END-WHILE
ELSE IF password_inserita != password THEN
    tentativi = tentativi - 1
    PRINT messaggio di errore
END-IF
END-WHILE
ELSE IF scelta_menu == "esci" THEN
    BREAK
END-IF
END-WHILE
creazione_log (volo, numero_voli)
remove ("gest/ voli.csv")
PRINT messaggio di chiusura
CU_cleanup_registry()
RETURN CU_get_error();
END main

```

# **CODIFICA**

## **- DOCUMENTAZIONE DOXYGEN**

Collegamento all'indice del file doxygen [QUI](#)

# TESTING

## - DEFINIZIONE DEL PIANO DI TEST

Di seguito, sono elencate le definizioni dei piani di test effettuati per il progetto:

<b>Codice della funzione</b>	<b>Nome</b>	<b>Codice del test</b>	<b>Descrizione (asserzione effettuata con CUnit)</b>	<b>Eventuale input</b>	<b>Risultato atteso</b>	<b>Risultato ottenuto</b>
<b>C1</b>	<b>Caricamento dei database (file modelli.csv)</b>	1.01	La stringa modello[i].azienda non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	Airbus, Boeing, ecc.
		1.02	La stringa modello[i].modello non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	A330-300, 747-100, ecc
		1.03	La variabile modello[i].propulsore deve essere 'g' o 'e' (Con asserzione CUnit)	Tutti i valori del campo, presi da file	'g'    'e'	'g'    'e'
		1.04	La variabile modello[i].numero_motori deve essere maggiore di 0 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	> 0	2, 3, 4
		1.05	La variabile modello[i].numero_massimo_passeggeri deve essere maggiore di 0 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	> 0	200, 220, 524, ecc.
	<b>Caricamento dei database (file aeromobili.csv)</b>	1.06	La stringa aereo[i].modello non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	737-800, 707-120B, ecc
		1.07	La stringa aereo[i].targa non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	I-ESHT, CSZ-EF, ecc
	<b>Caricamento dei database (file voli.csv)</b>	1.08	La stringa volo[i].codice_id non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	FR7070, BA4501, ecc
		1.09	La stringa volo[i].nome_compagnia non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	Ryanair, Meridiana, ecc.



<b>C1</b>	<b>Caricamento dei database (file voli.csv)</b>	1.10	La stringa volo[i].targa non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	I-ESHT, NO-WMA, ecc
		1.11	La variabile volo[i].partenza_arrivo deve essere o 'p' o 'a' (Con asserzione CUnit)	Tutti i valori del campo, presi da file	'p'    'a'	'p'    'a'
		1.12	La stringa volo[i].codice_iata non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	FCO, LGK, RIX, ecc
		1.13	La variabile volo[i].partenza.minuto deve essere compresa tra 0 e 60 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	0 <= var < 60	50, 30, 0, ecc
		1.14	La variabile volo[i].partenza.ora deve essere compresa tra 0 e 23 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	0 <= var <= 23	9, 20, 15, ecc
		1.15	La variabile volo[i].partenza.giorno deve essere compresa tra 1 e 31 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	1 <= var <= 31	1
		1.16	La variabile volo[i].partenza.mese deve essere compresa tra 1 e 12 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	1 <= var <= 12	6
		1.17	La variabile volo[i].partenza.anno deve essere maggiore di 2017 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	var >= 2017	2017
		1.18	La variabile volo[i].partenza.data_estesa deve essere maggiore di 0 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	var > 0	0.201706010950, 0.201706011530, ecc
		1.19	La variabile volo[i].partenza.minuto deve essere compresa tra 0 e 60 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	0 <= var < 60	45, 30, 15, ecc

<b>Caricamento dei database (file voli.csv)</b>	1.20	La variabile volo[i].arrivo.ora deve essere compresa tra 0 e 23 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	0 <= var <= 23	11, 14, 18, ecc
	1.21	La variabile volo[i].arrivo.giorno deve essere compresa tra 1 e 31 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	1 <= var <= 31	1, 2
	1.22	La variabile volo[i].arrivo.mese deve essere compresa tra 1 e 12 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	1 <= var <= 12	6
	1.23	La variabile volo[i].arrivo.anno deve essere maggiore di 2017 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	var >= 2017	2017
	1.24	La variabile volo[i].arrivo.data_estesa deve essere maggiore di 0 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	var > 0	0.201706011530, 0.201706010945, ecc
	1.25	La variabile volo[i].numero_passeggeri deve essere maggiore di 0 (Con asserzione CUnit)	Tutti i valori del campo, presi da file	var > 0	178, 120, ecc
	1.26	La stringa volo[i].gate non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	A1, A2 A3, A4 A5
	1.27	La variabile volo[i].cancellato deve essere uguale ad 'n' o 'N' (Con asserzione CUnit)	Tutti i valori del campo, presi da file	'n'    'N'	'n'
	1.28	La variabile volo[i].emergenza deve essere uguale ad 'n' o 'N' (Con asserzione CUnit)	Tutti i valori del campo, presi da file	'n'    'N'	'n'
	1.29	La variabile volo[i].autorizzato deve essere uguale ad 'n' o 'N' (Con asserzione CUnit)	Tutti i valori del campo, presi da file	'n'    'N'	'n'
<b>Caricamento dei database</b>	1.30	La stringa aeroporto[i].codice non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	FCO, LGK, ecc

	(file aeroporti.csv)	1.31	La stringa aeroporto[i].nome non deve essere vuota (Con asserzione CUnit)	Tutti i valori del campo, presi da file	Stringa con almeno un carattere	Roma Fiumicino, ecc
<b>C2</b>	<b>Caricamento della password</b>	2.1	La variabile password deve essere maggiore di 0 (Con asserzione Cunit)	Valore preso da file	var > 0	1*****8 (*) password coperta per motivi di protezione
<b>M1</b>	<b>Visualizzazione del menù principale</b>	3.1	Stampa a video il menù principale	---	Menu principale	Menu principale
<b>M2</b>	<b>Visualizzazione del menù <i>tabella</i></b>	3.2	Stampa a video il menu delle tabelle	---	Menu delle tabelle	Menu delle tabelle
<b>M3</b>	<b>Visualizzazione del menù <i>riservata</i></b>	3.3	Stampa a video il menu dell'area riservata	---	Menu dell'area riservata	Menu dell'area riservata
<b>T1</b>	<b>Rappresentazion e della tabella dei voli in partenza</b>	4.1	Le variabili della struttura volo devono essere ordinati per data di arrivo (Con asserzione Cunit)	Volo[i]. arrivo. data_estesa e Volo[i + 1]. arrivo. data_estesa	Volo[i]. arrivo. data_estesa < Volo[i + 1]. arrivo. data_estesa	1 < 2 2 < 3 3 < 4 4 < 5
		4.2	Stampa a video della lista dei voli in partenza	---	Info dei voli in lista, uno per volta	Info dei voli in lista, uno per volta
		4.3	Controllo della scelta inserita	“prossimo”	Info del volo successivo alla lista	Info del volo successivo alla lista
		4.4	Controllo della scelta inserita	“esci tabella”	Esce dalla modalità	Esce dalla modalità
<b>T2</b>	<b>Rappresentazion e della tabella dei voli in uscita</b>	5.1	Le variabili della struttura volo devono essere ordinati per data di arrivo (Con asserzione Cunit)	Volo[i]. arrivo. data_estesa e Volo[i + 1]. arrivo. data_estesa	Volo[i]. arrivo. data_estesa < Volo[i + 1]. arrivo. data_estesa	1 < 2 2 < 3 3 < 4 4 < 5
		5.2	Stampa a video della lista dei voli in uscita	---	Info dei voli in lista, uno per volta	Info dei voli in lista, uno per volta

		5.3	Controllo della scelta inserita	“prossimo”	Info del volo successivo alla lista	Info del volo successivo alla lista
		5.4	Controllo della scelta inserita	“esci tabella”	Esce dalla modalità	Esce dalla modalità
<b>T3</b>	<b>Rappresentazione della tabella del volo ricercato per codice</b>	6.1	Controllo del codice inserito	Codice esatto	Info del volo con codice ID uguale al codice	Info del volo con codice ID uguale al codice
		6.2	Controllo del codice inserito	Codice sbagliato	Messaggio di errore per codice non trovato	Messaggio di errore per codice non trovato
		6.3	Controllo del codice inserito	Codice sbagliato con più di 24 caratteri	Messaggio di errore per codice non trovato	Blocco anomalo del sistema
		6.4	Stampa a video delle informazioni del volo trovato	---	Info del volo trovato	Info del volo trovato
		6.5	Controllo della scelta inserita	“sì”	Continua a far inserire un nuovo codice da ricercare	Continua a far inserire un nuovo codice da ricercare
		6.6	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità
<b>T4</b>	<b>Rappresentazione della tabella dell'aereo ricercato per codice</b>	7.1	Controllo del codice inserito	Codice esatto	Info dell'aereo il cui volo ha codice ID uguale al codice	Info del volo con codice uguale al codice
		7.2	Controllo del codice inserito	Codice sbagliato	Messaggio di errore per codice non trovato	Messaggio di errore per codice non trovato
		7.3	Controllo del codice inserito	Codice sbagliato con più di 24 caratteri	Messaggio di errore per codice non trovato	Blocco anomalo del sistema
		7.4	Stampa a video delle informazioni dell'aereo trovato	---	Info dell'aereo trovato	Info dell'aereo trovato

		7.5	Controllo della scelta inserita	“si”	Continua a far inserire un nuovo codice da ricercare	Continua a far inserire un nuovo codice da ricercare
		7.6	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità
<b>P1</b>	<b>Acquisizione della password</b>	8.1	Controllo della password inserita	Password errata	Contatore dei tentativi decresce o, se è già ad 1, blocca l'area riservata	Contatore dei tentativi decresce o, se è già ad 1, blocca l'area riservata
		8.2	Controllo della password inserita	Password giusta	Area riservata viene sbloccata	Area riservata viene sbloccata
		8.3	Controllo della password inserita	Password errata con presenza di caratteri alfabetici	Contatore dei tentativi decresce o, se è già ad 1, blocca l'area riservata	Contatore dei tentativi azzerato e l'area si blocca, anche se rimangono più di un tentativo
<b>R1</b>	<b>Cancellazione del volo</b>	9.1	Stampa a video della lista dei voli in partenza ordinati per data di partenza	---	Lista degli aerei in partenza	Lista degli aerei in partenza
		9.2	Controllo del codice inserito	Codice esatto	Sistema cancella il volo settando a -1 i campi	Sistema cancella il volo settando a -1 i campi
		9.3	Controllo del codice inserito	Codice sbagliato	Messaggio di errore per codice non trovato	Messaggio di errore per codice non trovato
		9.4	Controllo della scelta inserita	“si”	Continua a far inserire un nuovo codice da ricercare	Continua a far inserire un nuovo codice da ricercare
		9.5	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità

<b>R2</b>	<b>Modifica del gate del volo</b>	10.1	Stampa a video della lista dei voli in partenza ed in arrivo ordinati per data di partenza	---	Lista degli aerei in partenza ed in arrivo	Lista degli aerei in partenza ed in arrivo
		10.2	Controllo del codice inserito	Codice esatto	Mostra la selezione del gate da cambiare	Mostra la selezione del gate da cambiare
		10.3	Controllo del codice inserito	Codice sbagliato	Messaggio di errore per codice non trovato	Messaggio di errore per codice non trovato
		10.4	Controllo del gate inserito	Gate esatto	Cambia il gate del volo cercato	Cambia il gate del volo cercato
		10.5	Controllo del gate inserito	Gate sbagliato	Mostra un messaggio di errore	Mostra un messaggio di errore
		10.6	Controllo della scelta inserita	“si”	Continua a inserire un codice da ricercare	Continua a inserire un codice da ricercare
		10.7	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità
<b>R3</b>	<b>Gestione dei voli in emergenza</b>	11.1	Controllo del codice inserito	Codice esatto	Sistema chiede di inserire dopo quanto è stata richiesta la emergenza	Sistema chiede di inserire dopo quanto è stata richiesta la emergenza
		11.2	Controllo del codice inserito	Codice sbagliato	Mostra un messaggio di errore	Mostra un messaggio di errore
		11.3	Inserimento del tempo tra la partenza e la chiamata dell'emergenza	Tempo incluso tra 1 e 60 minuti	Cambia la dicitura del volo su 'emergenza'	Cambia la dicitura del volo su 'emergenza'
		11.4	Inserimento del tempo tra la partenza e la chiamata dell'emergenza	Tempo non incluso tra 1 e 60 minuti	Mostra un messaggio di errore	Mostra un messaggio di errore
		11.5	Controllo della scelta inserita	“si”	Continua a far inserire un nuovo codice da ricercare	Continua a far inserire un nuovo codice da ricercare

		11.6	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità
<b>R4</b>	<b>Autorizzazione dei voli</b>	12.1	Controllo del codice inserito	Codice esatto	Autorizza il volo cambiando la dicitura su autorizzato	Autorizza il volo cambiando la dicitura su autorizzato
		12.2	Controllo del codice inserito	Codice sbagliato	Mostra un messaggio di errore	Mostra un messaggio di errore
		12.3	Stampa a video la lista dei voli già autorizzati	---	Lista dei voli autorizzati	Lista dei voli autorizzati
		12.4	Controllo della scelta inserita	“si”	Continua a far inserire un nuovo codice da ricercare	Continua a far inserire un nuovo codice da ricercare
		12.5	Controllo della scelta inserita	“no”	Esce dalla modalità	Esce dalla modalità
<b>L1</b>	<b>Creazione del file di log</b>	13.1	Cancellazione del file dei voli	Azione riuscita	Cancella il file voli.csv	Cancella il file voli.csv
		13.2	Cancellazione del file dei voli	Azione non riuscita	Mostra un messaggio di errore	Mostra un messaggio di errore
		13.3	Creazione del file di log	Azione riuscita	Crea il file di log	Crea il file di log
		13.4	Creazione del file di log	Azione non riuscita	Mostra un messaggio di errore	Mostra un messaggio di errore

## - ESITI DEL PIANO DI TEST


Di seguito viene mostrata la fase di testing avvenuta con l'aiuto del tool Cunit. Implementando nel test registry le varie suite (viste come l'astrazione del test dei vari file del progetto) e aggiungendo i diversi test method alla sua suite d'appartenenza, si avvia il test prima dell'esecuzione vera e propria del programma.

```
CUnit - A Unit testing framework for C - Version 2.1-0
http://cunit.sourceforge.net/

Suite: Suite di gestione_dati.c
Test: Test di caricamento_info_modelli() ... passed
Test: Test di caricamento_info_aeromobili() ... passed
Test: Test di caricamento_info_voli() ... passed
Test: Test di caricamento_info_aeroporti() ... passed
Test: Test di acquisizione_password() ... passed
Suite: Suite di funzioni_varie.c
Test: Test di cambio_data_indietro() ... passed
Test: Test di cambio_data_avanti() ... passed
Suite: Suite di gestione_aeroporto.c
Test: Test di mergesort_data_per_arrivo() ... passed
Test: Test di mergesort_data_per_partenza() ... passed
Test: Test di riconoscimento_aeroporto() ... passed

--Run Summary: Type      Total      Ran      Passed      Failed
                suites         3         3         n/a         0
                tests        10         10         10         0
                asserts    1375    1375    1375         0
Premere un tasto per continuare . . .
```

**N.B.** Tale funzionalità non viene implementata nella fase di release del software. Viene usata soltanto delle build di debug per il programmatore del progetto.


Come mostrato nella fase precedente, le procedure individuate con i codici 6.2, 6.3 e  potrebbero riscontrare problemi nel caso di utilizzo errato del sistema.



## - PIANIFICAZIONE DI MIGLIORAMENTI

Il bug scovato sulle procedure di codice 6.2 e 6.3 possono essere risolti ampliando la memoria allocata in sistema per la stringa di acquisizione del codice.

L'errore è dovuto all'allocazione della memoria dedicata all'inserimento del codice di volo, per un totale di 24 byte, non più sufficiente se l'utente immette più di 24 caratteri in input.

Il bug scovato sulla procedura  può essere risolta cambiando il tipo della variabile che acquisisce la password da tastiera. In questo caso, occorrerebbe effettuare poi un casting della variabile della password salvata da file per permettere a quest'ultima di essere intesa come un'altra stringa ed effettuare un semplice confronto.