

TIPI DI DATO

RECORD Brano

idBran	Identificativo univoco del brano	Intero
titolo	Titolo del brano	Stringa lun>1 e <30
durata	Durata del brano in secondi	Intero >0
ascolti	Numero di ascolti	Intero >=0
anno	Anno di uscita del brano	Intero >=ANNOMIN<=AnnoCorrente

DEFINIZIONE DatabaseBran = array di brani

RECORD Album

idAlbum	Identificativo dell'album	Intero
titolo	Titolo dell'album	Stringa
anno	Anno di uscita dell'album	Stringa

DEFINIZIONE DatabaseAlbum = array di Album

RECORD Artista

idArtista	identificativo dell'artista	Intero
nome	nome dell'artista	Stringa
cognome	Cognome dell'artista	Stringa
nomeArte	Nome d'arte dell'artista	Stringa
linkBio	Url contenente la bio dell'artista	Stringa

DEFINIZIONE DatabaseArtista = array di artisti

RECORD BranoAlbum

idBran	Identificativo del brano	Intero
idAlbum	Identificativo dell'album	Intero

DEFINIZIONE DatabaseBranAlbum = array di BranoAlbum

RECORD BranoGenere

idBran	Identificativo del brano	Intero
idGenere	Identificativo del genere	Intero

DEFINIZIONE DatabaseBranoGenere = array di BranoGenere

RECORD BranoArtista

idBrano	Identificativo del brano	Intero	idArtista	Identificativo dell'artista
	Intero			

DEFINIZIONE DatabaseBranoArtista = array di BranoArtista

RECORD Utente

idUtente	ID univoco che identifica un utente	Intero
NomeUtente	Nome scelto dall'utente.	Stringa
Password	Password scelta dall'utente	Stringa
Admin	Valore booleano che permette di capire se l'utente è admin o meno.	Intero

DEFINIZIONE DatabaseUtenti = array di utenti

RECORD Genere

idGenere	ID univoco che identifica un genere.	Intero
nome	Nome del genere scelto dall'utente.	Stringa

DEFINIZIONE DatabaseGeneri = array di Genere

RECORD PLAYLIST

idPlaylist	Identificativo della playlist.	Intero
idUtente	Identificativo dell'utente.	Intero
nome	Nome della playlist.	Stringa lun>1 e <30
descrizione	Descrizione della playlist.	Stringa lun>1 e<250
pubblica	Booleano che ci dice se la playlist è pubblica o meno.	Intero

DEFINIZIONE DatabasePlaylist = array di Playlist

RECORD PLAYLISTBRANO

idPlaylist	Identificativo della playlist.	Intero
idBrani	Array contenente gli id della playlist.	Array di interi

FUNZIONI

InserireBrano()

INPUT

DBBrano	Database dei vari brani in cui inseriremo un nuovo brano	DatabaseBrano
NuovoBrano	Brano da inserire nella base dati letto dall'utente	Brano
DBBranoAlbum	Database delle relazioni tra album e brani.	DatabaseBranoAlbum
DBAlbum	Database che contiene le informazioni sugli album per creare le relazioni tra questi e il nuovo brano	DatabaseAlbum
DBBranoArtista	Database delle relazioni tra artisti e brani.	DatabaseBranoArtista
DBArtista	Database che contiene le informazioni sugli artisti per creare le relazioni tra questi e il nuovo brano inserito.	DatabaseArtista
DBBranoGenere	Database delle relazioni tra album e generi.	DatabaseBranoGenere
DBGenere	Database che contiene le informazioni sui generi per creare le relazioni tra questi e il nuovo brano inserito.	DatabaseGenere

OUTPUT

DBBrano	Database con all'interno il nuovo brano.	DatabaseBrano
---------	--	---------------

DATI LAVORO

Risposta	Input dell'utente che decide cosa fare.	Carattere
----------	---	-----------

ALGORITMO

ESEGUI

```
NuovoBranò=CreaBranò(DBBranò)
StampaBranò(NuovoBranò)
STAMPA("Confermi la tua scelta?(Y/N)")
Risposta=LeggiCarattere()
MENTRE(Risposta != 'Y')
DBBranò in posizione idBranò di nuovoBranò=nuovoBranò
DBBranò in posizione idBranò di nuovoBranò +1=IDSENTINELLA
CreaRelazioneBranòAlbum(DBBranòAlbum,nuovoBranò,DBAlbum)
CreaRelazioneBranòArtista(DBBranòArtista,nuovoBranò,DBArtista)
CreaRelazioneBranòGenere(DBBranòGenere,nuovoBranò,DBGenere)
```

CreaBranò()

INPUT

DBBranò Base dati che contiene da cui prenderemo la prima posizione disponibile DatabaseBranò
che daremo come id al nuovo brano.

Vari input da tastiera

OUTPUT

NuovoBranò Brano creato e dato in output. Brano

DATI LAVORO

AnnoSbagliato Variabile che tiene conto se l'anno inserito è sbagliato o è corretto Intero

ALGORITMO

```
Campo idBranò di NuovoBranò=CercaUltimoBranò(DBBranò)
STAMPA("Inserisci il nome del brano")
Campo titolo di NuovoBranò = LeggiStringa()
STAMPA("Inserisci la durata del brano, in secondi") Campo
Durata di NuovoBranò = LeggiIntero() Campo Ascolti di
NuovoBranò = 0
STAMPA("Inserisci l'anno di uscita del brano")
Campo anno di NuovoBranò=LeggiIntero()
AnnoSbagliato=1;
ESEGUI
    Campo anno di NuovoBranò = LeggiIntero()
    SE(Campo anno di NuovoBranò <ANNOMIN)OR(Campo anno di NuovoBranò>ANNOATTUALE)
    ALLORA
        STAMPA("Anno sbagliato, inseriscilo di nuovo")
    ALTRIMENTI
        AnnoSbagliato=0
    FINE
MENTRE(AnnoSbagliato= 1)
FINE
```

CercaUltimoBranò()

INPUT

DBBranò Database contenente tutti i brani da cui andremo a prendere la posizione DatabaseBranò
del brano sentinella nel database.

OUTPUT

UltimoID Ultimo id in cui andare a salvare il nuovo brano Intero

```

UltimoID=0
MENTRE(DBBrano in posizione UltimoID <> IDSENTINELLA)
    UltimoID++
FINE

```

BranoStampa	Funzione che stampa tutti i campi di un brano	Brano
-------------	---	-------

Stampe a schermo

```
STAMPA("Titolo: "Campo nome di BranoStampa)
STAMPA("Titolo: "Campo titolo di BranoStampa)
STAMPA("Durata: "Campo durata di BranoStampa)
STAMPA("Anno: "Campo anno di BranoStampa)
```

DBBranoAlbum	Database che contiene le relazione tra Brani ed Album, in cui inseriremo una nuova relazione.	DatabaseBranoAlbum
nuovoBrano	Brano a cui dovremo associare un album.	Brano
DBAlbum	Database che contiene gli album da cui prenderemo quello che ci interessa.	DatabaseAlbum

DBBranoAlbum Database con all'interno la nuova relazione tra brani ed Album DatabaseBranoAlbum

AlbumDaCercare	Input dell'utente che ci indica quale è l'album a cui è associato il brano.	Stringa
----------------	---	---------

Risposta	Input dell'utente che decide cosa fare.	Intero Carattere
RisultatoRicercaAlbum	ID dell'album ricercato.	

```
STAMPA("Questo brano all'interno di quale album si trova?")
AlbumDaCercare=LeggiStringa()
```

```
RisultatoRicercaAlbum=CercaAlbum(DBAlbum,AlbumDaCercare)
```

SE(RisultatoRicercaAlbum>=0)

ALLORA DBBranoAlbum=AggiungiRelazioneBranoAlbum(campo idBrano di NuovoBrano, RisultatoRicercaAlbum, DBBranoAlbum)

ALTRIMENTI STAMPA("L'album scelto non è nel catalogo, il brano verrà cancellato, confermi?(Y\N)")

```
Risposta=LeggiCarattere()
```

SE(Risposta='N') ALLORA

DBAlbum = InserireAlbum(DBAlbum)

```
DBBranoAlbum = AggiungiRelazioneBranoAlbum(campo idBrano di NuovoBrano, CercaUltimoAlbum
```

(DBAlbum), DBBranoAlbum)

ALTRIMENTI

```
//Qui va la funzione di cancellazione
```

FINE

FINE

DBAlbum	Database contenente tutti gli album su cui effettueremo la ricerca.	DatabaseAlbum
---------	---	---------------

NomeAlbum Nome da cercare nel database degli album. Stringa

OUTPUT

IDTrovato ID corrispondente all'album scelto. Intero

DATI LAVORO

Trovato Variabile che ci indica se l'id è stato trovato o meno. Intero i
 Indice usato per scandire il database. Intero

ALGORITMO

```
IDTrovato=-1
Trovato=0
MENTRE(DBAlbum in posizione i <> IDSENTINELLA AND Trovato=0)
    SE(campo nome di DBAlbum in posizione i=NomeAlbum)
        ALLORA
            Trovato=1
            IDTrovato=i
        FINE    i++
    FINE
```

AggiungiRelazioneBranAlbum()

INPUT

idBran Identificativo del brano da inserire nel database. Intero idAlbum
 Identificativo dell'album da inserire nel database. Intero DBBranAlbum
Database contenente le relazioni tra brani e album in cui aggiungeremo DatabaseBranAlbum una nuova
relazione.

OUTPUT

DBBranAlbum Database con all'interno la nuova relazione. DatabaseBranAlbum

DATI DI LAVORO

primaPosizioneLibera Primo id libero nel database per inserire la relazione. Intero

ALGORITMO

```
primaPosizioneLibera=cercaPrimaRelazioneBranAlbumLibera(DBBranAlbum)    campo
idAlbum di DBBranAlbum in posizione primaPosizioneLibera = idAlbum    campo idBran
di DBBranAlbum in posizione primaPosizioneLibera = idBran
DBBranAlbum in posizione primaPosizioneLibera +1= SENTINELLA
```

cercaPrimaRelazioneBranAlbumLibera()

INPUT

DBBranAlbum Base dati che contiene le relazioni tra brani e album di cui andremo a DatabaseBranAlbum
 trovare il primo posto libero. **OUTPUT**
i Intero che rappresenta il primo id disponibile. Intero

ALGORITMO

```
i=0
MENTRE(DBBranAlbumInPosizioneI<>IDSENTINELLA)
    i=i+1
FINE
```

CreaRelazioneBranArtista()

INPUT

DBBranArtista Database che contiene le relazione tra Brani ed Artisti, in cui inseriremo DatabaseBranArtista
 una nuova relazione.
nuovoBran Brano a cui dovremo associare un artista. Brano

DBArtista Database che contiene gli album da cui prenderemo quello che ci interessa. DatabaseArtista

OUTPUT

DBBranArtista Database con all'interno la nuova relazione tra brani ed artista.

DatabaseBranArtista

ALGORITMO

STAMPA("Chi è l'artista del brano che vuoi inserire?")

ArtistaDaCercare=LeggiStringa()

RisultatoRicercaArtista=CercaArtista(ArtistaDaCercare)

SE(RisultatoRicercaArtista>=0)

ALLORA DBBranArtista= AggiungiRelazioneBranArtista(campo idBran di NuovoBran,
RisultatoRicercaArtista, DBBranArtista)

ALTRIMENTI STAMPA("L'artista scelto non è nel catalogo, il brano verrà cancellato, confermi?(Y\N)")

Risposta=LeggiCarattere()

SE(Risposta='Y')

ALLORA

ALTRIMENTI

DBArtista=InserireArtista(DBArtista)

DBBranArtista=AggiungiRelazioneBranArtista(campo idBran di NuovoBran,
CercaUltimoArtista(DBArtista), DBBranArtista)

FINE

FINE

CercaArtista()

INPUT

DBArtista Database in cui cercheremo un artista dato il nome.

DatabaseArtista

NomeArtista Nome da cercare nel database degli artisti.

Stringa

OUTPUT

IDTrovato ID corrispondente all'artista scelto

Intero

DATI LAVORO

Trovato Variabile che ci indica se l'id è stato trovato o meno.

Intero i

Indice usato per scandire il database.

Intero

ALGORITMO

IDTrovato=-1

Trovato=0

MENTRE(DBArtista in posizione i <> IDSENTINELLA AND Trovato=0)

SE(campo nome di DBArtista in posizione i=NomeArtista)

ALLORA

Trovato=1

IDTrovato=i FINE i++

FINE

AggiungiRelazioneBranArtista()

INPUT

idBran Identificativo del brano da inserire nel database.

Intero idArtista

Identificativo dell'artista da inserire nel database.

Intero

DBBranArtista Database contenente le relazioni tra brani e artisti in cui aggiungeremo
una nuova relazione.

DatabaseBranArtista

OUTPUT

DBBranArtista Database con all'interno la nuova relazione.

DatabaseBranArtista

DATI DI LAVORO

primaPosizioneLibera Primo id libero nel database per inserire la relazione. Intero

ALGORITMO

primaPosizioneLibera=cercaPrimaRelazioneBranArtistaLibera(DBBranArtista) campo
idArtista di DBBranArtista in posizione primaPosizioneLibera = idArtista campo
idBran di DBBranArtista in posizione primaPosizioneLibera = idBran
DBBranArtista in posizione primaPosizioneLibera +1= IDSENTINELLA

cercaPrimaRelazioneBranArtistaLibera()

INPUT

DBBranArtista Base dati che contiene le relazioni tra brani e artisti di cui andremo a
DatabaseBranArtista trovare il primo posto libero.

OUTPUT

i Intero che rappresenta il primo id disponibile. Intero

ALGORITMO

i=0
MENTRE(DBBranArtistaInPosizioneI<>IDSENTINELLA) i=i+1
FINE

CreaRelazioneBranGenere()

INPUT

DBBranGenere Database che contiene le relazione tra Brani e Generi, in cui inseriremo
DatabaseBranGenere una nuova relazione.
nuovoBran Brano a cui dovremo associare un genere. Brano
DBGenere Database che contiene i generi da cui prenderemo quello che ci interessa. DatabaseGenere

OUTPUT

DBBranGenere Database con all'interno la nuova relazione tra brani e genere. DatabaseBranGenere

ALGORITMO

STAMPA("Il brano in questione di che genere è?")
GenereDaCercare=LeggiStringa()
RisultatoRicercaGenere=CercaGenere(GenereDaCercare)
SE(RisultatoRicercaGenere>=0)
ALLORA DBBranGenere=AggiungiRelazioneBranGenere(campo idBran di NuovoBran,
RisultatoRicercaGenere, DBBranGenere)

ALTRIMENTI STAMPA("Il genere scelto non è nel catalogo, il brano verrà cancellato, confermi?(Y\N)")
Risposta=LeggiCarattere()
SE(Risposta='Y')

ALLORA //Da Definire, cancella l'ultimo brano inserito e le relazioni a cui è connesso
ALTRIMENTI DBGenere=InserireGenere (DBGenere)
DBBranGenere = AggiungiRelazioneBranGenere (campo idBran di NuovoBran,
CercaUltimoGenere(DBGenere), DBBranGenere)

FINE
FINE

CercaGenere()

INPUT

DBGenere Database contenente tutti i generi. DatabaseGenere
NomeGenere Nome da cercare nel database dei generi. Stringa

OUTPUT

IDTrovato ID corrispondente al genere scelto. Intero

DATI LAVORO

Trovato Variabile che ci indica se l'id è stato trovato o meno. Intero i
Indice usato per scandire il database. Intero

ALGORITMO

```
IDTrovato=-1
Trovato=0
MENTRE(DBGenere in posizione i <> IDSENTINELLA AND Trovato=0)
    SE(campo nome di DBGenere in posizione i=NomeGenere)
    ALLORA
        Trovato=1
        IDTrovato=i
    FINE i++
FINE
```

AggiungiRelazioneBranogenere()

INPUT idBran Identificativo del brano da inserire nel database.
Intero idGenere Identificativo del genere da inserire nel database. Intero
DBBranogenere Database contenente le relazioni tra brani e generi in cui aggiungeremo una nuova relazione. DatabaseBranogenere

OUTPUT

DBBranogenere Database con all'interno la nuova relazione.
DatabaseBranogenere **DATI DI LAVORO**
primaPosizioneLibera Primo id libero nel database per inserire la relazione. Intero

ALGORITMO

```
primaPosizioneLibera=cercaPrimaRelazioneBranogenereLibera(DBBranogenere) campo
idGenere di DBBranogenere in posizione primaPosizioneLibera = idGenere campo
idBran di DBBranogenere in posizione primaPosizioneLibera = idBran DBBranogenere
in posizione primaPosizioneLibera +1= IDSENTINELLA
```

cercaPrimaRelazioneBranogenereLibera()

INPUT

DBBranogenere Base dati che contiene le relazioni tra brani e generi di cui andremo a DatabaseBranogenere
trovare il primo posto libero.

OUTPUT

i Intero che rappresenta il primo id disponibile. Intero

ALGORITMO

```
i=0
MENTRE(DBBranogenereInPosizioneI<> IDSENTINELLA) i=i+1
FINE
```

InserireArtista()

INPUT

DBArtista Database contenente tutti gli artisti in cui inseriremo un nuovo artista. DatabaseArtista
NuovoArtista Artista da inserire nel database. Artista

OUTPUT

DBArtista Database aggiornato con il nuovo artista. DatabaseArtista

ALGORITMO

ESEGUI

```
NuovoArtista=CreaArtista(DBArtista)
StampaArtista(NuovoArtista)
STAMPA("Confermi la tua scelta?(Y/N) Risposta=LeggiCarattere()
MENTRE(Risposta != 'Y')
```

FINE

DBArtista In Posizione campo IdArtista Di NuovoArtista)=NuovoArtista
DBArtista in Posizione (campo idArtista di NuovoArtista)+1 = IDSENTINELLA

CreaArtista()

INPUT

DBArtista Database utile per assegnare un id all'artista da inserire. DatabaseArtista

OUTPUT

NuovoArtista Artista inserito dall'utente tramite tastiera Artista

ALGORITMO campo

```
idArtista=cercaUltimoArtista(DBArtista)
STAMPA("Qual è il nome dell'artista da inserire?") Campo
nome di NuovoArtista = LeggiStringa()
STAMPA("Qual è il cognome dell'artista da inserire?")
Campo cognome di NuovoArtista = LeggiStringa()
STAMPA("Qual è il nome d'arte dell'artista da inserire?")
Campo nomeArte di NuovoArtista = LeggiStringa()
STAMPA("Inserire il link della biografia dell'artista")
Campo linkBio di NuovoArtista = LeggiStringa()
```

CercaUltimoArtista()

INPUT

DBArtista Database contente tutti gli artisti da cui andremo a prendere l'id in cui DatabaseArtista
è posizionato IDSENTINELLA.

OUTPUT

UltimoID Posizione di IDSENTINELLA Intero

ALGORITMO

```
UltimoID=0
MENTRE(DBArtista in posizione UltimoID <> IDSENTINELLA)
```

```
UltimoID++
```

FINE

StampaArtista()

INPUT

ArtistaStampa Artista di cui stamperemo a schermo tutti i campi.

Artista

OUTPUT

Vari output a schermo

ALGORITMO

```
STAMPA("Nome dell'artista: "Campo nome di ArtistaStampa)
STAMPA("Cognome dell'artista: "Campo cognome di ArtistaStampa)
STAMPA("Nome d'arte dell'artista: "Campo nomeArte di ArtistaStampa)
STAMPA("Link della bio dell'artista : "Campo linkBio di ArtistaStampa)
```

InserireAlbum()

INPUT

DBAlbum Database contenente tutti gli album in cui inseriremo un nuovo album.
NuovoAlbum Album da inserire nel database.

DatabaseAlbum
Album

OUTPUT

DBAlbum Database aggiornato con il nuovo album.

DatabaseAlbum

ALGORITMO

```
ESEGUI
    NuovoAlbum=CreaAlbum(DBAlbum) StampaArtista(NuovoAlbum)
    STAMPA("Confermi la tua scelta?(Y/N)")
    Risposta=LeggiCarattere()
MENTRE(Risposta <> 'Y')
FINE
DBAlbum In Posizione campo IdAlbum Di NuovoAlbum)=NuovoAlbum
DBAlbum in Posizione (campo idAlbum di NuovoAlbum)+1 = IDSENTINELLA
```

CreaAlbum()

INPUT

DBAlbum Database utile per assegnare un id all'album da inserire.

DatabaseAlbum

OUTPUT

NuovoAlbum Album inserito dall'utente tramite tastiera

Album

```
ALGORITMO campo idAlbum=cercaUltimoAlbum(DBAlbum)
STAMPA("Qual è il titolo dell'album da inserire?")
Campo titolo di NuovoAlbum = LeggiStringa()
STAMPA("Qual è l'anno di uscita dell'album da inserire?")
Campo anno di NuovoAlbum = LeggiStringa()
```

CercaUltimoAlbum()

INPUT

DBAlbum Database contenente tutti gli artisti da cui andremo a prendere l'id in cui
è posizionato IDSENTINELLA.

DatabaseAlbum

OUTPUT

UltimoID Posizione di IDSENTINELLA

Intero

ALGORITMO

```
UltimoID=0
MENTRE(DBAlbum in posizione UltimoID <> IDSENTINELLA)

    UltimoID++
FINE
```

INPUT

StampaAlbum()

AlbumStampa Album di cui stamperemo a schermo tutti i campi.

Album

OUTPUT

Vari output a schermo

ALGORITMO

STAMPA("Titolo dell'album: "Campo titolo di AlbumStampa)
STAMPA("Anno di uscita dell'album: "Campo anno di AlbumStampa)

InserireGenere()

INPUT

DBGenere Database contenente tutti i generi in cui inseriremo un nuovo Genere.
NuovoGenere Genere da inserire nel database.

DatabaseGenere
Genere

OUTPUT

DBGenere Database aggiornato con il nuovo genere.

DatabaseGenere

ALGORITMO

ESEGUI
 NuovoAlbum=CreaGenere(DBGenere)
 StampaGenere(NuovoGenere)
 STAMPA("Confermi la tua scelta?(Y/N)")
 Risposta=LeggiCarattere()
MENTRE(Risposta <> 'Y')
FINE
DBGenere In Posizione campo IdGenere Di NuovoGenere)=NuovoGenere
DBGenere in Posizione (campo idGenere di NuovoGenere)+1 = IDSENTINELLA

CreaGenere()

INPUT

DBGenere Database utile per assegnare un id al genere da inserire.

DatabaseGenere

OUTPUT

NuovoGenere Genere inserito dall'utente tramite tastiera

Genere

ALGORITMO campo idGenere=cercaUltimoGenere(DBGenere)
STAMPA("Qual è il nome del genere da inserire?")
Campo nome di NuovoGenere = LeggiStringa()

CercaUltimoGenere()

INPUT

DBGenere Database contenente tutti i generi da cui andremo a prendere l'id in cui
è posizionato IDSENTINELLA.

DatabaseGenere

OUTPUT

UltimoID Posizione di IDSENTINELLA

Intero

ALGORITMO

UltimoID=0
MENTRE(DBGenere in posizione UltimoID <> IDSENTINELLA)
 UltimoID++
FINE

INPUT

StampaGenere()

GenereStampa Genere di cui stamperemo a schermo tutti i campi.

Genere

OUTPUT

Vari output a schermo

ALGORITMO

STAMPA("Nome del genere: "Campo nome di GenereStampa)

InserireUtente()

INPUT

DBUtente Database contenente tutti gli utenti in cui inseriremo un nuovo Utente.

DatabaseUtente

NuovoUtente Utente da inserire nel database.

Utente

OUTPUT

DBUtente Database aggiornato con il nuovo utente.

DatabaseUtente

ALGORITMO

ESEGUI

NuovoUtente=CreaUtente(DBUtente) StampaUtente(NuovoUtente)

STAMPA("Confermi la tua scelta?(Y/N)")

Risposta=LeggiCarattere()

MENTRE(Risposta <> 'Y')

FINE

DBUtente In Posizione campo IdUtente Di NuovoUtente)=NuovoUtente

DBUtente in Posizione (campo idUtente di NuovoUtente)+1 = IDSENTINELLA

CreaUtente()

INPUT

DBUtente Database utile per assegnare un id all'utente da inserire.

DatabaseUtente

OUTPUT

NuovoUtente Utente inserito dall'utente tramite tastiera

Utente

ALGORITMO campo

idUtente=cercaUltimoUtente(DBUtente)

STAMPA("Qual è il nome utente dell'utente da inserire?") Campo

nome di NuovoUtente = LeggiStringa() STAMPA("Inserisci la password")

Campo password di NuovoUtente = LeggiStringa()

CercaUltimoUtente()

INPUT

DBUtente Database contenente tutti gli utenti da cui andremo a prendere l'id in cui posizionato IDSENTINELLA.

è DatabaseUtente

OUTPUT

UltimoID Posizione di IDSENTINELLA

Intero

ALGORITMO

UltimoID=0

MENTRE(DBUtente in posizione UltimoID <> IDSENTINELLA)

UltimoID++

FINE

StampaUtente()

INPUT

INPUT

UtenteStampa Utente di cui stamperemo a schermo tutti i campi. Utente

OUTPUT

Vari output a schermo

ALGORITMO

STAMPA("Nome utente: "Campo nome utente di UtenteStampa)

STAMPA("Password: "Campo pass di UtenteStampa)

InserisciPlaylist()

INPUT

IdUtente	Id dell'utente che sta utilizzando il sistema.	Intero
DBPlaylist	Database contenente le playlist in cui andremo ad inserire una nuova playlist.	DatabasePlaylist
DBBrano	Database contenente i brani da inserire nella playlist.	DatabaseBrano
NuovaPlaylist	Nuova playlist da inserire nel database.	Playlist

OUTPUT

DBPlaylist	Database con all'interno la nuova playlist.	DatabasePlaylist
------------	---	------------------

DATI DI LAVORO

Risposta	Carattere inserito dall'utente che ci dice se vuole creare la playlist o meno, poi se vuole aggiungere un brano o meno.	Carattere
----------	---	-----------

TitoloBrano	Titolo del brano da aggiungere alla playlist.
-------------	---

ALGORITMO

ESEGUI

NuovaPlaylist=CreaPlaylist(DBPlaylist,IdUtente)

StampaPlaylist(NuovaPlaylist)

STAMPA("Confermi la tua scelta?(Y/N)")

Risposta=LeggiCarattere()

MENTRE(Risposta<>"Y")

FINE

ESEGUI

STAMPA("Vuoi aggiungere un brano alla playlist?")

Risposta=LeggiCarattere()

SE(Risposta<>"Y")

ALLORA STAMPA("Vuoi aggiungere un brano alla playlist?")

titoloBrano=LeggiStringa()

br=RicercaBranoTramiteTitoloBrano(Risposta)

aggiungereBranoAPlaylist(nuovaPlaylist,br)

FINE

MENTRE(Risposta="Y")

FINE

DBPlaylist in posizione idPlaylist di nuovaPlaylist = nuovaPlaylist

CreaPlaylist()

INPUT

DBPlaylist	Database da cui prenderemo l'id.	DatabasePlaylist
IdUtente	Id dell'utente che vuole creare la playlist.	Intero

OUTPUT

NuovaPlaylist Playlist creata e data in output.

Playlist

ALGORITMO

```
campo idPlaylist di nuovaPlaylist=cercaUltimaPlaylist(DBPlaylist)
campoIdUtente di nuovaPlaylist= idUtente
STAMPA("Qual è il nome della playlist da inserire?")
Campo nome di NuovaPlaylist = LeggiStringa()
STAMPA("Inserisci la descrizione")
Campo descrizione di NuovoUtente = LeggiStringa() ESEGUI
    STAMPA("La Playlist è pubblica? (Y\N)")
    Risposta=LeggiCarattere()
MENTRE(Risposta <>"Y" AND Risposta <>"N")
FINE
SE(Risposta="Y")
    ALLORA campo pubblica di Nuova Playlist = 1
    ALTRIMENTI campo pubblica di NuovaPlaylist =0
FINE
```

cercaUltimaPlaylist()

INPUT

DBPlaylist Database contenente tutti le playlist da cui andremo a prendere l'id in cui
è posizionato IDSENTINELLA.

DatabasePlaylist

OUTPUT

UltimoID Posizione di IDSENTINELLA

Intero

ALGORITMO

```
UltimoID=0
MENTRE(DBUtente in posizione UltimoID <> IDSENTINELLA)
    UltimoID++
FINE
```

aggiungereBranoAPlaylist()

INPUT

branoPlaylist Brano da inserire nella playlist.
nuovaPlaylist Playlist in cui andrà il branoPlaylist.
DBPlaylistBrano Database in cui inseriremo la relazione tra la playlist e il brano.

Brano
Playlist
DatabasePlaylistBrano

OUTPUT

DBPlaylistBrano Database con all'interno la relazione tra la playlist e il brano.

DatabasePlaylistBrano

DATI DI LAVORO

primaPosizioneLibera Primo id libero nel database per inserire la relazione.

Intero

ALGORITMO

```
primaPosizioneLibera=cercaPrimaRelazionePlaylistBranoLibera(DBPlaylistBrano)
campo idPlaylist di DBPlaylistBrano in posizione primaPosizioneLibera = idPlaylist
campo idBrano di DBPlaylistBrano in posizione primaPosizioneLibera = idBrano
DBPlaylistBrano in posizione primaPosizioneLibera +1= IDSENTINELLA
```

cercaPrimaRelazionePlaylistBranoLibera()

INPUT

DBPlaylistBrano Base dati che contiene le relazioni tra brani e playlist di cui andremo a trovare il primo posto libero.

DatabasePlaylistBrano

OUTPUT

i Intero che rappresenta il primo id disponibile.

Intero

ALGORITMO

```
i=0
MENTRE(DBPlaylistBranoInPosizioneI<> IDSENTINELLA)
```

i=i+1
FINE

INPUT

idBrano	Identificatore del brano da modificare.	Intero
Campo	Campo della struttura dati da modificare.	Intero
DBBrani	DataBase contenente il brano da modificare.	DatabaseBrani

OUTPUT

DBBrani	DataBase contenente i Brani modificato.	DatabaseBrani
---------	---	---------------

DATI LAVORO

titolo	Titolo del brano.	Stringa
durata	Nuova durata del brano.	Intero durata>0
anno	anno del brano.	Intero

ALGORITMO

```
SE(Campo=1)
  ALLORA
    STAMPA("Inserisci nuovo nome")
    titolo=LeggiStringa()

    campo titolo di DBBrano in posizione idBrano=titolo
  ALTRIMENTI
    SE(Campo=2)
      ALLORA
        STAMPA("Qual è la nuova durata?")
        durata=LeggiIntero()

        campo durata di DBBrano in posizione idBrano=durata
      ALTRIMENTI
        SE(Campo=3)
          ALLORA
            STAMPA("In che anno è uscito il brano?")
            ESEGUI
            anno=LeggiIntero()
            MENTRE(anno<1000 OR anno >ANNOATTUALE)
            campo anno di DBBrano in posizione idBrano=anno FINE
```

ModificaAlbum()

INPUT

DBAlbum	DataBase contenente gli Album.	DataBaseAlbum
idAlbum	Identificatore dell'album da modificare.	Intero
Campo	Campo della struttura dati da modificare.	Intero

OUTPUT

DBAlbum	Album a cui sono state applicate le modifiche	DatabaseAlbum
---------	---	---------------

DATI LAVORO

titolo	titolo dell'album	Stringa
anno	anno del brano	Intero >=ANNOMIN<=ANNOATTUALE

ALGORITMO

```
SE(Campo=1)
  ALLORA
```



```

        STAMPA("nuovo nome")  titolo=LeggiString()
        campo titolo di DBAlbum in posizione idAlbum=titolo
    ALTRIMENTI
SE(Campo=2)  ALLORA
    STAMPA("quando è
uscito l'album?")

        ESEGUI
        anno=LeggiIntero()
        MENTRE(anno<1000 OR anno>ANNOATTUALE)
        FINE  campo anno di DBAlbum in posizione
idAlbum=anno
    FINE

```

ModificaArtista()

INPUT

DBArtista	DataBase contenente gli artisti.	DatabaseArtista
idArtista	Identificatore dell'artista da modificare.	Intero
Campo	Campo della struttura dati da modificare.	Intero

OUTPUT

DBArtista	DataBase contenente gli artisti modificato.	DatabaseArtista
-----------	---	-----------------

DATI LAVORO

nuovoNome	Nuovo nome attribuito all'artista.	Stringa
nuovoCognome	Nuovo cognome attribuito all'artista.	Stringa
nomeArte	Nuovo nome d'arte attribuito all'artista.	Stringa

ALGORITMO

```

SE(Campo=1)
    ALLORA
        STAMPA("Inserisci nuovo nome")
        nuovoNome=LeggiStringa()
        campo nome di DBArtista in posizione idartista=nuovoNome
    ALTRIMENTI
SE(Campo=2)
    ALLORA
        STAMPA("Inserisci nuovo cognome")
        nuovoCognome=LeggiStringa()
        campo cognome di
DBArtista in posizione idartista=nuovoCognome
        ALTRIMENTI
SE(Campo=3)
    ALLORA
        STAMPA("Qual è il suo nome d'arte")
        nomeArte=LeggiStringa()
        campo nomeArte di
DBArtista in
posizione idartista=nomeArte

    ALTRIMENTI
SE(Campo=4)
    ALLORA
        STAMPA("Inserisci nuovo linkbio")
        nuovoLinkBio=LeggiStringaLunga()
        Campo linkbio di DBArtista in posizione idArtista=nuovoLinkBio
    FINE
FINE
FINE
FINE

```

ModificaGenere()

INPUT

DBGenere	DataBase contenente i generi	DatabaseGenere
Idgenere	Identificatore dell'artista da modificare	Intero

OUTPUT

DBGenere	DataBase contenente i generi	DatabaseGenere
----------	------------------------------	----------------

DATI LAVORO nuovoNome Nuovo nome attribuito all'artista Stringa

ALGORITMO

```
STAMPA("nuovo nome")    nuovoNome=LeggiStringa()  
campo nome di DBGenere in  
posizione idgenere=nuovoNome
```

ModificaPlaylist()**INPUT**

DBPlaylist	DataBase contenente le playlist	DataBasePlaylist
Idplaylist	Identificatore della playlist da modificare	Intero
Campo	Campo della struttura dati da modificare	Intero

OUTPUT

DBPlaylist	DataBase contenente le playlist	DataBasePlaylist
------------	---------------------------------	------------------

DATI LAVORO

nuovoNome	Nuovo nome attribuito alla Playlist	Stringa
nuovadescrizione	Nuova descrizione della playlist	Stringa

ALGORITMO

```
SE(Campo=1)  
ALLORA  
    STAMPA("InserisciNuovo nome")  
    nuovonome=LeggiStringa()  
    campo nome di DBPlaylist in posizione idplaylist=nuovonome  
    ALTRIMENTI  
SE(Campo=2)  
ALLORA  
    STAMPA("InserisciNuova descrizione")  
    nuovadescrizione=LeggiStringaLunga()  
    campo descrizione di DBPlaylist in posizione idplaylist=nuovadescrizione  
FINE  
FINE
```

ModificaUtente()**INPUT**

DBUtente	DataBase contenente dell'Utente	DataBaseUtenti
idUtente	Identificatore dell'utente da	Intero modificare
Campo	Campo della struttura dati da	Intero
	modificare	

OUTPUT

DBUtente	DataBase contenente dell'Utente	DataBaseUtenti
----------	---------------------------------	----------------

DATI LAVORO

nuovoNomeUtente	Nuovo nome dell'utente	Stringa
nuovaPassword	Nuova Password dell'utente	Stringa

ALGORITMO

```
SE(Campo=1)  
ALLORA  
    STAMPA("Inserisci nuovo nome utente")  
    nuovoNomeUtente=LeggiStringa()
```

```

        campo nomeUtente di DBUtente in posizione idUtente=nuovoNomeUtente
ALTRIMENTI
SE(Campo=2)
ALLORA
    STAMPA("Inserisci nuova password")
    nuovaPasswor
d=LeggiStringa()
    campo password di DBUtente in posizione idUtente=nuovaPassword
FINE
FINE

```

CancellaBran()

INPUT

idBran	ID del brano da cancellare.	Intero
DBBran	Database di brani da cui cancellare un elemento	DatabaseBran
DBBranArtista	Database di BranoArtista da cui cancellare un elemento	DatabaseBranArtista
DBBranAlbum	Database di BranoAlbum da cui cancellare un elemento	DatabaseBranAlbum
DBBranGenere	Database di BranoGenere da cui cancellare un elemento	DatabaseBranGenere
DBBranPlaylist	Database di BranoPlaylist da cui cancellare un elemento	DatabaseBranPlaylist

OUTPUT

DBBran	Database di brani da cui è stato cancellato un elemento	DatabaseBran
--------	---	--------------

ALGORITMO

```

DBBran=campo idBran di (DBBran in posizione idBran)=-2
DBBranArtista=CancellaRelBranArtista(idBran, DBBranArtista)
DBBranAlbum=CancellaRelBranAlbum(idBran, DBBranAlbum)
DBBranGenere=CancellaRelBranGenere(idBran, DBBranGenere)
DBBranPlaylist=CancellaRelBranPlaylist(idBran, DBBranPlaylist)

```

CancellaArtista()

INPUT

idArtista	ID dell'artista da cancellare	Intero
DBArtisti	Database di artisti da cui cancellare un elemento	DatabaseArtisti
DBBran	Database di brani da cui cancellare un elemento	DatabaseBran
DBBranArtista	Database di BranoArtista da cui cancellare un elemento	DatabaseBranArtista
DBBranAlbum	Database di BranoAlbum da cui cancellare un elemento	DatabaseBranAlbum
DBBranGenere	Database di BranoGenere da cui cancellare un elemento	DatabaseBranGenere
DBBranPlaylist	Database di BranoPlaylist da cui cancellare un elemento	DatabaseBranPlaylist

OUTPUT

DBArtisti	Database di artisti da cui è stato cancellato un elemento	DatabaseArtisti
-----------	---	-----------------

ALGORITMO

```

Campo idArtista di DBArtisti in posizione idArtista =-2 i=0
MENTRE(DBBranArtista in posizione i!= IDCANCELLAZIONE)
    SE(campo idArtista di DBBranArtista in posizione i= idArtista)
        ALLORA
            CancellaBran(campo idBran di DBBranArtista in posizione i, DBBran, DBBranArtista, DBBranAlbum,
            DBBranGenere,
            DBBranPlaylist) FINE i++
FINE

```

CancellaGenere()

INPUT

idGenere	ID del genere da cancellare	Intero
DBGeneri	Database di brani da cui cancellare un elemento	DatabaseGeneri
DBBran	Database di brani da cui cancellare un elemento	DatabaseBran

DBBranoArtista	Database di BranoArtista da cui cancellare un elemento	DatabaseBranoArtista
DBBranoAlbum	Database di BranoAlbum da cui cancellare un elemento	DatabaseBranoAlbum
DBBranoGenere	Database di BranoGenere da cui cancellare un elemento	DatabaseBranoGenere
DBBranoPlaylist	Database di BranoPlaylist da cui cancellare un elemento	DatabaseBranoPlaylist

OUTPUT

DBGenere	Database di generi da cui è stato cancellato un elemento	DatabaseGeneri
----------	--	----------------

ALGORITMO

Campo idGenere di DBGenere in posizione idGenere = -2 i=0

MENTRE(DBBranoGenere in posizione i!= IDSENTINELLA)

SE(campo idGenere di DBBranoArtista in posizione i= idGenere)

ALLORA

CancellaBrano(campo idBrano di DBBranoArtista in posizione i, DBBrani, DBBranoArtista, DBBranoAlbum, DBBranoGenere,

DBBranoPlaylist) FINE i++

FINE

CancellaPlaylist()

INPUT

idPlaylist	ID del genere da cancellare	Intero
DBPlaylist	Database di playlist da cui cancellare un elemento	DatabasePlaylist
DBBrani	Database di brani da cui cancellare un elemento	DatabaseBrani
DBBranoArtista	Database di BranoArtista da cui cancellare un elemento	DatabaseBranoArtista
DBBranoAlbum	Database di BranoAlbum da cui cancellare un elemento	DatabaseBranoAlbum
DBBranoGenere	Database di BranoGenere da cui cancellare un elemento	DatabaseBranoGenere
DBPlaylist	Database di brani da cui è stato cancellato un elemento	DatabasePlaylist
DBBranoPlaylist	Database di BranoGenere da cui è stato cancellato un elemento	DatabaseBranoPlaylist

OUTPUT

ALGORITMO

Campo idPlaylist di DBPlaylist in posizione idPlaylist=-2

i=0

MENTRE(DBBranoPlaylist in posizione i <> IDCANCELLAZIONE AND campo idPlaylist di DBBranoPlaylist in posizione i <> idPlaylist)

i++

FINE

SE(DBBranoPlaylist in posizione i <> IDCANCELLAZIONE)

ALLORA campo idPlaylist di DBBranoPlaylist in posizione i =-2

FINE

CancellaRelBranoArtista()

INPUT

idBrano	ID del brano di cui cancellare la relazione	Intero
DBBranoArtista	Database di Brano-Artista da cui cancellare un elemento	DatabaseBranoArtista

OUTPUT

DBBranoArtista	Database di Brano-Artista da cui è stato cancellato un elemento	DatabaseBranoArtista
----------------	---	----------------------

ALGORITMO

i=0

MENTRE(campo idBrano di DBBranoArtista in posizione i <> idBrano AND DBBranoArtista <> IDSENTINELLA) i=i+1

FINE

SE(DBBranoArtista in posizione i != IDCANCELLAZIONE)

ALLORA campo idBrano di DBBranoArtista in posizione i=-2 campo

idArtista di DBBranoArtista in posizione i=-2

FINE

CancellaRelBranoAlbum()

INPUT

idBrano ID del brano di cui cancellare la relazione Intero
DBBranoAlbum Database di Brano-Album da cui cancellare un elemento DatabaseBranoAlbum

OUTPUT

DBBranoAlbum Database di Brano-Album da cui è stato cancellato un elemento DatabaseBranoAlbum

DATI LAVORO

i indice utilizzato per effettuare la ricerca(per poi cancellare le intero eventualirelazioni)

ALGORITMO

```
i=0  MENTRE(campo idBrano di DBBranoAlbum in posizione i <> idBrano AND DBBranoAlbum  
<> IDCANCELLAZIONE)  i=i+1  FINE  
SE(DBBranoAlbum in posizione i <> IDCANCELLAZIONE)  
ALLORA campo idBrano di DBBranoAlbum in poszione i=-2  
campo idAlbum di DBBranoAlbum in poszione i=-2  
FINE
```

CancellaRelBranoGenere()

INPUT

idBrano ID del brano di cui cancellare la relazione Intero
DBBranoGenere Database di Brano-Genere da cui cancellare un elemento DatabaseBranoGenere

OUTPUT

DBBranoGenere Database di Brano-Genere da cui è stato cancellato un elemento DatabaseBranoGenere

ALGORITMO

```
i=0  
MENTRE(campo idBrano di DBBranoGenere in posizione i <> idBrano AND DBBranoGenere <>  IDSENTINELLA)  
i=i+1  
FINE  
SE(DBBranoGenere in posizione i <> IDCANCELLAZIONE)  
ALLORA campo idBrano di DBBranoGenere in poszione i=-2  
campo idGenere di DBBranoGenere in poszione i=-2  
FINE
```

CancellaRelBranoPlaylist()

INPUT

idBrano ID del brano di cui cancellare la relazione intero
DBBranoPlaylist Database di Brano-Playlist da cui cancellare un elemento DatabaseBranoPlaylist

OUTPUT

DBBranoPlaylist Database di Brano-Playlist da cui è stato cancellato un elemento DatabaseBranoPlaylist

ALGORITMO i=0

```
MENTRE(campo idBrano di DBBranoPlaylist in posizione i <> idBrano AND DBBranoPlaylist <>  
IDCANCELLAZIONE)  
i=i+1  
FINE  
SE(DBBranoPlaylist in posizione i <> IDCANCELLAZIONE)  
ALLORA campo idBrano di DBBranoPlaylist in poszione i=-2  
FINE
```