

## 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 DatabaseBranGenere** = array di BranoGenere

### RECORD BranoArtista

idBran	Identificativo del brano	Intero
idArtista	Identificativo dell'artista	Intero

**DEFINIZIONE DatabaseBranArtista** = 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	Intero

o meno.

**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 inserito.	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

```
NuovoBrano=CreaBrano(DBBrano)
StampaBrano(NuovoBrano)
STAMPA("Confermi la tua scelta?(Y/N)")
Risposta=LeggiCarattere()
```

MENTRE(Risposta != 'Y')

```
DBBrano in posizione idBrano di nuovoBrano=nuovoBrano
DBBrano in posizione idBrano di nuovoBrano +1=BRANOSENTINELLA
CreaRelazioneBranoAlbum(DBBranoAlbum,nuovoBrano,DBAlbum)
CreaRelazioneBranoArtista(DBBranoArtista,nuovoBrano,DBArtista)
CreaRelazioneBranoGenere(DBBranoGenere,nuovoBrano,DBGenere)
```

##### CreaBrano()

###### INPUT

DBBrano	Base dati che contiene da cui prenderemo la prima posizione disponibile che daremo come id al nuovo brano.	DatabaseBrano
---------	--	---------------

Vari input da tastiera

###### OUTPUT

NuovoBrano	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 idBrano di NuovoBrano=CercaUltimoBrano(DBBrano)
STAMPA("Inserisci il nome del brano")
Campo titolo di NuovoBrano = LeggiStringa()
STAMPA("Inserisci la durata del brano, in secondi")
```

```

Campo Durata di NuovoBranco = LeggiIntero()
Campo Ascolti di NuovoBranco = 0
STAMPA("Inserisci l'anno di uscita del brano")
Campo anno di NuovoBranco=LeggiIntero()
AnnoSbagliato=1;
ESEGUI
    Campo anno di NuovoBranco = LeggiIntero()
    SE(Campo anno di NuovoBranco <ANNOMIN)OR(Campo anno di NuovoBranco>ANNOATTUALE)
    ALLORA
        STAMPA("Anno sbagliato, inseriscilo di nuovo")
    ALTRIMENTI
        AnnoSbagliato=0
MENTRE(AnnoSbagliato= 1)

```

### CercaUltimoBranco()

#### INPUT

DBBranco	Database contenente tutti i brani da cui andremo a prendere la posizione del brano sentinella nel database.	DatabaseBranco
----------	---	----------------

#### OUTPUT

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

#### ALGORITMO

```

UltimoID=0
MENTRE(DBBranco in posizione UltimoID <> BRANOSENTINELLA)
    UltimoID++
FINE

```

### StampaBranco()

#### INPUT

BrancoStampa	Funzione che stampa tutti i campi di un brano	Branco
--------------	---	--------

#### OUTPUT

Stampe a schermo

#### ALGORITMO

```

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

```

### CreaRelazioneBrancoAlbum()

#### INPUT

DBBrancoAlbum	Database che contiene le relazione tra Brani ed Album, in cui inseriremo una nuova relazione.	DatabaseBrancoAlbum
nuovoBranco	Branco a cui dovremo associare un album.	Branco
DBAlbum	Database che contiene gli album da cui prenderemo quello che ci interessa.	DatabaseAlbum

#### OUTPUT

DBBrancoAlbum	Database con all'interno la nuova relazione tra brani ed Album	DatabaseBrancoAlbum
---------------	--	---------------------

#### DATI DI LAVORO

AlbumDaCercare	Input dell'utente che ci indica quale è l'album a cui è associato il brano.	Stringa
RisultatoRicercaAlbum	ID dell'album ricercato.	Intero
Risposta	Input dell'utente che decide cosa fare.	Carattere

#### ALGORITMO

```

STAMPA("Questo brano all'interno di quale album si trova?")
AlbumDaCercare=LeggiStringa()
RisultatoRicercaAlbum=CercaAlbum(DBAlbum,AlbumDaCercare)

```

```

SE(RisultatoRicercaAlbum>=0)
    ALLORA DBBranAlbum=AggiungiRelazioneBranAlbum(campo idBran di NuovoBran,
        RisultatoRicercaAlbum, DBBranAlbum)
    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)
        DBBranAlbum = AggiungiRelazioneBranAlbum(campo idBran di NuovoBran, CercaUltimoAlbum
            (DBAlbum), DBBranAlbum)
    ALTRIMENTI
        //Qui va la funzione di cancellazione
    FINE
FINE

```

### CercaAlbum()

#### INPUT

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 <> ALBUMSENTINELLA 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 una nuova relazione.	DatabaseBranAlbum

#### 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= BRANOALBUMSENTINELLA

```

### cercaPrimaRelazioneBranAlbumLibera()

#### INPUT

DBBranAlbum	Base dati che contiene le relazioni tra brani e album di cui andremo a trovare il primo posto libero.	DatabaseBranAlbum
-------------	---	-------------------

#### OUTPUT

i                      Intero che rappresenta il primo id disponibile.                      Intero

#### ALGORITMO

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

#### CreaRelazioneBranoArtista()

##### INPUT

DBBranArtista	Database che contiene le relazione tra Brani ed Artisti, in cui inseriremo una nuova relazione.	DatabaseBranArtista
nuovoBrano	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= AggiungiRelazioneBranoArtista(campo idBrano di NuovoBrano,
        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=AggiungiRelazioneBranoArtista(campo idBrano di NuovoBrano,
                    CercaUltimoArtista(DBArtista), DBBranArtista)
            FINE
        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 <> ARTISTASENTINELLA AND Trovato=0)
    SE(campo nome di DBArtista in posizione i=NomeArtista)
        ALLORA
            Trovato=1
            IDTrovato=i
        FINE
    i++
FINE
```

#### AggiungiRelazioneBranoArtista()

##### INPUT

idBrano	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= BRANOARTISTASENTINELLA

```

#### cercaPrimaRelazioneBranArtistaLibera()

##### INPUT

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

##### OUTPUT

i	Intero che rappresenta il primo id disponibile.	Intero
---	---	--------

#### ALGORITMO

```

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

```

#### CreaRelazioneBranGenere()

##### INPUT

DBBranGenere	Database che contiene le relazione tra Brani e Generi, in cui inseriremo una nuova relazione.	DatabaseBranGenere
nuovoBran	Bran a cui dovremo associare un genere.	Bran
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 <> GENERESENTINELLA AND Trovato=0)
    SE(campo nome di DBGenere in posizione i=NomeGenere)
        ALLORA
            Trovato=1
            IDTrovato=i
        FINE
    i++
FINE
```

### **AggiungiRelazioneBranoGenere()**

#### **INPUT**

idBrano	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 idBrano di DBBranoGenere in posizione primaPosizioneLibera = idBrano
DBBranoGenere in posizione primaPosizioneLibera +1= BRANOGENERESSENTINELLA
```

### **cercaPrimaRelazioneBranoGenereLibera()**

#### **INPUT**

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

#### **OUTPUT**

i	Intero che rappresenta il primo id disponibile.	Intero
---	---	--------

### **ALGORITMO**

```
i=0
MENTRE(DBBranoGenereInPosizioneI<>BRANOGENERESSENTINELLA)
    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')  
DBArtista In Posizione campo IdArtista Di NuovoArtista)=NuovoArtista  
DBArtista in Posizione (campo idArtista di NuovoArtista)+1 = ARTISTASENTINELLA

### 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 contenente tutti gli artisti da cui andremo a prendere l'id in cui è posizionato ARTISTASENTINELLA. DatabaseArtista

#### OUTPUT

UltimoID Posizione di ARTISTASENTINELLA Intero

#### ALGORITMO

UltimoID=0  
MENTRE(DBArtista in posizione UltimoID <> ARTISTASENTINELLA)  
    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. DatabaseAlbum  
NuovoAlbum Album da inserire nel database. 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')
DBAlbum In Posizione campo IdAlbum Di NuovoAlbum)=NuovoAlbum
DBAlbum in Posizione (campo idAlbum di NuovoAlbum)+1 = ALBUMSENTINELLA

```

### 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 ALBUMSENTINELLA.	DatabaseAlbum
---------	--	---------------

#### OUTPUT

UltimoID	Posizione di ALBUMSENTINELLA	Intero
----------	------------------------------	--------

#### ALGORITMO

```

UltimoID=0
MENTRE(DBAlbum in posizione UltimoID <> ALBUMSENTINELLA)
    UltimoID++
FINE

```

### StampaAlbum()

#### INPUT

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.	DatabaseGenere
NuovoGenere	Genere da inserire nel database.	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')
DBGenere In Posizione campo IdGenere Di NuovoGenere)=NuovoGenere
DBGenere in Posizione (campo idGenere di NuovoGenere)+1 = GENERESENTINELLA

```

### **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 GENERESENTINELLA. DatabaseGenere

#### **OUTPUT**

UltimoID Posizione di GENERESENTINELLA Intero

#### **ALGORITMO**

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

### **StampaGenere()**

#### **INPUT**

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')  
DBUtente In Posizione campo IdUtente Di NuovoUtente)=NuovoUtente  
DBUtente in Posizione (campo idUtente di NuovoUtente)+1 = UTENTESENTINELLA

### **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 è posizionato UTENTESENTINELLA.	DatabaseUtente
----------	--	----------------

#### OUTPUT

UltimoID	Posizione di UTENTESENTINELLA	Intero
----------	-------------------------------	--------

### ALGORITMO

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

### StampaUtente()

#### 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")
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")

DBPlaylist in posizione idPlaylist di nuovaPlaylist = nuovaPlaylist

### CreaPlaylist()

#### INPUT

DBPlaylist Database da cui prenderemo l'id.

IdUtente Id dell'utente che vuole creare la playlist.

DatabasePlaylist

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")

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 PLAYLISTSENTINELLA.

DatabasePlaylist

#### OUTPUT

UltimoID Posizione di PLAYLISTSENTINELLA

Intero

#### ALGORITMO

UltimoID=0

MENTRE(DBUtente in posizione UltimoID <> PLAYLISTSENTINELLA)

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= PLAYLISTBRANOSENTINELLA

### cercaPrimaRelazionePlaylistBranoLibera()

**INPUT**

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

**OUTPUT**

i	Intero che rappresenta il primo id disponibile.	Intero
---	---	--------

**ALGORITMO**

```

i=0
MENTRE(DBPlaylistBranInPosizioneI<>PLAYLISTBRANSENTINELLA)
    i=i+1
FINE

```

**ModificaBran()****INPUT**

idBran	Identificatore del brano da modificare.	Intero
Campo	Campo della struttura dati da modificare.	Intero
DBBran	DataBase contenente il brano da modificare.	DatabaseBran

**OUTPUT**

DBBran	DataBase contenente i Brani modificato.	DatabaseBran
--------	---	--------------

**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 DBBran in posizione idBran=titolo
    ALTRIMENTI
        SE(Campo=2)
            ALLORA
                STAMPA("Qual è la nuova durata?")
                durata=LeggiIntero()
                campo durata di DBBran in posizione idBran=durata
            ALTRIMENTI
                SE(Campo=3)
                    ALLORA
                        STAMPA("In che anno è uscito il brano?")
                        anno=LeggiIntero()
                        campo anno di DBBran in posizione idBran=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=LeggiStringa()
        campo titolo di DBAlbum in posizione idAlbum=titolo
ALTRIMENTI
SE(Campo=2)
ALLORA
    STAMPA("quando è uscito l'album?")
    ESEGUI
        anno=LeggiAnno()
    MENTRE(anno<1000 && anno>ANNOATTUALE)
        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 modificare	Intero
Campo	Campo della struttura dati da modificare	Intero

#### 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")
    nuovaPassword=LeggiStringa()
    campo password di DBUtente in posizione idUtente=nuovaPassword
FINE
FINE
```

### CancellaBrano()

#### INPUT

idBrano	ID del brano da cancellare.	Intero
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
DBBranoPlaylist	Database di BranoPlaylist da cui cancellare un elemento	DatabaseBranoPlaylist

#### OUTPUT

DBBrani Database di brani da cui è stato cancellato un elemento

DatabaseBrani

#### ALGORITMO

DBBrani=campo idBrano di (DBBrani in posizione idBrano)=-2  
DBBranoArtista=CancellaRelBranoArtista(idBrano, DBBranoArtista)  
DBBranoAlbum=CancellaRelBranoAlbum(idBrano, DBBranoAlbum)  
DBBranoGenere=CancellaRelBranoGenere(idBrano, DBBranoGenere)  
DBBranoPlaylist=CancellaRelBranoPlaylist(idBrano, DBBranoPlaylist)

#### CancellaArtista()

##### INPUT

idArtista ID dell'artista da cancellare  
DBArtisti Database di brani da cui cancellare un elemento  
DBBrani Database di brani da cui cancellare un elemento  
DBBranoArtista Database di BranoArtista da cui cancellare un elemento  
DBBranoAlbum Database di BranoAlbum da cui cancellare un elemento  
DBBranoGenere Database di BranoGenere da cui cancellare un elemento  
DBBranoPlaylist Database di BranoPlaylist da cui cancellare un elemento

Intero  
DatabaseArtisti  
DatabaseBrani  
DatabaseBranoArtista  
DatabaseBranoAlbum  
DatabaseBranoGenere  
DatabaseBranoPlaylist

##### 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(DBBranoArtista in posizione i!= BRANOARTISTASENTINELLA)  
SE(campo idArtista di DBBranoArtista in posizione i= idArtista)  
ALLORA  
    CancellaBrano(campo idBrano di DBBranoArtista in posizione i, DBBrani, DBBranoArtista, DBBranoAlbum,  
DBBranoGenere, DBBranoPlaylist)  
FINE  
i++  
FINE

#### CancellaGenere()

##### INPUT

idGenere ID del genere da cancellare  
DBGeneri Database di brani da cui cancellare un elemento  
DBBrani Database di brani da cui cancellare un elemento  
DBBranoArtista Database di BranoArtista da cui cancellare un elemento  
DBBranoAlbum Database di BranoAlbum da cui cancellare un elemento  
DBBranoGenere Database di BranoGenere da cui cancellare un elemento  
DBBranoPlaylist Database di BranoPlaylist da cui cancellare un elemento

Intero  
DatabaseGeneri  
DatabaseBrani  
DatabaseBranoArtista  
DatabaseBranoAlbum  
DatabaseBranoGenere  
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!= BRANOARTISTASENTINELLA)  
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  
DBPlaylist Database di playlist da cui cancellare un elemento  
DBBrani Database di brani da cui cancellare un elemento  
DBBranoArtista Database di BranoArtista da cui cancellare un elemento  
DBBranoAlbum Database di BranoAlbum da cui cancellare un elemento  
DBBranoGenere Database di BranoGenere da cui cancellare un elemento

Intero  
DatabasePlaylist  
DatabaseBrani  
DatabaseBranoArtista  
DatabaseBranoAlbum  
DatabaseBranoGenere



**OUTPUT**

DBPlaylist	Database di brani da cui è stato cancellato un elemento	DatabasePlaylist
DBBranoPlaylist	Database di BranoGenere da cui è stato cancellato un elemento	DatabaseBranoPlaylist

**ALGORITMO**

Campo idPlaylist di DBPlaylist in posizione idPlaylist=-2  
i=0  
MENTRE(DBBranoPlaylist in posizione i <> BRANOPLAYLISTSENTINELLA AND campo idPlaylist di DBBranoPlaylist in posizione i <> idPlaylist)  
i++  
FINE  
SE(DBBranoPlaylist in posizione i <> BRANOPLAYLISTSENTINELLA)  
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 <> BRANOARTISTASENTINELLA)  
i=i+1  
FINE  
SE(DBBranoArtista in posizione i !=BRANOARTISTASENTINELLA)  
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		

**ALGORITMO**

i=0  
MENTRE(campo idBrano di DBBranoAlbum in posizione i <> idBrano AND DBBranoAlbum <> BRANOALBUMSENTINELLA)  
i=i+1  
FINE  
SE(DBBranoAlbum in posizione i <>BRANOARTISTASENTINELLA)  
ALLORA campo idBrano di DBBranoAlbum in posizione i=-2  
campo idAlbum di DBBranoAlbum in posizione 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 <> BRANOGENERESSENTINELLA)  
i=i+1  
FINE  
SE(DBBranoGenere in posizione i <> BRANOARTISTASENTINELLA)

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 <>  
BRANOARTISTASENTINELLA)  
i=i+1  
FINE  
SE(DBBranoPlaylist in posizione i <> BRANOPLAYLISTSENTINELLA)  
ALLORA  
campo idBrano di DBBranoPlaylist in poszione i=-2  
FINE