

## Gruppo 2 - PSEUDOCODIFICA FUNZIONI DI INSERIMENTO

### FUNZIONE “ProceduraInserimentoDaBran”

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale verrà aggiunto il nuovo brano nel caso non sia già presente, insieme ad altre informazioni collegate al brano, come l'artista, l'album e il genere.	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato con nuovi elementi inseriti	Database	
VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
nuovobrano	Struttura d'appoggio per l'acquisizione dei dati del nuovo brano	Brano	
nuovoartista	Struttura d'appoggio per l'acquisizione dei dati dell'artista che canta il brano	Artista	
nuovoalbum	Struttura d'appoggio per l'inserimento dell'album in cui è contenuto il brano	Album	
esistenzabrano	Esito della ricerca per nome del brano, vale -1 se il titolo non è presente nel database, nel caso esista un brano con stesso titolo ricevuto in input, vale quanto la posizione del brano nel vettore “brano” di Database	intero	
esistenzartista	Esito della ricerca per nome d'arte del brano, vale -1 se il nome d'arte non è presente nel database, nel caso esista un artista con stesso nome d'arte di quello ricevuto in input, vale quanto la posizione dell'artista nel vettore “artista” di Database	intero	
esistenzalbum	Esito della ricerca per titolo di album, vale -1 se il nome non è presente nel database, nel caso esista un album con stesso titolo ricevuto in input, vale quanto la posizione dell'album nel vettore “album” di Database	intero	
genereinput	Genere ricevuto in input che si intende assegnare al brano	stringa	
ris	Esito della ricerca del genere all'interno del vettore genere di Database. Vale -1 se il genere non è presente e dunque non è valido, vale quanto l'id del genere se è valido	intero	

### ALGORITMO

Campo nome di nuovobrano <- ChiediInserimentoNomeBran()o()

campo NomeArte di nuovoartista <- ChiediInserimentoNomeArteArtista()

esistenzabran()o <- CercaEsistenzaNomeBran()o(campo brano di musica, campo nome di nuovobran()o)

esistenzartista <- CercaEsistenzaNomeArtista(campo artista di musica, campo NomeArte di nuovoartista)

SE (esistenzabran()o <> SENTINELLA) AND (esistenzartista <> SENTINELLA)

ALLORA

Stampa a video “Bran()o già presente”

ALTRIMENTI

SE (esistenzartista <> SENTINELLA)

ALLORA

Stampa a video "Completa inserimento dati bran()o"

Campo durata di nuovobran()o <- ChiediInserimentoDurataBran()o()

Campo anno di nuovobran()o <- ChiediInserimentoAnnoBran()o()

InserireBran()o(musica, nuovobran()o)

InserireBran()oArtista(musica, CalcolaMaxBran()o(campo brano di musica), valore del campo idArtista in posizione “esistenzartista” del vettore artista di musica)

FINE ALLORA

ALTRIMENTI

Stampa a video “ Completa inserimento dati bran()o ”

Campo durata di nuovobran()o <- ChiediInserimentoDurataBran()o()

Campo anno di nuovobran()o <- ChiediInserimentoAnnoBran()o()

InserireBran()o(musica, nuovobran()o)

Stampa a video “Completa l’inserimento dei dati dell’artista”

Campo nome di nuovoartista <- ChiediInserimentoNomeArtista()

campo cognome di nuovoartista <- ChiediInserimentoCognomeArtista()

campo linkBio di nuovoartista <- ChiediInserimentoLinkArtista()

InserireArtista(musica, nuovoartista)

InserireBran()oArtista(musica, CalcolaMaxBran()o(campo brano di musica), CalcolaMaxArtista(campo artista di musica))

FINE ALTRIMENTI

Stampa a video “ Inserire album d'appartenenza per il nuovo bran()o ”

campo titolo di nuovoalbum <- ChiediInserimentoTitoloAlbum()

esistenzalbum <- CercaEsistenzaTitoloAlbum(campo album di musica, campo titolo di nuovoalbum)

SE (esistenzalbum <> SENTINELLA)

ALLORA

InserireBran()oAlbum(musica, CalcolaMaxBran()o(campo brano di musica), valore del campo idAlbum in posizione

“esistenzalbum” vettore album di musica”)

ALTRIMENTI

Campo anno di nuovoalbum <- ChiediInserimentoAnnoAlbum()

InserireAlbum(musica, nuovoalbum)

InserireBran()oAlbum(musica, CalcolaMaxBran()o(campo brano di musica), CalcolaMaxAlbum(campo album di musica))

FINE ALTRIMENTI

Stampa a video "Inserire genere per il nuovo bran()o"

ESEGUI

genereinput <- stringa letto da tastiera

ris <- CercaEsistenzaGenere(campo genere di musica, genereinput)  
SE (ris = SENTINELLA)

ALLORA

Stampa a video "Inserire un genere valido"

FINE ALLORA

FINE SE

FINCHE' ris = SENTINELLA

InserireBranoGenere(musica, CalcolaMaxBrano(campo brano di musica), ris)

FINE ALTRIMENTI

FINE SE

#### FUNZIONE "ProceduraInserimentoAlbum"

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale si intende inserire un nuovo album	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato dall' inserimento dell' album	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
nuovoalbum	Struttura d'appoggio per l'acquisizione dei dati del nuovo album	Album	
esistenzalbum	Esito della ricerca per titolo dell' album , vale -1 se il titolo non è presente nel database, nel caso esista un album avente lo stesso titolo di ricevuto in input, vale quanto la posizione del vettore album di musica nel quale si trova	intero	

#### ALGORITMO

Campo titolo di nuovoalbum <- ChiediInserimentoTitoloAlbum()

esistenzalbum <- CercaEsistenzaTitoloAlbum(campo album di musica, campo titolo di nuovoalbum)

SE (esistenzalbum <> SENTINELLA)

ALLORA

Stampa a video "Album gia' presente! Non è possibile avere album omonimi"

ALTRIMENTI

Stampa a video "Completa inserimento album"

Campo anno di nuovoalbum <- ChiediInserimentoAnnoAlbum()

InserireAlbum(musica, nuovoalbum)

#### FUNZIONE “ ProceduralInserimentoArtista “

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale si intende inserire un nuovo artista	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato	Database	
	dall’inserimento del nuovo artista		

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
nuovortista	Struttura d’appoggio per l’inserimento dei dati del nuovo artista	Artista	
esistenzartista	Variabile che vale -1 se l’artista non è presente nel Database, altrimenti ha lo stesso valore della posizione dell’artista cercato nel vettore artista di Database	intero	

#### ALGORITMO

ChiediInserimentoNomeArteArtista(campo NomeArte di nuovoartista)

esistenzartista <- CercaEsistenzaNomeArtista(campo artista di musica, campo NomeArte di nuovoartista)

SE (esistenzartista <> SENTINELLA)

ALLORA

Stampa a video “Artista già presente!” ALTRIMENTI

campo nome di nuovoartista <- ChiediInserimentoNomeArtista()  
 campo cognome di nuovoartista <- ChiediInserimentoCognomeArtista() campo linkBio di nuovoartista  
 <- ChiediInserimentoLinkArtista() InserireArtista(musica, nuovoartista)

#### FUNZIONE “ InserireBranco ”

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale verrà aggiunto il nuovo brano	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato dall’inserimento di un nuovo brano	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il campo “brano” di	intero	>=0

	musica		
inserito	Variabile a cui viene assegnato 1 nel momento in cui il brano viene inserito, altrimenti vale 0	booleano	0 OR 1

## ALGORITMO

i <- 0

inserito <- 0

ESEGUI

SE (campo idBrano dell'i-esimo elemento di brano in musica = SENTINELLA ) AND (inserito = 0) ALLORA

campo "idBrano" dell'i-esimo elemento di brano di musica <- 1 + CalcolaMaxBrano(campo brano di musica)

campo "nome" dell'i-esimo elemento di brano di musica <- campo nome di nuovo

campo "durata" dell'i-esimo elemento di brano di musica <- campo durata di nuovo

campo "anno" dell'i-esimo elemento di brano di musica <- campo anno di nuovo

campo "ascolti" dell'i-esimo elemento di brano di musica <- campo ascolti di nuovo  
inserito <- 1

FINE ALLORA

FINE SE

i <- i+1

FINCHE inserito <> SENTINELLA

## FUNZIONE "InserireAlbum "

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale si intende inserire il nuovo album	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato dall'inserimento del nuovo album	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettore "album" di musica	intero	>=0
inserito	Variabile booleana a cui viene assegnato 1 nel momento in cui l'album viene inserito nell'album, altrimenti vale 0	booleano	0 OR 1

## ALGORITMO

i <- 0

inserito <- 0

ESEGUI

SE (campo "idAlbum" dell'esimo elemento di album in musica = SENTINELLA) AND (inserito = 0)  
ALLORA

campo idAlbum dell'i-esimo elemento di album in musica <- 1 + CalcolaMaxAlbum(campo album di musica);

campo titolo dell'i-esimo elemento di album in musica <- campo titolo di nuovo

campo anno dell'i-esimo elemento di album in musica <- campo anno di nuovo

```

        inserito <- 1

    FINE ALLORA

    i <- i+1
    FINCHE inserito <> 1

```

#### FUNZIONE “InserireArtista”

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database nel quale si intende inserire il nuovo artista	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database modificato dall’inserimento del nuovo artista	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettore artista di Database	intero	$\geq 0$
inserito	Variabile a cui viene assegnato 1 nel momento in cui l’artista viene inserito, altrimenti vale 0	intero	0 OR 1

#### ALGORITMO

```

i <- 0

inserito <- 0
ESEGUI

    SE (campo “idArtista” dell’i-esimo elemento di artista in musica = SENTINELLA) AND (inserito=0)
    ALLORA

        campo “idArtista” dell’i-esimo elemento di artista in musica = 1+CalcolaMaxArtista(campo artista di musica);

        campo “nome” dell’i-esimo elemento di artista in musica <- campo nome di nuovo

        campo “cognome” dell’i-esimo elemento di artista in musica <- campo cognome di nuovo
        campo “NomeArte” dell’i-esimo elemento di artista in musica <- campo NomeArte di nuovo);

        campo “linkBio” dell’i-esimo elemento di artista in musica <- campo linkBio di nuovo

        inserito <- 1

    FINE ALLORA

    FINE SE

    i < i+1
    FINCHE inserito <> 1

```

#### FUNZIONE “InserireBranoArtista”

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database contenente tutte le informazioni del sistema	Database	
chiavebrano	Numero indicante l’id del brano da inserire nel database, da collegare a chiaveartista	intero	$\geq 0$
chiaveartista	Numero indicante l’id dell’artista da inserire nel database, da collegare a chiavebrano	intero	$\geq 0$

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database aggiornato contenente il nuovo collegamento	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettore BranoArtista di musica	Intero	$\geq 0$

#### ALGORITMO

i <- 0

MENTRE (campo idBrano dell'iesimo elemento di branoArtista di musica <> SENTINELLA)  
i <- i+1

FINE MENTRE

campo idBrano dell'iesimo elemento di branoArtista di musica <- chiavebrano  
campo idArtista dell'iesimo elemento di branoArtista di musica <- chiaveartista

#### FUNZIONE "InserireBranoAlbum "

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database contenente tutte le informazioni del sistema	Database	
chiavebrano	Numero indicante l'id del brano da inserire nel database, da collegare a chiavealbum	intero	$\geq 0$
chiavealbum	Numero indicante l'id dell'album da inserire nel database, da collegare a chiavebrano	intero	$\geq 0$
OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database aggiornato contenete il nuovo collegamento	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettore BranoAlbum di musica	Intero	$\geq 0$

#### ALGORITMO

i <- 0

MENTRE (campo idBrano dell'iesimo elemento di branoAlbum di musica <> SENTINELLA)  
i <- i+1

FINE MENTRE

campo idBrano dell'iesimo elemento di branoAlbum di musica <- chiavebrano  
campo idAlbum dell'iesimo elemento di branoAlbum di musica <- chiavealbum

#### FUNZIONE "InserireBranoGenere "

INPUT	DESCRIZIONE	TIPO	VINCOLI
-------	-------------	------	---------

musica	Database contenente tutte le informazioni del sistema	Database	
chiavebrano	Numero indicante l'id del brano da inserire nel database, da collegare a chiavegenere	intero	>=0
chiavegenere	Numero indicante l'id del genere da inserire nel database, da collegare a chiavebrano	intero	>=0

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database aggiornato contenete il nuovo collegamento	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettore BranoGenere di musica	Intero	>=0

#### ALGORITMO

i<-0

MENTRE (campo idBrano dell'iesimo elemento di branoGenere di musica <> SENTINELLA)  
i <- i+1

FINE

campo idBrano dell'iesimo elemento di branoGenere di musica <- chiavebrano  
campo idGenere dell'iesimo elemento di branoGenere di musica <- chiavegenere

#### FUNZIONE “ ChiediInserimentoNomeBrano “

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome del brano letto da tastiera	stringa	<= STRING MAX

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome del brano	stringa	<= STRING MAX

#### ALGORITMO

Stampa a video “ Inserire nome del brano “

nuovo <- Leggere stringa letto da tastiera

fine

#### FUNZIONE “ ChiediInserimentoDurataBrano “

INPUT	DESCRIZIONE	TIPO	VINCOLI
durata	Durata del brano letto da tastiera	intero	>0

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
durata	Durata del brano da inserita dall'utente	intero	>0

#### ALGORITMO

ESEGUI

Stampa a video "Inserire durata del brano del brano (in secondi) “



```

durata <- leggere valore intero datastiera

SE (ControlloValidataDurata(durata)==1)
    Stampa a video "durata non valida"
FINE SE
FINCHE' (ControlloValidataDurata(durata)==1)

```

**FUNZIONE “ ChiediInserimentoAnnoBranò “**

INPUT	DESCRIZIONE	TIPO	VINCOLI
anno	anno del brano letto da tastiera	intero	>1000

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
anno	anno del brano	intero	>1000

**ALGORITMO**

```

ESEGUI

Stampa a video "Inserire anno di pubblicazione del brano:"
anno <- leggere valore intero da tastiera

SE (ControlloValidataAnno(anno)==1)
    Stampa a video "anno non valido"
FINE SE
FINCHE' (ControlloValidataAnno(anno)==1)

```

**FUNZIONE “ ChiediInserimentoNomeArteArtista “**

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome d’arte dell’artista letto da tastiera	stringa	<= STRING MAX
OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome d’arte dell’artista	stringa	<= STRING MAX

**ALGORTIMO**

```

Stampa a video “Inserire nome d’arte dell' artista”
nuovo <- Leggere stringa letto da tastiera

```

**FUNZIONE “ ChiediInserimentoNomeArtista”**

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome dell’artista letto da tastiera	stringa	<= STRING MAX

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome dell’artista	stringa	<= STRING MAX

**ALGORTIMO**

```

Stampa a video “Inserire nome dell' artista”

nuovo <- Leggere stringa letto da tastiera

```

**FUNZIONE “ ChiediInserimentoCognomeArtista “**

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	cognome dell'artista letto da tastiera	stringa	<= STRING MAX

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	cognome dell'artista	stringa	<= STRING MAX

#### ALGORTIMO

Stampa a video "Inserire cognome dell' artista"

nuovo <- Leggere stringa letto da tastiera

#### FUNZIONE " ChiediInserimentoLinkArtista "

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Link biografia dell'artista letto da tastiera	stringa	<= STRING MAX

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Link biografia dell'artista	stringa	<= STRING MAX

#### ALGORTIMO

Stampa a video "Inserire link biografia dell'artista"

nuovo <- Leggere stringa letto da tastiera

#### FUNZIONE " ChiediInserimentoAnnoAlbum "

INPUT	DESCRIZIONE	TIPO	VINCOLI
anno	anno dell'album che deve inserire l'utente	intero	>1000

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
anno	anno dell'album inserito dall'utente	intero	>1000

#### ALGORITMO

ESEGUI

Stampa a video "Inserire anno di pubblicazione dell'album: "

anno <- leggere valore intero da tastiera

SE (ControlloValidataAnno(anno) = 1)

Stampa a video "anno non valido"

FINE SE

FINCHE' (ControlloValidataAnno(anno) = 1)

#### FUNZIONE " ChiediInserimentoTitoloAlbum "

INPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome del brano letto da tastiera	stringa	<= STRING MAX

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
nuovo	Nome del brano	stringa	<= STRING MAX

#### ALGORITMO

Stampa a video " Inserire titolo dell'album: "

nuovo <- Leggere stringa letto da tastiera

#### FUNZIONE "AggiungerePlaylist"

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura contenente tutte le informazioni del sistema	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura aggiornata contenente tutte le informazioni del sistema	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
NuovoPlaylist	struttura contenente le informazioni della playlist da inserire in musica	Playlist	
esito	variabile che sarà vera se ho effettuato correttamente l'inserimento, falsa altrimenti	booleana	0 OR 1

#### ALGORITMO

esito <- 0

NuovoPlaylist <- LeggerePlaylistDaTastiera(NuovoPlaylist)

esito <- RicercaPlaylistSimili(BaseDati, NuovoPlaylist)

SE (esito = 0)

ALLORA BaseDati <- InserirePlaylist(BaseDati, NuovoPlaylist)  
BaseDati <- ScrivereUltimoEsito(BaseDati, OP\_RIUSCITA)  
Stampa("Inserimento effettuato!")

ALTRIMENTI

BaseDati <- ScrivereUltimoEsito(BaseDati, INS\_FALLITO)  
Stampa a video "Inserimento annullato!"

#### FUNZIONE "InserirePlaylist"

INPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	struttura contenente tutte le informazioni del sistema in cui inserire NuovoPlaylist	Database	
NuovoPlaylist	struttura contenente tutte	Playlist	

	le informazioni del sistema in cui inserire NuovoPlaylist		
--	-----------------------------------------------------------	--	--

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	struttura contenente tutte le informazioni aggiornate del sistema	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	numero che indica l'iesimo elemento del vettore playlist di musica considerato	intero	$\geq 0$
vettorePlaylist	vettore dove memorizzare il vettore di playlist di BaseDati e NuovoPlaylist, alla fine andrà inserita in BaseDati	Array di playlist	

#### ALGORITMO

i <- 0

LeggereVettorePlaylist(vettorePlaylist, BaseDati)

MENTRE(LeggereIdPlaylist(LeggerePlaylist(vettorePlaylist,i)) <> SENTINELLA)

i <- i+1

FINE MENTRE

SE i = 0

ALLORA i<- i+1

ScrivereIdUtentePlaylist(NuovoPlaylist, LeggereUtenteCorrente(BaseDati))

ScrivereIdPlaylist(NuovoPlaylist, LeggereIdPlaylist(LeggerePlaylist(vettorePlaylist,i-1)) + 1)

ScriverePlaylist(vettorePlaylist, i, NuovoPlaylist)

ScrivereVettorePlaylist(vettorePlaylist, BaseDati)

#### FUNZIONE "InserirePlaylistBranò"

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura contenente tutte le informazioni del sistema		
chiavebrano	numero indicante l'id del brano da associare alla playlist in BaseDati	intero	$\geq 0$
chiaveplaylist	numero indicante l'id della playlist da associare al brano in BaseDati	intero	$\geq 0$

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura contenente tutte le informazioni aggiornate del sistema	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	numero che indica	intero	$\geq 0$

	l'iesimo elemento del vettore branoPlaylist di BaseDati		
--	---------------------------------------------------------------	--	--

#### ALGORITMO

i <- 0

MENTRE (campo "idBranò" dell' i-esimo elemento del vettore PlaylistBranò di musica <> SENTINELLA)

i<-i+1

FINE

campo "idBranò" dell' i-esimo elemento del vettore PlaylistBranò di musica <- chiavebrano

campo "idPlaylist" dell' i-esimo elemento del vettore PlaylistBranò di musica <- chiaveplaylist

#### FUNZIONE "AggiungereUtente"

INPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	il database nel quale verrà inserito il nuovo utente	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	base di dati aggiornata dalle informazioni dell'utente	Database	Campo idUtente del dato Utente >0

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
NuovoUtente	Struttura di tipo utente da inserire nel database dopo averne compilato i campi	Utente	Campo id NuovoUtente>0
esito	Esito della ricerca di altri utenti con dettagli simili: se esito =0, si procede ad inserire i dati nel database, se =1, il programma si ferma e non inserisce il nuovo utente	Intero	=0 OR =1

#### ALGORITMO

esito <-0

NuovoUtente <- LeggiUtenteDaTastiera()

esito <- RicercaUtentiSimili(BaseDati, NuovoUtente)

SE (esito=0)

InserireUtente(BaseDati, NuovoUtente)

ScrivereUltimoEsito(BaseDati, OP\_RIUSCITA)

Stampa a video "inserimento effettuato!"

ALTRIMENTI

ScrivereUltimoEsito(BaseDati, INS\_FALLITO)

Stampa a video "Inserimento annullato!"

#### **FUNZIONE "InserireUtente"**

INPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	Database in cui scrivere il nuovo utente	Database	
NuovoUtente	Nuovo utente da aggiungere al database	Utente	Campo idUtente di utente=0

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	Database aggiornato in cui è inserito il nuovo utente	Database	

#### **ALGORITMO**

i <- 0

LeggereVettoreUtente(vettoreUtente, BaseDati)

MENTRE (LeggereIdUtente(LeggereUtente(vettoreUtente,i)) <> SENTINELLA)

    i <- i+1

}

SE (i=0)

ALLORA i <- i+1

ScrivereIdUtente(NuovoUtente, LeggereIdUtente(LeggereUtente(vettoreUtente, i-1)) + 1)

ScrivereUtente(vettoreUtente, i, NuovoUtente)

ScrivereVettoreUtente(vettoreUtente, BaseDati)

#### **FUNZIONE "AggiungereGenere"**

INPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	il database nel quale verrà inserito il nuovo utente	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
BaseDati	base di dati modificata	Database	Campo idGenere del dato Genere >0

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
esito	Esito della ricerca di altri generi con lo stesso nome se esito =0, si procede ad inserire i dati nel database, se =1, il programma si ferma e non inserisce il nuovo genere	Intero	=0 OR =1
NuovoGenere	Struttura per l'acquisizione dei dati del nuovo genere	Database	Campo idGenere >0

#### ALGORITMO

esito<-0

NuovoGenere <- LeggiGenereDaTastiera()

esito <- RicercaGeneriSimili(BaseDati, NuovoGenere)

SE (esito=0)

InserireGenere(BaseDati, NuovoGenere)  
 ScrivereUltimoEsito(BaseDati, OP\_RIUSCITA)  
 Stampa a video ("inserimento effettuato!")

ALTRIMENTI

ScrivereUltimoEsito(BaseDati, INS\_FALLITO)  
 Stampa a video "inserimento annullato!"

#### FUNZIONE "InserireGenere"

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database in cui scrivere il nuovo utente	Database	
NuovoGenere	Nuovo genere da aggiungere al database	Utente	Campo idGenere di utente=0

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	Database aggiornato in cui è inserito il nuovo genere	Database	

LAVORO	DESCRIZIONE	TIPO	VINCOLI
i	Indice per scandire il vettoreGenere	Database	
vettoreGenere	Vettore di genere dove inserire il campo genere di BaseDati		

#### ALGORITMO

i <- 0

```

vettoreGenere <- LeggereVettoreGenere()

MENTRE (LeggereIdGenere(LeggereGenere(vettoreGenere, i) <> SENTINELLA)

    i<-i+1

FINE MENTRE

ScrivereIdGenere(NuovoGenere, LeggereIdGenere(LeggereGenere(vettoreGenere,i-1)) +1)

ScrivereGenere(vettoreGenere, i , NuovoGenere)

ScrivereVettoreGenere(vettoreGenere, BaseDati)

```

#### FUNZIONE “Inserire”

INPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura contenente tutte le informazioni del sistema, nel quale inserire uno o più elementi	Database	

OUTPUT	DESCRIZIONE	TIPO	VINCOLI
musica	struttura contenente tutte le informazioni del sistema aggiornate	Database	

VARIABILI LAVORO	DESCRIZIONE	TIPO	VINCOLI
scelta	valore indicante il tipo di inserimento che si vuole effettuare nel Database: 1 per inserire brani, 2 per inserire album, 3 per inserire artista, 4 per inserire genere, 5 per inserire utente, 0 per non fare nulla		

#### ALGORITMO

SE (LeggereAdmin( RicercareUtenteld(LeggereVettoreUtente(musica), LeggereUtenteCorrente(musica)) ) = 0)

Allora stampa a video “Non hai i privilegi amministrativi per compiere questa azione”

ALTRIMENTI

Stampa a video “cosa inserire? 1 per inserire brani, 2 per inserire album, 3 per inserire artista, 4 per inserire genere, 5 per inserire utente, 0 per uscire”

ESEGUI

scelta <- LeggereIntero()

FINCHE’ (scelta <> 1) AND (scelta <> 2) AND (scelta <> 3) AND (scelta <> 4) AND (scelta <> 5) AND (scelta<>0)

MENTRE (scelta <>0)

SE(scelta = 1)

ALLORA musica <- ProceduraInserimntoDaBrano((musica)

ALTRIMENTI

SE scelta=2)

ALLORA musica <- ProceduraInserimntoDaAlbum(musica)

ALTRIMENTI

SE(scelta=3)

ALLORA musica <- ProceduraInserimntoDaArtista(musica)

ALTRIMENTI

SE (scelta = 4)



```

                                ALLORA musica <- AggiungereGenere(musica)
                                ALTRIMENTI
                                    SE (scelta = 5)
                                        ALLORA musica <- AggiungereUtente(musica)
                                        FINE
                                    FINE
                                FINE
                            FINE
                        FINE
                    Stampa a video "inserire altro? 1 per inserire brani, 2 per inserire album, 3 per inserire artista, 4 per inserire genere, 5 per
                        inserire utente, 0 per uscire"
                    ESEGUI
                        scelta <- LeggereIntero()
                        FINCHE' (scelta <> 1) AND (scelta <> 2) AND (scelta <> 3) AND (scelta <> 4) AND (scelta <> 5) AND (scelta<>0)
                            FINE
                    FINE
                FINE
            FINE
        FINE
    FINE

```