

```

/+++++ ARCHITETTURA DEGLI ELABORATI ++++++ \
\ /
/ Progetto: Facedoor \
\ /
/ Data: 22.08.2015 \
\ Nome e cognome: Belloni Francesco /
/ Matricola: 725676 \
\+++++ \

```

1. Pseudocodice -----

Il codice ha diverse funzioni, ecco in pseudo codice alcune delle più significative:

- Compara stringhe/sub stringhe:

Confronta carattere per carattere due stringhe. Ad ogni confronto si aumentano dei contatori in funzione se il confronto è andato a buon fine oppure no. La funzione esce una volta che si è raggiunto il limite dei caratteri uguali o diversi oppure se in entrambe le stringhe si incontra il carattere di NULL (fine stringa).

<pre> START_COMPARE_STRING:  int i int i3 (# conta i caratteri in comune) int i4 (# conta quante volte "sbaglio") int limite_substring = 3 # Dopo quante volte sbagliate / giuste esco dal ciclo start:  int x = charAT(stringaUno, i) int y = chatAt(stringaDue, i)  if x&gt;Y i4++ if x&lt;y i4++  i3++  i++  if ( charAT(stringaUno, i) == NULL AND chatAt(stringaDue, i) == NULL ) return 1 // stinghe uguali  if i4== limite_substring return 0 // substringhe diverse  if i3== limite_substring return 1 // substringhe uguali  goto start; </pre>	<pre> START_COMPARE_STRING:  li \$t3, 0 # conta i caratteri in comune li \$t4, 0 # conta quante volte "sbaglio" li \$t5, 3 # Dopo quante volte sbagliate / giuste esco dal ciclo  COMPARA_STRINGHE:  lbu \$t0,0(\$a0)# Load a byte from the first source string. lbu \$t1,0(\$a1) # Load a byte from the second source string.  # Compare the two characters loaded. blt\$t0,\$t1,COMPARE_CARATTERI_DIVERSI bgt\$t0,\$t1,COMPARE_CARATTERI_DIVERSI  addi \$t3,\$t3,1 # Aumenta "carattere uguale" beq \$t3, \$t5, COMPARE_EXIT_STRINGHE_UGUALI  # Both characters are the same. beqz \$t1,COMPARE_EXIT_STRINGHE_UGUALI # Return 0 if both are equal to NULL  COMPARE_CONTINUE: # o/w Increment both addresses and continue. addi \$a0,\$a0,1 addi \$a1,\$a1,1 b COMPARA_STRINGHE  COMPARE_CARATTERI_DIVERSI: # Aumenta "carattere diverso" addi \$t4,\$t4,1 beq \$t4, \$t5,  COMPARE_EXIT_STRINGHE_DIVERSE </pre>
--	---

	<pre> j COMPARE_CONTINUE  COMPARE_EXIT_STRINGHE_UGUALI:     li \$v0,1          # vo a 1 uguale     jr \$ra  COMPARE_EXIT_STRINGHE_DIVERSE:     li \$v0,2          # vo a 2 --&gt; stringhe diverse     jr \$ra </pre>
--	---

#### Menu Cerca Utente:

Inizialmente controllo che il profilo passato sia privato poiché se lo è esco subito. Invece se il profilo è pubblico proseguo con la funzione.

Il passo successivo è confrontare il nome dell'utente con il nome da cercare (attraverso la funzione di confronto delle stringhe descritta in precedenza). Se sono uguali salto il confronto con il cognome e stampo il risultato. Se invece i due nomi non sono simili controllo il cognome.

In questa funzione c'è un registro (s4) che viene settato ad 1 quando si trova un risultato. Successivamente infatti viene controllato lo stato di questo registro e se si trova ancora a 0 viene stampata la frase "nessun risultato trovato".

```
Function boolean Cerca(Classe Utente, String NomeCercato, String CognomeCercato) {
```

```
bool trovato=false;
```

```
If Utente.privacy=='Privato' Return False
```

```
if Confronta(Utente.Nome, NomeCercato) ==true goto Stampa
```

```
if Confronta(Utente.Cognome, CognomeCercato) ==true goto Stampa
```

```
return false
```

```
Stampa:
```

```
Printf("Nome: $1 ", Utente.Nome );
```

```
Printf("Cognome: $1 ", Utente.Cognome );
```

```
return true; }
```

#### Controlla massimo utenti:

Questa funzione controlla che non abbia raggiunto il numero massimo di utenti pubblici/privati in base al parametro che gli passo.

```
# call the controlla_max_utenti_privati_pubblici function
# ritorna 1 in a0 se sono continuo con la registrazione
# ritorna 2 in a0 se devo uscire (messaggi di errori già stampati nella funzione)
# nella funzione in $s4 ci sarà il valore 0 [pubblico] o privato [1]
```

```

functio n int controlla_max_utenti_privati_pubblici(int Pubblico){

if Pubblico== 0
    if GLOBAL_N_PUBLIC_USER < 5
        return 1
    else
        printf("raggiunto il numero massimo di utenti pubblici)
        return 2
}else
{
    if GLOBAL_N_PRIVATE_USER < 5
        return 1
    else
        printf("raggiunto il numero massimo di utenti privati)
        return 2
}

```

Non tutte le funzioni del progetto sono state inserite come pseudocodice poiché tutto il codice è stato commentato e quindi si è deciso di entrare nel dettaglio delle più significative.

## 2. Analisi dei Registri -----

```

#li $s4,0 # Uso s4 per settare se trovo l'utente nella parte cerca utente [DA NON MODIFICARE NELLA
SUA PROCEDURA]

li $s4,0 # Uso s4 per settare se l'utente è pubblico / privato nella procedura di creazione
utente [DA NON MODIFICARE NELLA SUA PROCEDURA]

li $s5,7 # mi segno quanti utenti già salvati ci [DA NON MODIFICARE]
li $s6,4 # mi segno quanti utenti pubblici ci sono [DA NON MODIFICARE]
li $s7,3 # mi segno quanti utenti privati ci sono [DA NON MODIFICARE]

```

I registri utilizzati globalmente che nessuna procedura deve "sporcare" sono 3: S5 che mi segnala quanti utenti ci sono salvati centralmente in Facedoor, S6 che mi indica quanti utenti pubblici ed s7 che mi indica quanti utenti privati.

Il registro s4 invece viene usato nelle varie funzioni che ci sono nel processo di ricerca: viene inizializzato inizialmente e non deve essere sporcato nelle procedure che interessano il processo di ricerca se non quando un utente è stato trovato. Il registro s4 viene usato anche nella procedura di registrazione e mi segna quale tipo di privacy voglio (pubblico/privato).

### 2.1 Alcune funzioni del programma:

```

STAMPA_UTENTE
la $s0,UtenteUno # indirizzo memoria utentel
jal STAMPA_UTENTE
input: in $S0 salvo l'indirizzo di memoria dell'utente da stampare
return: niente, perchè stampa a video le informazioni

```

START\_COMPARE\_STRING:

input: \$a0 Prima stringa, \$a1 seconda stringa

return: in \$s0=1 se uguali \$s0=2 se diverse

STAMPA\_UTENTE:

input: in \$s0 l'indirizzo dell'utente

return: niente, stampa a video le informazioni se l'utente è pubblico

CONTROLLA\_MAX\_UTENTI\_PRIVATI\_PUBBLICI:

input: in \$a0=0 (se l'utente che voglio inserire è pubblico), \$a1=1 se è privato

return: in \$v0=0 se non ho raggiunto il limite massimo di utenti

AUMENTA\_CONTATORI:

input: in \$s4=1 se l'utente è privato, \$s4=0 se è pubblico

return: niente aumenta i "registri globali" che mi indicano il numero di iscritti

2.1 Alcune procedure che non chiedono nessun input, ovvero salti incondizionati che eseguono "semplici" compiti (stampare a video, richiedere informazioni, ...):

scelta\_CreaProfilo:

scelta\_CercaProfilo:

scelta\_CancellaProfilo:

crea\_utente... :

cancella\_utente... :

fine:

.....

### 3. Vincoli -----

Il progetto Facedoor prevede la gestione di 10 profili, ma per accorciare le linee di codice e soprattutto per la fase di debug 7 profili sono stati già inseriti manualmente. Di questi 4 pubblici e 3 privati.

La scelta di accorciare le linee di codice è dettata dal fatto che le funzioni che gestiscono cancellazione ed inserimento sono scritte unicamente per l'utente 8, l'utente 9 e l'utente 10. Visto che queste 3 procedure sono identiche ma cambia solamente per l'indirizzamento la difficoltà del progetto rimane la stessa senza appesantire il progetto con codice uguale. Ovviamente però tutte le altre funzioni presenti nel progetto agiscono in maniera più astratta agendo sull'indirizzo passato nei registri appropriati.

#### 4. Codice Assembly -----

.data

str\_welcome: .asciiiz "Welcome to facedoor, il primo programma che ti insegna a programmare in Assembly!"

str\_Errore: .asciiiz "\n\n Hai sbagliato ad inserire la scelta ! \n\n"

str\_Errore\_Max\_public\_user: .asciiiz "\n\n Mi dispiace, e' stato raggiunto il limite di profili pubblici!\n Se vuoi puoi creare un profilo privato ... \n\n"

str\_Errore\_Max\_private\_user: .asciiiz "\n\n Mi dispiace, e' stato raggiunto il limite di profili privati!\n Se vuoi puoi creare un profilo pubblico ... \n\n"

str\_Errore\_Max\_user: .asciiiz "\n\n Mi dispiace, e' stato raggiunto il limite di profili!\n Torna tra qualche settimana : ( ... \n\n"

str\_Spazio: .asciiiz " "

str\_ACapo: .asciiiz "\n"

str\_Pubblico: .asciiiz "111111"

str\_Privato: .asciiiz "0000000"

str\_MenuPrincipale: .asciiiz "\n\n -- MENU PRINCIPALE -- \n\n0-Crea Profilo\n1-Cerca Profilo\n2-Cancella Profilo\n3-Mostra tutti Profili\n4-Esci\n\n"

str\_MenuCreaProfilo: .asciiiz "\n\n -- CREA PROFILO --\n\n"

str\_MenuCercaProfilo: .asciiiz "\n\n -- CERCA PROFILO --\n\n"

str\_MenuCancellaProfilo: .asciiiz "\n\n -- CANCELLA PROFILO --\n\n"

str\_MenuMostraProfilo: .asciiiz "\n\n -- MOSTRA TUTTI I PROFILI --\n\n"

str\_MenuEsciProfilo: .asciiiz "\n\n -- ESCI --\n\nInserisci il nome\n\n"

str\_SottoMenuCreaProfilo: .asciiiz "\n\nUtente creato con successo\n\n0-Modifica Profilo e 1-Torna al menu principale\n\n"

str\_SottoMenuCercaProfilo: .asciiiz "\n\n0-Torna al menu principale\n\n"

str\_SottoMenuCancellaProfilo\_cancellato: .asciiiz "\n\nProfilo cancellato con successo, ci dispiace che tu abbia lasciato facedoor :( \n\n"

str\_SottoMenuCancellaProfilo\_accessoristretto: .asciiiz "\n\nATTENZIONE!\nNon puoi cancellare un profilo che non hai inserito!\n\n"

\_str\_CercaUtenti\_nessunRisultato: .asciiz "\nNon ci sono risultati cercando questo nome e cognome :( "

str\_ComparaStringheUguali: .asciiz "\nAbbiamo un risultato: "

str\_ComparaStringheUgualiNome: .asciiz "\nAbbiamo un risultato: "

str\_ComparaStringheUgualiCognome: .asciiz "\nAbbiamo un risultato: "

str\_stampa\_nome: .asciiz "Nome:"

str\_stampa\_cognome: .asciiz "Cognome:"

str\_stampa\_userID: .asciiz "UserID:"

str\_stampa\_interessi: .asciiz "Interessi:"

str\_stampa\_email: .asciiz "Email:"

ciao: .asciiz "\nCiao Ciao"

UtenteUno: .word nome, cognome, interessi, userID, visibilita, email, password

nome: .asciiz "Paolino\n"

cognome: .asciiz "Paperino\n"

interessi: .asciiz "Dormire, pisolare e riposini vari!\n"

userID: .asciiz "DonalDuck\n"

visibilita: .asciiz "111111\n"

email: .asciiz "paperino@disney.it\n"

password: .asciiz "313\n"

UtenteDue: .word nome2, cognome2, interessi2, userID2, visibilita2, email2, password2

nome2: .asciiz "Gastone\n"

cognome2: .asciiz "Paperone\n"

interessi2: .asciiz "Viaggiare e corteggiare paperina!\n"

userID2: .asciiz "Fortunello\n"

visibilita2: .asciiz "000000\n"

email2: .asciiz "gastone@disney.it\n"

password2: .asciiz "1000cliente\n"

UtenteTre: .word nome3, cognome3, interessi3, userID3, visibilita3, email3, password3

nome3: .asciiz "Paperoga\n"

cognome3: .asciiz "Paperone\n"

interessi3: .asciiz "Combinare disastri :P\n"

userID3: .asciiz "PaparBat\n"

visibilita3: .asciiz "11111\n"

email3: .asciiz "Paperoga@papersera.it\n"

password3: .asciiz "paperoga\n"

UtenteQuattro: .word nome4, cognome4, interessi4, userID4, visibilita4, email4, password4

nome4: .asciiz "Paperina\n"

cognome4: .asciiz "Paperone\n"

interessi4: .asciiz "Farmi bella :) \n"

userID4: .asciiz "Paperetta\n"

visibilita4: .asciiz "11111\n"

email4: .asciiz "paperotta@disney.it\n"

password4: .asciiz "paperina\n"

UtenteCinque: .word nome5, cognome5, interessi5, userID5, visibilita5, email5, password5

nome5: .asciiz "Nonna Papera\n"

cognome5: .asciiz "Paperone\n"

interessi5: .asciiz "Farmi tante belle torte per i miei tanti nipoti\n"

userID5: .asciiz "FattoriaDellaNonna\n"

visibilita5: .asciiz "11111\n"

email5: .asciiz "nonna@fattoria.it\n"

password5: .asciiz "Mucca\n"

UtenteSei: .word nome6, cognome6, interessi6, userID6, visibilita6, email6, password6

nome6: .asciiz "Qui\n"

cognome6: .asciiz "Paperino\n"

interessi6: .asciiz "Giovani marmotte e giocare! \n"

userID6: .asciiz "Qui\n"

visibilita6: .asciiz "000000\n"

email6: .asciiz "qui@disney.it\n"

password6: .asciiz "nipotino1\n"

UtenteSette: .word nome7, cognome7, interessi7, userID7, visibilita7, email7, password7

nome7: .asciiz "Quo\n"

cognome7: .asciiz "Paperino\n"

interessi7: .asciiz "Giocare e giovani marmotte!\n"

userID7: .asciiz "Quo\n"

visibilita7: .asciiz "000000\n"

email7: .asciiz "quo@disney.it\n"

password7: .asciiz "nipotino2\n"

UtenteOtto: .word nome8, cognome8, interessi8, userID8, visibilita8, email8, password8

nome8: .space 256

cognome8: .space 256

interessi8: .space 256

userID8: .space 256

visibilita8: .space 256

email8: .space 256

password8: .space 256

UtenteNove: .word nome9, cognome9, interessi9, userID9, visibilita9, email9, password9

nome9: .space 256

cognome9: .space 256

interessi9: .space 256

userID9: .space 256

visibilita9: .space 256

email9: .space 256

password9: .space 256

UtenteDieci: .word nome10, cognome10, interessi10, userID10, visibilita10, email10, password10

nome10: .space 256

cognome10: .space 256



interessil0: .space 256

userID10: .space 256

visibilita10: .space 256

email10: .space 256

password10: .space 256

# PARTE CERCA

Str\_ChiediNome: .asciiz "Inserisci nome: "

Space\_Nome: .space 256

Str\_ChiediCognome: .asciiz "Inserisci cognome: "

Space\_Cognome: .space 256

# PARTE INSERISCI UTENTE

Str\_ChiediInteressi: .asciiz "Inserisci i tuoi interessi: "

Space\_Interessi: .space 256

Str\_ChiediUserID: .asciiz "Inserisci il tuo UserID: "

Space\_UserID: .space 256

Str\_ChiediPrivacy: .asciiz "Setta la tua privacy: [000] per un profilo privato o [111] per un profilo pubblico "

Space\_Privacy: .space 256

Str\_ChiediEmail: .asciiz "Inserisci email: "

Space\_Email: .space 256

Str\_ChiediPassword: .asciiz "Inserisci password: "

Space\_Password: .space 256

Str\_debug\_uno: .asciiz "\n debug 1 \n "

Str\_debug\_due: .asciiz "\n debug 2 \n "

Str\_debug\_tre: .asciiz "\n debug 3 \n "

```

# LINK UTILI

# http://cs.lamar.edu/faculty/foreman/MIPS_example_programs/StringEx.s

# http://www.cs.unibo.it/~solmi/teaching/arch_2004-2005/Assembly3.pdf

# http://stackoverflow.com/questions/4351500/help-trimming-string-input-of-its-return-character-mips (rimuovere alla
fine della stringa l'ultimo carattere)

.text

.globl main

main:

    # j stampa_utenti

    #li $s4,0 # Uso s4 per settare se trovo l'utente nella parte cerca utente [DA NON MODIFICARE NELLA SUA
PROCEDURA]

    li $s4,0 # Uso s4 per settare se l'utente è pubblico / privato nella procedura di creazione utente [DA NON
MODIFICARE NELLA SUA PROCEDURA]

    li $s5,7 # mi segno quanti utenti già salvati ci [DA NON MODIFICARE]

    li $s6,4 # mi segno quanti utenti pubblici ci sono [DA NON MODIFICARE]

    li $s7,3 # mi segno quanti utenti privati ci sono [DA NON MODIFICARE]

    la $a0,str_welcome # Stampa frase di benvenuto

    li $v0,4 # print_string

    syscall

MENU_PRINCIPALE:

    la $a0,str_MenuPrincipale # STAMPA MENU PRINCIPALE

    li $v0,4 # print_string

    syscall

```

```

li $v0,5 # read_int

syscall      # risultato in $v0

move $t0,$v0    # salva la scelta in t0

li $t1, 0
beq $t0, $t1, scelta_CreaProfilo # Vai al menu crea profilo

li $t1, 1
beq $t0, $t1, scelta_CercaProfilo # Vai al menu cerca profilo

li $t1, 2
beq $t0, $t1, scelta_CancellaProfilo # Vai al menu cancella profilo

li $t1, 3
beq $t0, $t1, scelta_MostraProfilo # Vai al menu mostra profili

li $t1, 4
beq $t0, $t1, fine # Vai al menu esci

la $a0, str_Errore      # STAMPA stringa errore se la scelta non è tra quelle di prima.
li $v0,4 # print_string
syscall

j MENU_PRINCIPALE # TORNO AL MENU PRINCIPALE!

```

```

# SEZIONE GESTISCI SCELTE MENU PRINCIPALE

```

```

scelta_CreaProfilo: # Crea profilo

```

```

la $a0, str_MenuCreaProfilo # spiega scelta
li $v0,4      # print_string
syscall

```

```
li $t0, 7 #carico il numero sette, se sono a sette vado a creare l'utente 8
```

```
beq $t0, $s5, crea_utente_otto
```

```
li $t0, 8 #carico il numero otto, se sono a sette vado a creare l'utente 9
```

```
beq $t0, $s5, crea_utente_nove
```

```
li $t0, 9 #carico il numero nove, se sono a sette vado a creare l'utente 10
```

```
beq $t0, $s5, crea_utente_dieci
```

```
li $t0, 10 #carico il numero dieci, il massimo numero di utenti disponibili
```

```
beq $t0, $s5, Err_MaxUser
```

```
j Continua_NO_Err_MaxUser # evito il messaggio di errore, saltando un paio di istruzioni dopo
```

```
Err_MaxUser:
```

```
la $a0, str_Errore_Max_user # spiega scelta
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
Continua_NO_Err_MaxUser:
```

```
j MENU_PRINCIPALE
```

```
scelta_CercaProfilo: # Cerca profilo
```

```
la $a0, str_MenuCercaProfilo # spiega scelta
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
j SCELTA_CERCA_UTENTE
```

```
scelta_CancellaProfilo: # Cancella profilo
```

```
la $a0, str_MenuCancellaProfilo # spiega scelta
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
j SCELTA_CANCELLA_UTENTE
```

```
scelta_MostraProfilo: # Mostra profili
```

```
la $a0, str_MenuMostraProfilo # spiega scelta
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
j STAMPA_UTENTI_PUBBLICI
```

```
# FINE SEZIONE GESTISCI SCELTE MENU PRINCIPALE
```

```
# STAMPA UTENTI PUBBLICI
```

```
STAMPA_UTENTI_PUBBLICI:
```

```
# UTENTE UNO
```

```
la $s0, UtenteUno # indirizzo memoria utente1
```

```
jal STAMPA_UTENTE
```

```
# UTENTE DUE
```

```
la $s0, UtenteDue # indirizzo memoria utente2
```

```
jal STAMPA_UTENTE
```

```
# UTENTE TRE
```

```
la $s0, UtenteTre # indirizzo memoria utente3
```

```
jal STAMPA_UTENTE
```

```
# UTENTE QUATTRO
```

```
la $s0, UtenteQuattro # indirizzo memoria utente4
```

```
jal STAMPA_UTENTE
```

```
# UTENTE CINQUE
```

```

la $s0,UtenteCinque # inirizzo memoria utente5

jal STAMPA_UTENTE

# UTENTE SEI

la $s0,UtenteSei # inirizzo memoria utente6

jal STAMPA_UTENTE

# UTENTE SETTE

la $s0,UtenteSette # inirizzo memoria utente7

jal STAMPA_UTENTE

# If $s is less than immediate, $t is set to one. It gets zero otherwise.

# Syntax: slti $t, $s, imm

# se il numero di iscritti e' meno di 7 salto direttamente alla fine

# se il numero di iscritti e' meno di 8 salto direttamente alla fine

slti $t1, $s5, 8 # meno di 8 ovvero 7

li $t0, 1

beq $t0, $t1, salta_stampa_utente

# UTENTE OTTO

la $s0,UtenteOtto # inirizzo memoria utente8

jal STAMPA_UTENTE

# se il numero di iscritti e' meno di 9 salto direttamente alla fine

slti $t1, $s5, 9 # meno di 10 ovvero 9

li $t0, 1

beq $t0, $t1, salta_stampa_utente

# UTENTE NOVE

la $s0,UtenteNove # inirizzo memoria utente9

jal STAMPA_UTENTE

```

```
# se il numero di iscritti e' meno di 10 salto direttamente alla fine
```

```
slti $t1, $s5, 10 # meno di 11 ovvero 10
```

```
li $t0, 0
```

```
beq $t0, $t1, salta_stampa_utente
```

```
# UTENTE DIECI
```

```
la $s0,UtenteDieci # inirizzo memoria utente10
```

```
jal STAMPA_UTENTE
```

```
salta_stampa_utente:
```

```
la $a0,str_SottoMenuCercaProfilo # Stampa scelta torna indietro, ed aspetto che l'utente prema un tasto
```

```
li $v0,4 # print_string
```

```
syscall
```

```
li $v0,5 # read_int
```

```
syscall # risultato in $v0
```

```
j MENU_PRINCIPALE
```

```
# FINE STAMPA UTENTI PUBBLICI
```

```
# STRING COMPARE -----
```

```
# compara le 2 stringhe se uguali fino a 3 caratteri
```

```
# setta a0 a 1
```

```
# se diversi setta a0 a 2
```

```
START_COMPARE_STRING:
```

```
li $t3, 0 # conta i caratteri in comune
```

```

li $t4, 0 # conta quante volte "sbaglio"

li $t5, 3 # Dopo quante volte sbagliate / giuste esco dal ciclo

COMPARA_STRINGHE:

lbu          $t0,0($a0)          # Load a byte from the first source string.

lbu          $t1,0($a1)          # Load a byte from the second source string.

# Compare the two characters loaded.

blt          $t0,$t1,COMPARE_CARATTERI_DIVERSI

bgt          $t0,$t1,COMPARE_CARATTERI_DIVERSI

addi        $t3,$t3,1 # Aumenta "carattere uguale"

beq $t3, $t5, COMPARE_EXIT_STRINGHE_UGUALI

# Both characters are the same.

beqz        $t1,COMPARE_EXIT_STRINGHE_UGUALI # Return 0 if both are equal to NULL

COMPARE_CONTINUE: # o/w Increment both addresses and continue.

    addi    $a0,$a0,1

    addi    $a1,$a1,1

    b          COMPARA_STRINGHE

COMPARE_CARATTERI_DIVERSI: # Aumenta "carattere diverso"

    addi    $t4,$t4,1

    beq $t4, $t5, COMPARE_EXIT_STRINGHE_DIVERSE

    j COMPARE_CONTINUE

COMPARE_EXIT_STRINGHE_UGUALI:

    li $v0,1          # vo a 1 uguale

    jr $ra

COMPARE_EXIT_STRINGHE_DIVERSE:

    li $v0,2          # vo a 2 --> stringhe diverse

```



```
jr $ra
```

```
# FINE FUNZIONE STRING COMPARE -----
```

```
# MENU CERCA UTENTE -----
```

```
SCELTA_CERCA_UTENTE:
```

```
la $a0, Str_ChiediNome # chiediamo nome
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, Space_Nome # leggiamo nome
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediCognome # chiediamo il cognome
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, Space_Cognome # leggiamo il cognome
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
li $s4, 0 # Setto che nessun utente è stato trovato[0], nelle procedure se trovo una stringa lo setto ad 1
```

```
# UTENTE UNO
```

```
la $s0, UtenteUno # indirizzo memoria utentel
```

```
jal COMPARA_UTENTE
```

```

# UTENTE DUE

la $s0,UtenteDue # inirizzo memorio utente2

jal COMPARA_UTENTE

# UTENTE TRE

la $s0,UtenteTre # inirizzo memorio utente3

jal COMPARA_UTENTE

# UTENTE QUATTRO

la $s0,UtenteQuattro # inirizzo memorio utente4

jal COMPARA_UTENTE

# UTENTE CINQUE

la $s0,UtenteCinque # inirizzo memorio utente5

jal COMPARA_UTENTE

# UTENTE SEI

la $s0,UtenteSei # inirizzo memorio utente6

jal COMPARA_UTENTE

# UTENTE SETTE

la $s0,UtenteSette # inirizzo memorio utente7

jal COMPARA_UTENTE

# If $s is less than immediate, $t is set to one. It gets zero otherwise.

# Syntax: slti $t, $s, imm

# se il numero di iscritti e' meno di 7 salto direttamente alla fine

# se il numero di iscritti e' meno di 8 salto direttamente alla fine

slti $t1, $s5, 8 # meno di 8 ovvero 7

li $t0, 1

beq $t0, $t1, salta_cerca_utenti

```

```

# UTENTE OTTO

la $s0,UtenteOtto # inidirizzo memoria utente8

jal COMPARA_UTENTE

# se il numero di iscritti e' meno di 9 salto direttamente alla fine

slti $t1, $s5, 9 # meno di 9 ovvero 8

li $t0, 1

beq $t0, $t1, salta_cerca_utenti

# UTENTE NOVE

la $s0,UtenteNove # inidirizzo memoria utente9

jal COMPARA_UTENTE

# se il numero di iscritti e' meno di 10 salto direttamente alla fine

slti $t1, $s5, 10 # meno di 10 ovvero 9

li $t0, 1

beq $t0, $t1, salta_cerca_utenti

# UTENTE DIECI

la $s0,UtenteDieci # inidirizzo memoria utente10

jal COMPARA_UTENTE

salta_cerca_utenti:

#li $s4,0 # Setto che nessun utente è stato trovato[0], nelle procedure se trovo una stringa lo setto ad 1

li $t0, 1 #carico 1

beq $t0, $s4, procedura_cerca_esci # se c'è uno vuol dire che ho trovato qualche stringa

la $a0,_str_CercaUtenti_nessunRisultato # spiega scelta

li $v0,4 # print_string

syscall

```

```
procedura_cerca_esci:
```

```
la $a0, str_SottoMenuCercaProfilo # Stampa scelta torna indietro, ed aspetto che l'utente preme un tasto
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
li $v0, 5 # read_int
```

```
syscall # risultato in $v0
```

```
j MENU_PRINCIPALE
```

```
# FINE MENU CERCA UTENTE -----
```

```
# FUNZIONE CERCA
```

```
COMPARA_UTENTE:
```

```
# CERCO SE IL PROFILO è PUBBLICO
```

```
lw $t1, 16($s0) # $t1 puntatore a privato(0) o pubblico(1)
```

```
la $a0, 0($t1) # copio in a0 il "tipo di privacy"
```

```
la $a1, str_Pubblico # copio in a1 il nome inserito dall'utente
```

```
# call the strcpy function
```

```
# ritorna 1 in a0 se sono uguali
```

```
# ritorna 2 in a0 se sono diverse
```

```
addi $sp, $sp, -4
```

```
sw $ra, 0($sp)
```

```
jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $t0, 1 #carico 1
```

```

beq $t0, $v0, utente_pubblico # se è ad 1 le c'è una sottostringa e stampo l'utente

j $ra # esco se l'utente è privato ....

utente_pubblico:

# FINE SE IL PROFILO è PUBBLICO

# Cerco nome

lw $t1,0($s0) # $t1 puntatore al nome

la $a0,0($t1) # copio in a0 il "nome"

la $a1,Space_Nome # copio in a1 il nome inserito dall'utente

# call the strcpy function

# ritorna 1 in a0 se sono uguali

# ritorna 2 in a0 se sono diverse

addi $sp, $sp, -4

sw $ra, 0($sp)

jal START_COMPARE_STRING; # Procedura per confronto stringhe

lw $ra, 0($sp)

addi $sp, $sp, 4

li $t0, 1 #carico 1

beq $t0, $v0, utenteunotrovato # se è ad 1 le c'è una sottostringa e stampo l'utente

# fine cerco nome

# cerco cognome

lw $t1,4($s0) # $t1 puntatore al conome

la $a0,0($t1) # copio in a0 il "conome"

la $a1,Space_Cognome # copio in a1 il cognome inserito dall'utente

```

```

# call the strcpy function

# ritorna 1 in a0 se sono uguali

# ritorna 2 in a0 se sono diverse

addi $sp, $sp, -4

sw $ra, 0($sp)

jal START_COMPARE_STRING; # Procedura per confronto stringhe

lw $ra, 0($sp)

addi $sp, $sp, 4

li $t0, 1 #carico 1

beq $t0, $v0, utenteunotrovato # se è ad 1 le c'è una sottostringa e stampo l'utente

# Fine cerco cognome

j $ra # esco se l'utente non è stato trovato ...

utenteunotrovato:

la $a0, str_ACapo # str_ComparaStringheUguali stampo che ho trovato un risultato

li $v0, 4 # print_string

syscall

# stampo nome e cognome

la $a0, str_stampa_nome # stampo nome:

li $v0, 4

syscall

lw $t1, 0($s0) # Puntatore all'utente

la $a0, 0($t1) # Puntatore al nome

li $v0, 4 # print_string

syscall

```

```
la $a0, str_stampa_cognome # stampo cognome:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1, 4($s0) # $t1 puntatore al conome
```

```
la $a0, 0($t1) # Puntatore al cognome
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
la $a0, str_stampa_interessi# stampo interessi:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1, 8($s0) # $t1 puntatore agli interessi
```

```
la $a0, 0($t1) # Puntatore al cognome
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
la $a0, str_stampa_interessi# stampo UserID:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1, 12($s0) # $t1 puntatore agli userID
```

```
la $a0, 0($t1) # Puntatore al cognome
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
la $a0, str_stampa_email# stampo la mail:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1,20($s0)    # $t1 puntatore agli userID
```

```
la $a0,0($t1) # Puntatore al cognome
```

```
li $v0,4          # print_string
```

```
syscall
```

```
li $s4,1 # Setto che nessun utente è stato trovato[0], nelle procedure se trovo una stringa lo setto ad 1
```

```
j $ra # esco se l'utente è stato trovato
```

```
# FINE FUNZIONE CERCA
```

```
# FUNZIONE STAMPA UTENTE PUBBLICO
```

```
STAMPA_UTENTE:
```

```
# CONTROLLO SE IL PROFILO è PUBBLICO
```

```
lw $t1,16($s0)    # $t1 puntatore a privato(0) o pubblico(1)
```

```
la $a0,0($t1)     # copio in a0 il "tipo di privacy"
```

```
la $a1,str_Pubblico # copio in a1 il nome inserito dall'utente
```

```
# call the strcpy function
```

```
# ritorna 1 in a0 se sono uguali
```

```
# ritorna 2 in a0 se sono diverse
```

```
addi $sp, $sp, -4
```

```
sw $ra, 0($sp)
```

```
jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $t0, 1 #carico 1
```

```
beq $t0, $v0, stampa_utente_pubblico # se è ad 1 le c'è una sottostringa e stampo l'utente
```



```
j $ra # esco se l'utente è privato ....
```

```
stampa_utente_pubblico:
```

```
la $a0, str_ACapo # str_ComparaStringheUguali stampo che ho trovato un risultato
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
# stampo nome e cognome
```

```
la $a0, str_stampa_nome # stampo Nome:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1, 0($s0) # Puntatore all'utente
```

```
la $a0, 0($t1) # Puntatore al nome
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
la $a0, str_stampa_cognome # stampo Cognome:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1, 4($s0) # $t1 puntatore al cognome
```

```
la $a0, 0($t1) # Puntatore al cognome
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
la $a0, str_stampa_userID # stampo UserID:
```

```
li $v0, 4
```

```
syscall
```

```
lw $t1,12($s0)    # $t1 puntatore al userID
```

```
la $a0,0($t1)
```

```
li $v0,4          # stampo
```

```
syscall
```

```
la $a0,str_stampa_interessi # stampo interessi:
```

```
li $v0,4
```

```
syscall
```

```
lw $t1,8($s0)    # $t1 puntatore al interessi
```

```
la $a0,0($t1)
```

```
li $v0,4          # stampo
```

```
syscall
```

```
j $ra # esco dalla funzione
```

```
# FINE FUNZIONE STAMPA UTENTE PUBBLICO
```

```
# CREA UTENTE OTTO
```

```
crea_utente_otto:
```

```
la $a0, Str_ChiediPrivacy # chiediamo la privacy
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, visibilita8      # leggiamo la privacy
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```

la $a0,visibilita8 # copio in a0 il nome inserito dall'utente

la $a1,str_Pubblico # copio in a1 il nome inserito dall'utente

# call the strcpy function

# ritorna 1 in a0 se sono uguali

# ritorna 2 in a0 se sono diverse

addi $sp, $sp, -4

sw $ra, 0($sp)

jal START_COMPARE_STRING; # Procedura per confronto stringhe

lw $ra, 0($sp)

addi $sp, $sp, 4

# call the controlla_max_utenti_privati_pubblici function

# ritorna 1 in a0 se sono continuo con la registrazione

# ritorna 2 in a0 se devo uscire (messaggi di errori già stampati nella funzione)

# in $s4 ci sarà il valore 0 [pubblico] o privato [1]

addi $sp, $sp, -4

sw $ra, 0($sp)

jal controlla_max_utenti_privati_pubblici

lw $ra, 0($sp)

addi $sp, $sp, 4

li $t1, 2 # carico lo zero

beq $t1, $a0, MENU_PRINCIPALE #Ritorna al menu principale (c'è stato un errore)

la $a0, Str_ChiediNome # chiediamo nome

li $v0, 4

syscall

la $a0, nome8 # leggiamo nome

li $a1, 256

```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediCognome # chiediamo il cognome
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, cognome8 # leggiamo il cognome
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediInteressi # chiediamo gli interessi
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, interessi8 # leggiamo gli interessi
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediUserID # chiediamo l'UserID
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, userID8 # leggiamo l'UserID
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediEmail # chiediamo la mail
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, email8 # leggiamo la mail
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediPassword# chiediamo la password
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, password8 # leggiamo la password
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, str_SottoMenuCreaProfilo # Visualizza SottoMenu
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
li $v0, 5 # read_int
```

```
syscall # risultato in $v0
```

```
move $t0, $v0 # salva la scelta in t0
```

```
li $t1, 0 # carico lo zero
```

```
beq $t1, $v0, crea_utente_otto #Ritorna a richiedere tutti i dati
```

```
j aumenta_contatori
```

```
# FINE CREA UTENTE OTTO
```

```
# CREA UTENTE NOVE
```

```
crea_utente_nove:
```

```

la $a0, Str_ChiediPrivacy # chiediamo la privacy

li $v0, 4

syscall

la $a0, visibilita9      # leggiamo la privacy

li $a1, 256

li $v0, 8

syscall

la $a0, visibilita9 # copio in a1 il nome inserito dall'utente

la $a1, str_Pubblico # copio in a1 il nome inserito dall'utente

# call the strcpy function

# ritorna 1 in a0 se sono uguali

# ritorna 2 in a0 se sono diverse

addi $sp, $sp, -4

sw $ra, 0($sp)

jal START_COMPARE_STRING; # Procedura per confronto stringhe

lw $ra, 0($sp)

addi $sp, $sp, 4

# call the controlla_max_utenti_privati_pubblici function

# ritorna 1 in a0 se sono continuo con la registrazione

# ritorna 2 in a0 se devo uscire (messaggi di errori già stampati nella funzione)

# in $s4 ci sarà il valore 0 [pubblico] o privato [1]

addi $sp, $sp, -4

sw $ra, 0($sp)

jal controlla_max_utenti_privati_pubblici

lw $ra, 0($sp)

addi $sp, $sp, 4

li $t1, 2 # carico lo zero

```

```
beq $t1, $a0, MENU_PRINCIPALE #Ritorna al menu principale (c'è stato un errore)
```

```
la $a0, Str_ChiediNome # chiediamo nome
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, nome9 # leggiamo nome
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediCognome # chiediamo il cognome
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, cognome9 # leggiamo il cognome
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediInteressi # chiediamo gli interessi
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, interessi9 # leggiamo gli interessi
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediUserID # chiediamo l'UserID
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, userID9 # leggiamo l'UserID
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediEmail # chiediamo la mail
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, email9 # leggiamo la mail
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediPassword # chiediamo la password
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, password9 # leggiamo la password
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, str_SottoMenuCreaProfilo # Visualizza SottoMenu
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
li $v0, 5 # read_int
```

```
syscall # risultato in $v0
```

```
move $t0, $v0 # salva la scelta in t0
```



```
li $t1, 0 # carico lo zero  
beq $t1, $v0, crea_utente_nove #Ritorna a richiedere tutti i dati
```

```
j aumenta_contatori
```

```
# FINE CREA UTENTE NOVE
```

```
# CREA UTENTE DIECI
```

```
crea_utente_dieci:
```

```
la $a0, Str_ChiediPrivacy # chiediamo la privacy
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, visibilita0 # leggiamo la privacy
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, visibilita0 # copio in a0 il nome inserito dall'utente
```

```
la $a1, str_Pubblico # copio in a1 il nome inserito dall'utente
```

```
# call the strcpy function
```

```
# ritorna 1 in a0 se sono uguali
```

```
# ritorna 2 in a0 se sono diverse
```

```
addi $sp, $sp, -4
```

```
sw $ra, 0($sp)
```

```
jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```

# call the controlla_max_utenti_privati_pubblici function

# ritorna 1 in a0 se sono continuo con la registrazione

# ritorna 2 in a0 se devo uscire (messaggi di errori già stampati nella funzione)

# in $s4 ci sarà il valore 0 [pubblico] o privato [1]

addi $sp, $sp, -4

sw $ra, 0($sp)

        jal controlla_max_utenti_privati_pubblici

lw $ra, 0($sp)

addi $sp, $sp, 4

li $t1, 2 # carico lo zero

beq $t1, $a0, MENU_PRINCIPALE #Ritorna al menu principale (c'è stato un errore)

la $a0, Str_ChiediNome    # chiediamo nome

li $v0, 4

syscall

la $a0, nome10    # leggiamo nome

li $a1, 256

li $v0, 8

syscall

la $a0, Str_ChiediCognome # chiediamo il cognome

li $v0, 4

syscall

la $a0, cognome10 # leggiamo il cognome

li $a1, 256

li $v0, 8

syscall

la $a0, Str_ChiediInteressi    # chiediamo gli interessi

```

```
li $v0, 4
```

```
syscall
```

```
la $a0, interessi10      # leggiamo gli interessi
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediUserID # chiediamo l'UserID
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, userID10 # leggiamo l'UserID
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediEmail  # chiediamo la mail
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, email10 # leggiamo la mail
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, Str_ChiediPassword # chiediamo la password
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, password10      # leggiamo la password
```

```
li $a1, 256
```

```
li $v0, 8
```

```
syscall
```

```
la $a0, str_SottoMenuCreaProfilo # Visualizza SottoMenu
```

```
li $v0, 4 # print_string
```

```
syscall
```

```
li $v0, 5 # read_int
```

```
syscall # risultato in $v0
```

```
move $t0, $v0 # salva la scelta in t0
```

```
li $t1, 0 # carico lo zero
```

```
beq $t1, $v0, crea_utente_dieci #Ritorna a richiedere tutti i dati
```

```
j aumenta_contatori
```

```
# FINE CREA UTENTE DIECI
```

```
# CONTROLLA SE HO RAGGIUNTO IL NUMERO MASSIMO DI UTENTI PRIVATI / PUBBLICI
```

```
# controlla_max_utenti_privati_pubblici function
```

```
# ritorna 1 in a0 se sono continuo con la registrazione
```

```
# ritorna 2 in a0 se devo uscire (messaggi di errori già stampati nella funzione)
```

```
controlla_max_utenti_privati_pubblici:
```

```
li $s4, 0 # Setto a zero S4, 'lo resetto'
```

```
li $t0, 1
```

```
beq $t0, $v0, check_utente_pubblico # vado all'utente privato o pubblico
```

```
li $s4, 1 # Mi setto s4 ad uno così so che contatore incrementare una volta che l'utente è stato inserito
```

```
j check_utente_privato
```

```
check_utente_pubblico:
```

```
li $t1, 5 # numero immediato massimo utenti pubblici
beq $t1, $s6, Max_utenti_pubblici # Controllo di non aver superato il max numero di iscritti
j continua_registrazione
```

```
Max_utenti_pubblici:
```

```
la $a0, str_Errore_Max_public_user # Mostro il messaggio di errore
li $v0, 4
syscall
li $a0, 2 # Esci con errore, setto quindi a0 a 2
j $ra
```

```
check_utente_privato:
```

```
li $t1, 5 # numero immediato massimo utenti privati
beq $t1, $s7, Max_utenti_privati # Controllo di non aver superato il max numero di iscritti
j continua_registrazione
```

```
Max_utenti_privati:
```

```
la $a0, str_Errore_Max_private_user # Mostro il messaggio di errore
li $v0, 4
syscall
li $a0, 2 # Esci con errore, setto quindi a0 a 2
j $ra
```

```
continua_registrazione:
```

```
j $ra # Ritorno alla fase di registrazione se non ho raggiunto i massimi ....
```

```
# FINE CONTROLLA SE HO RAGGIUNTO IL NUMERO MASSIMO DI UTENTI PRIVATI / PUBBLICI
```

```
# AUMENTA CONTATORI
```

```
# Promemoria
```

```
#li $s5,7 # mi segno quanti uteni già salvati ci sono
```

#li \$s6,4 # mi segno quanti utenti pubblici ci sono

#li \$s7,3 # mi segno quanti utenti privati ci sono

aumenta\_contatori:

li \$t0, 1

beq \$t0, \$s4, Aumenta\_Contatore\_UtentePrivato # vado all'utente privato o pubblico

addi \$s6,\$s6,1 # Aumenta indice contatore utenti pubblici

j Aumenta\_Contatore\_Utenti

Aumenta\_Contatore\_UtentePrivato:

addi \$s7,\$s7,1 # Aumenta indice contatore utenti privati

Aumenta\_Contatore\_Utenti:

addi \$s5,\$s5,1 # Aumenta indice contatore utenti

j MENU\_PRINCIPALE

# FINE AUMENTA CONTATORI

# CANCELLA UTENTI

SCELTA\_CANCELLA\_UTENTE:

li \$t0, 7

beq \$t0, \$s5, cancella\_utente\_accesso\_illegale # se provo a cancellare un utente che non ha inserito l'utente

li \$t0, 8

beq \$t0, \$s5, cancella\_utente\_8 #cancella utent8

```
li $t0, 9
beq $t0, $s5, cancella_utente_9 #cancella utentem8
```

```
li $t0, 10
beq $t0, $s5, cancella_utente_10 #cancella utentem8
```

```
cancella_utente_avvenuto:
```

```
la $a0, str_SottoMenuCancellaProfilo_cancellato # stampo utente cancellato
li $v0, 4 # print_string
syscall
j continua_cancella_utente
```

```
cancella_utente_accesso_illegale:
```

```
la $a0, str_SottoMenuCancellaProfilo_accesso_ristretto # Stampa scelta torna indietro, ed aspetto che l'utente
prema un tasto
li $v0, 4 # print_string
syscall
```

```
continua_cancella_utente:
```

```
la $a0, str_SottoMenuCercaProfilo # Stampa scelta torna indietro, ed aspetto che l'utente prema un tasto
li $v0, 4 # print_string
syscall
```

```
li $v0, 5 # read_int
```

```
syscall # risultato in $v0
```

```
j MENU_PRINCIPALE
```

```
# FINE CANCELLA UTENTE
```

```
# CANCELLA UTENTE OTTO
```

```
cancella_utente_8:
```

```
la $a0,visibilita8 # copio in a0 il nome inserito dall'utente
```

```
la $a1,str_Pubblico # copio in a1 il nome inserito dall'utente
```

```
# call the strcpy function
```

```
# ritorna 1 in v0 se sono uguali
```

```
# ritorna 2 in v0 se sono diverse
```

```
addi $sp, $sp, -4
```

```
sw $ra, 0($sp)
```

```
jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $t0, 1
```

```
beq $t0, $v0, cancella_pubblico8 # vado all'utente privato o pubblico
```

```
addi $s7,$s7,-1 # Aumenta indice contatore utenti privati
```

```
j cancella_utente8
```

```
cancella_pubblico8:
```

```
addi $s6,$s6,-1 # Aumenta indice contatore utenti pubblici
```

```
cancella_utente8:
```

```
addi $s5,$s5,-1 # Aumenta indice contatore utenti
```

```
j cancella_utente_avvenuto
```



```
# FINE CANCELLA OTTO
```

```
# CANCELLA UTENTE NOVE
```

```
cancella_utente_9:
```

```
    la $a0,visibilita9 # copio in al il nome inserito dall'utente
```

```
    la $a1,str_Pubblico # copio in al il nome inserito dall'utente
```

```
    # call the strcpy function
```

```
    # ritorna 1 in a0 se sono uguali
```

```
    # ritorna 2 in a0 se sono diverse
```

```
    addi $sp, $sp, -4
```

```
    sw $ra, 0($sp)
```

```
        jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $t0, 1
```

```
beq $t0, $v0, cancella_pubblico9 # vado all'utente privato o pubblico
```

```
addi    $s7,$s7,-1 # Aumenta indice contatore utenti privati
```

```
j cancella_utente9
```

```
cancella_pubblico9:
```

```
addi    $s6,$s6,-1 # Aumenta indice contatore utenti pubblici
```

```
cancella_utente9:
```

```
addi    $s5,$s5,-1 # Aumenta indice contatore utenti
```

```
j cancella_utente_avvenuto
```

```
# FINE CANCELLA NOVE
```

```
# CANCELLA UTENTE DIECI
```

```
cancella_utente_10:
```

```
    la $a0,visibilita10 # copio in a0 il nome inserito dall'utente
```

```
    la $a1,str_Pubblico # copio in a1 il nome inserito dall'utente
```

```
    # call the strcpy function
```

```
    # ritorna 1 in a0 se sono uguali
```

```
    # ritorna 2 in a0 se sono diverse
```

```
    addi $sp, $sp, -4
```

```
    sw $ra, 0($sp)
```

```
        jal START_COMPARE_STRING; # Procedura per confronto stringhe
```

```
lw $ra, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $t0, 1
```

```
beq $t0, $v0, cancella_pubblico10 # vado all'utente privato o pubblico
```

```
addi    $s7,$s7,-1 # Aumenta indice contatore utenti privati
```

```
j cancella_utente10
```

```
cancella_pubblico10:
```

```
addi    $s6,$s6,-1 # Aumenta indice contatore utenti pubblici
```

```
cancella_utente10:
```

```
addi    $s5,$s5,-1 # Aumenta indice contatore utenti
```

```
j cancella_utente_avvenuto
```

```
# FINE CANCELLA DIECI
```

fine:

```
la $a0,ciao      # saluti
```

```
li $v0,4 # print_string
```

```
syscall
```

```
li $v0,10      # exit
```

```
syscall
```