Progetto di programmazione   
 Social Network

A cura di Alessandro Bonomo, Francesco Bruno

# Approccio alla soluzione

Analizzando il problema del social network abbiamo individuato 3 strutture dati fondamentali che descrivono il sistema:

1. La **struct post**, composta dal testo (array di caratteri) che descrive il contenuto di un post.
2. La **struct listaUtenti** (lista di puntatori a utenti), composta da un utente e il puntatore al prossimo nodo utente.
3. La **struct utente**, composta da username, password (entrambi array di caratteri), utentiSeguiti (puntatore di tipo listaUtenti alla testa della lista degli utenti seguiti), contaPost (int) che conserva la conta dei post pubblicati dall’utente e un array di posts.

Una lista globale (listaUtenti \*utentiSocialNet) per memorizzare tutti gli utenti del social network.  
Un puntatore all’utente con cui si è attualmente loggati (utente\* utenteLoggato) che è NULL se non si è loggati.

## Motivazione della scelta delle strutture dati

Abbiamo scelto di creare una struct post per rendere il programma estendibile, per esempio, si potrebbe pensare di aggiungere un attributo alla struct post che contenga l’elenco degli utenti che hanno visto quel post. Dato che il social network non ha un numero fisso di utenti, abbiamo scelto di usare una lista di puntatori struct listaUtenti al posto di un array di utenti. Nella struct utente abbiamo deciso di inserire la lista degli utenti seguiti dall’utente in modo da avere una associazione 1-N tra utenti. La variabile contaPost è necessaria perché funge da dimensione logica dell’array posts. Abbiamo scelto di utilizzare un array per ricordare i post di un utente perché nella traccia è specificata una dimensione massima di post e non variabile (10).

# Struttura del programma

Leggendo le indicazioni della traccia abbiamo deciso di strutturare il programma nel modo seguente:

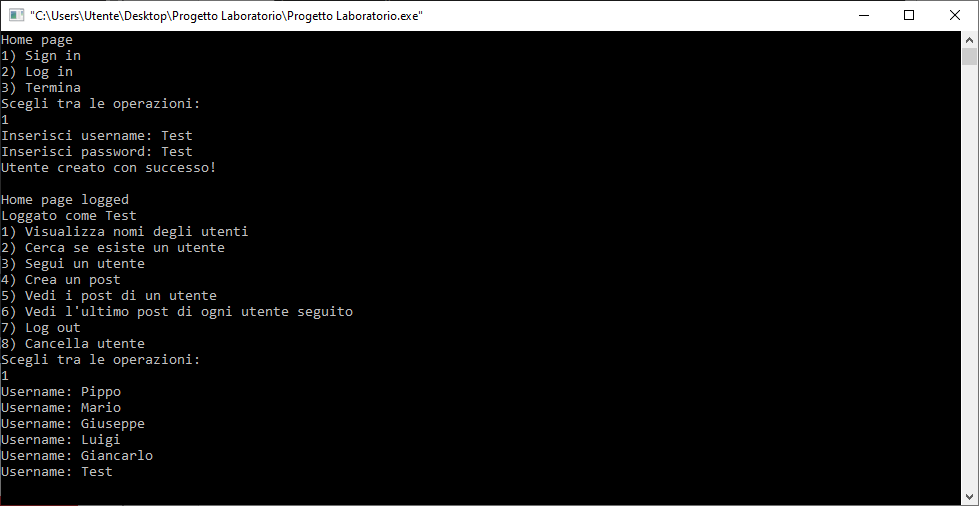
1. Una pagina iniziale (Home page) in cui ci sono due opzioni: Log In e Sign In. Sign In permette di creare un account, mentre Log In permette di accedere ad un account esistente.
2. Una pagina di benvenuto (Home page logged), in cui l’utente, appena effettuato l’accesso, ha la possibilità, a seconda della scelta richiesta, di:

* Vedere tutti gli utenti presenti nel programma (creati col sign-in o gia presettati);
* Cercare un utente scrivendo il suo nome utente (facendo attenzione anche alle minuscole e maiuscole);
* Seguire un utente del network.
* Creare un post, di un massimo di caratteri impostato nel programma, e, nel caso venga raggiunto il massimo numero di post, esso eseguirà lo shift dei post e sovrascriverà il post meno recente;
* Vedere tutti i post di un utente, che esso sia l’utente stesso che si sta usando, od un altro presente nel programma;
* Vedere l’ultimo post di ogni utente seguito presente nel programma;
* Fare il log-out dall’utente con cui stavamo interagendo, ritornando alla home page.
* Cancellare l’utente che stiamo usando, eliminandolo definitivamente dal programma insieme ai suoi dati ed ai post che erano contenuti in esso.

## Le istanze predefinite

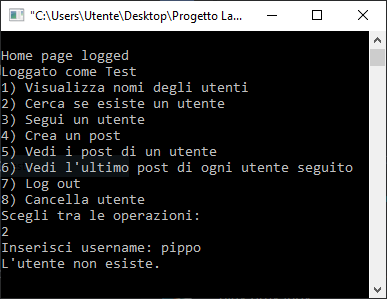
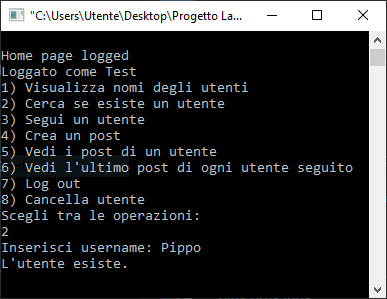
Nella traccia era indicato di caricare delle istanze predefinite di post (10 post predefiniti per utente) e utenti (5 utenti predefiniti). Abbiamo creato una matrice di post postsPredefiniti[dimUtentiPredefiniti][dimMaxPostsPerUtente] che contiene ‘dimMaxPostsPerUtente’ post per ‘dimUtentiPredefiniti’ e un array di utenti predefiniti utentiPredefiniti[dimUtentiPredefiniti]. La funzione caricaUtentiPredefiniti() provvede ad associare i post nella matrice ai rispettivi utenti predefiniti e, successivamente, inserire gli utenti predefiniti nella lista utentiSocialNet.

# Esempio di esecuzione

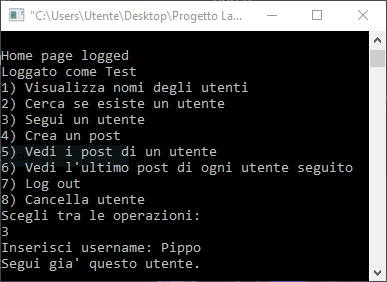
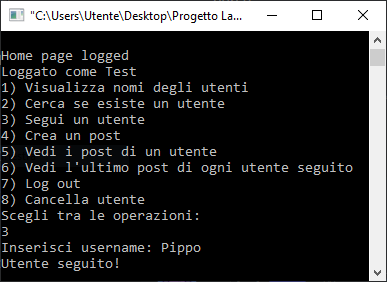


(Vedere freccia verde) All’avvio del programma ci si trova nella Home Page, le opzioni sono Sign-in (1), che registrerà un nuovo utente e lo aggiungerà alla lista “ListaUtenti”, ed inizializzerà la propria lista “Utente” (in questo caso “Test”), contenente username e password, e le susseguenti info dei post e degli utenti seguiti, Log-in (2), che permetterà di entrare in un utente, purchè esso sia presente già nel programma e che corrispondano username e password inseriti (tramite un confronto tra quelli inseriti da tastiera e quelli presenti nella lista).

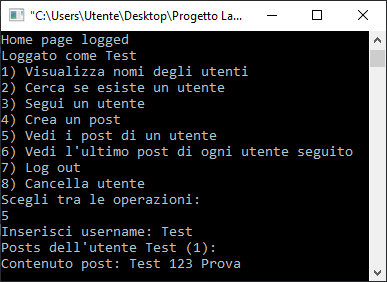
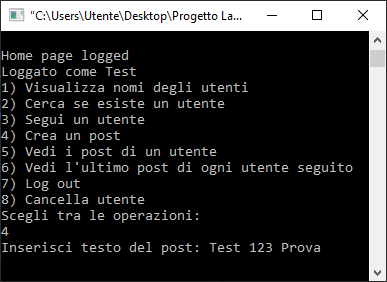
(Vedere freccia rossa) Dopo il log in, verrà chiesta l’operazione da eseguire, e tramite uno scanf ed uno switch, a seconda del valore inserito, verrà chiamata una diversa funzione. In questo caso, scegliendo l’opzione 1, verrà chiamata la funzione “StampaListaUtenti”, che stamperà tutti gli utenti presenti nella lista “ListaUtenti” con il puntatore “UtentiSocialNet”.



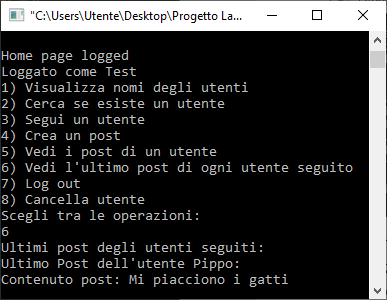
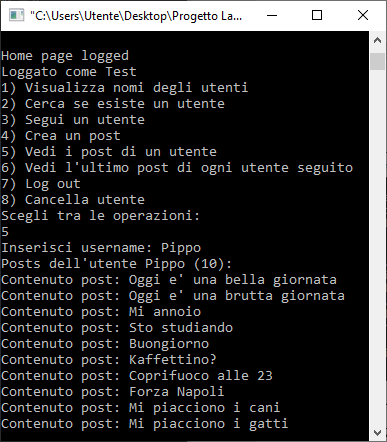
Selezionando l’opzione 2, si avvierà la funzione “OperazioneTrovaUtente”, che confronterà l’username inserito con quello delle liste tramite “GetPointerUtenteByUsername”.

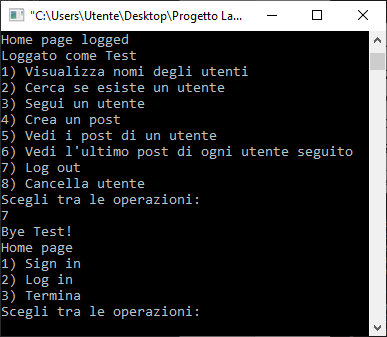


Selezionando l’opzione 3, si segue un utente, usando la funzione “OperazioneSeguiUtente”, aggiungendolo negli “UtentiSeguiti” dell’utente che sta facendo l’operazione, e se il “GetPointerUtenteByUsername” ritorna che nella lista “UtentiSeguiti” esiste già, restituisce il messaggio che già si segue quell’utente.



Selezionando l’opzione 4, si scrive un post, tramite la funzione “OperazioneCreaUnPost”, il post avrà massima dimensione “dimMaxPostsPerUtente”, e verrà poi inserito nella lista “Posts” il contatore “ContaPost” aumenta per ogni post inserito e se si raggiunge il limite massimo di post “dimMaxPosts” allora i post dell’utente verranno shiftati a sinistra tramite la funzione “shiftSxPostUtente” e si perderà il post meno recente (affianco all’username, nell’esempio “Test (1)”).



Selezionando l’opzione 5 e 6, si visualizzeranno a schermo tutti, o l’ultimo, post di un utente selezionato. Nel primo stamperà tutti i post che si trovano nella lista “Post” tramite la funzione “OperazioneVediPostDiUnUtente”; nel secondo stamperà unicamente l’ultimo post di ogni utente che si trova nella lista “UtentiSeguiti”, tramite la funzione “StampaUltimoPostDiOgniUtenteSeguito”.

Selezionando l’opzione 7, l’utente uscirà dalla sua pagina, ritornando alla Home Page iniziale, questo perché la variabile “esito” viene portata a “True” e ciò comporta un richiamo della funzione “HomePage”. Sia “HomePage” che “HomePageLogged” sono variabili booleane, in modo tale che si possano decidere le operazioni da eseguire attraverso un semplice confronto nello switch.

Selezionando l’opzione 8, si effettuerà l’eliminazione completa dell’utente loggato, con conseguente perdita dei suoi dati (username, password, post, utente seguiti), e quindi della sua lista dedicata, ed anche la sua presenza nella lista “ListaUtenti” (nell’esempio infatti, dopo aver effettuato l’operazione, entrando dall’account Pippo, e visualizzando a schermo tutti i nomi presenti nel programma, l’utente da noi creato Test, non è piu presente, e con lui anche tutto ciò correlato ad esso). Prima dell’eliminazione dell’account verrà richiesto di inserire la password per conferma. Tutto ciò viene effettuato tramite la funzione “OperazioneCancellaUtente”.

