

Documento descrizione progetto

1. Specifica dei requisiti

Il programma realizzato ha l'obiettivo di ricercare, all'interno di una matrice quadrata, tutte le parole contenute in essa, attraverso un metodo di ricerca molto simile al gioco per dispositivi mobile ruzzle.

Il programma svolge in completa autonomia la ricerca delle parole di senso compiuto all'interno della matrice che il medesimo andrà a creare partendo dalla stringa fornita in ingresso da parte dell'utente. Le parole lecite che possono essere trovate all'interno della matrice sono contenute in un dizionario fornito al programma. Ad ogni parola viene assegnato un punteggio.

1.1 Input del programma

IL programma per funzionare correttamente deve avere in ingresso 4 parametri che l'utente fornirà tramite console.

I quattro parametri in ingresso sono:

1) Una stringa di caratteri ASCII standard (sia maiuscoli che minuscoli) di lunghezza pari a N^2 dove N (che il programma estrapolerà in automatico da N^2) corrisponderà alla dimensione delle righe e delle colonne della matrice. La matrice verrà riempita dal programma in modo sequenziale facendo anche in modo che la differenza tra caratteri maiuscoli e minuscoli sia indifferente alla risoluzione di esso.

2) Il nome del dizionario dove sono salvate le parole lecite che possono essere trovate all'interno della matrice (Il nome del dizionario di questo programma è list.txt)

3) Il nome del file esterno in cui verranno salvati i risultati del programma (l'utente può chiamare questo file con qualsiasi voglia nome, l'importante è che il file abbia come estensione .txt ES: nomefile.txt)

4) Un numero naturale M che indichi il numero massimo di parole da far stampare, in ordine di punteggio decrescente ed alfabetico, sul file esterno dove vengono salvati i risultati dell'esecuzione del programma.

1.2 Regole di identificazione parole

Il programma una volta ricevuti i parametri in ingresso e formata la matrice, comincerà la ricerca delle parole valide controllando per ogni singola cella della matrice ed in modo sequenziale nelle seguenti direzioni

1. Orizzontalmente da sinistra a destra
2. Orizzontalmente da destra a sinistra
3. Verticalmente dall'alto in basso
4. Verticalmente dal basso in alto
5. Diagonalmente dall'alto in basso da sinistra a destra
6. Diagonalmente dall'alto in basso da destra a sinistra
7. Diagonalmente dall'alto in basso da sinistra a destra
8. Diagonalmente dal basso in alto da destra a sinistra

1.3 Regole calcolo punteggio

Una volta trovata la parola il punteggio della medesima sarà così calcolato:

$\text{Punteggio} = (\text{lunghezza parola}) * (\text{somma dei singoli punti per lettera})$

I punti assegnati alle singole lettere sono:

- 1) 3 punti per le lettere W, X, Y
- 2) 2 punti per le lettere H, Q, U, Z

3) 1 punto per tutte le restanti lettere dell'alfabeto.

1.4 Output del programma

Nel file di output verranno stampate le M parole trovate in ordine decrescente di punteggio ed in caso di parità di punteggio in ordine alfabetico.

La formattazione delle parole trovate sarà così formata:

N-Parola-Punteggio-Percorso

N=posizione in classifica in base a punteggio decrescente e alfabetico della parola

Parola=la parola trovata nella matrice

Punteggio=punteggio correlato alla parola trovata

Percorso= tutte le posizioni in cui si trovano le singole lettere che compongono la parola all'interno della matrice scritte come coordinate (x,y) x=riga , y=colonna .

2 Progettazione

2.1 Le librerie

Cominciamo l'analisi del programma dalle librerie incluse:

Iostream = libreria standard per la gestione degli input e degli output in c++.

Iomanip = libreria standard usata per la manipolazione dei parametri , nel caso del programma viene utilizzata principalmente per il comando setw.

Cstring=libreria standard per l'utilizzo delle stringhe all'interno del c++ usata all'interno del programma in quanto si fa un grande uso della variabile di tipo stringa .

Cstdlib= libreria standar utilizzata all'interno del programma per poter chiamare la funzione atoi per convertire una const char* in un intero

Fstream=libreria standard che viene utilizzata nel programma per la gestione dei file come il dizionario list.txt.

Vector=libreria del c++11 utilizzata nel programma per la gestione dei vettori

Cmath= libreria standard che viene implementata all'interno del programma per poter usufruire delle funzioni matematiche come il calcolo di una radice quadrata.

2.2 Le variabili globali

Subito dopo le librerie troviamo tra le variabili globali:

word_list: Struttura utilizzata per creare una lista contenente le parole trovate all'interno della matrice e che poi verranno stampate nel file di output.

Coordinate:struttura che serve per creare una lista nella quale vengono salvate le posizioni occupate dalla parola all'interno della matrice.

2.3 Il main

Passiamo ora all'analisi del main.

Attraverso la console vengono passati al main i parametri in questa forma:

```
nomeprogramma.exe "Stringainserita" "list.txt" "risultato.txt"  
numeromaxparole
```

Dopo che l'utente avrà inserito i parametri nel main si andrà a creare la matrice dinamica che conterrà la stringa definita dall'utente.

Nel programma viene utilizzata una matrice dinamica in modo tale che l'utente possa decidere deliberatamente la dimensione della medesima.

Subito dopo aver creato la matrice il main invocherà le funzioni dizionario e ordinamento.

2.4 La funzione dizionario

All'interno della funzione l'algoritmo legge una per volta le parole contenute all'interno del file "list.txt" (che rappresenta il dizionario) e per ogni parola contenuta in "list.txt" si invoca la funzione "cerca".

2.5 La funzione cerca

Una volta che la funzione dizionario ha passato la parola da ricercare, la funzione cerca tramite l'algoritmo, basato su funzioni ricorsive, cerca la parola passata all'interno della nostra matrice.

Processo di ricerca:

1) Ricerca, attraverso scansione con un ciclo for di tutta la matrice, della prima lettera formante la parola passata.

2) Trovata la prima lettera si richiama la medesima funzione passandogli le coordinate della cella sulla quale si trova.

3) Una volta richiamata, inizia il controllo in tutte le otto direzioni. Ogni volta che in una delle otto direzioni si trova che la lettera corrisponde alla parola cercata, si salvano le coordinate della cella nella lista "coordinate" e si richiama la medesima funzione passando sempre le coordinate dell'ultima cella in cui è stata trovata la lettera.

Questo procedimento continua fino a quando il programma o trova la parola desiderata oppure se non trova la parola si interrompe ritorna al chiamante (ossia la funzione ricorsiva medesima) e ripristina la condizione precedente alla chiamata alla funzione ricorsiva.

4) Se la funzione cerca ha trovato la parola chiama la funzione "creatlist"

5) La nostra funzione ritorna al dizionario dal quale continuerà a ricevere parole fino a quando non saranno esaurite quelle contenute in "list.txt"

2.6 Funzione creatlist

La funzione chiamata dalla funzione cerca crea la "word-list". Ogni nodo della word_list contiene le parole trovate nella matrice con il rispettivo punteggio e coordinate.

2.7 Funzione punteggio parola

La funzione punteggio parola calcola e restituisce il punteggio per ogni singola parola trovata all'interno della matrice. Il valore di ritorno della funzione viene passato alla funzione "creatlist".

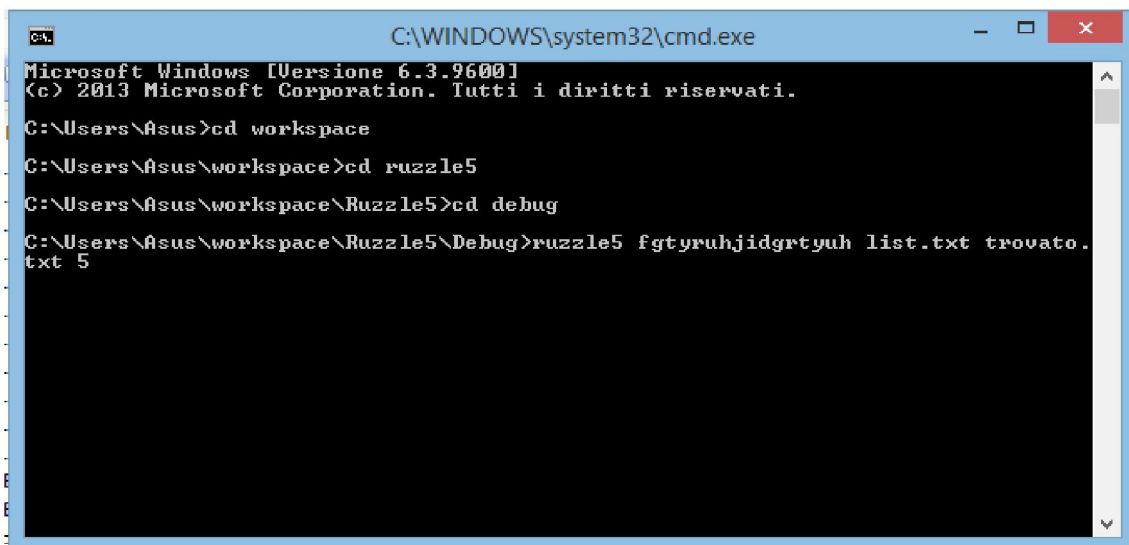
2.8 Funzione ordinamento stampa

Questa funzione viene chiamata nel main dopo che la funzione "dizionario" ha finito il suo compito. Alla funzione vengono passati come parametri di ingresso "word_list", la matrice dinamica, la dimensione della matrice, il file di output ed il numero massimo di parole da cercare.

A questo punto attraverso lo scorrimento della lista vengono trovate le parole con il punteggio più alto e vengono stampate in ordine decrescente di punteggio o alfabetico in caso di punteggio uguale.

3 Verifica codice

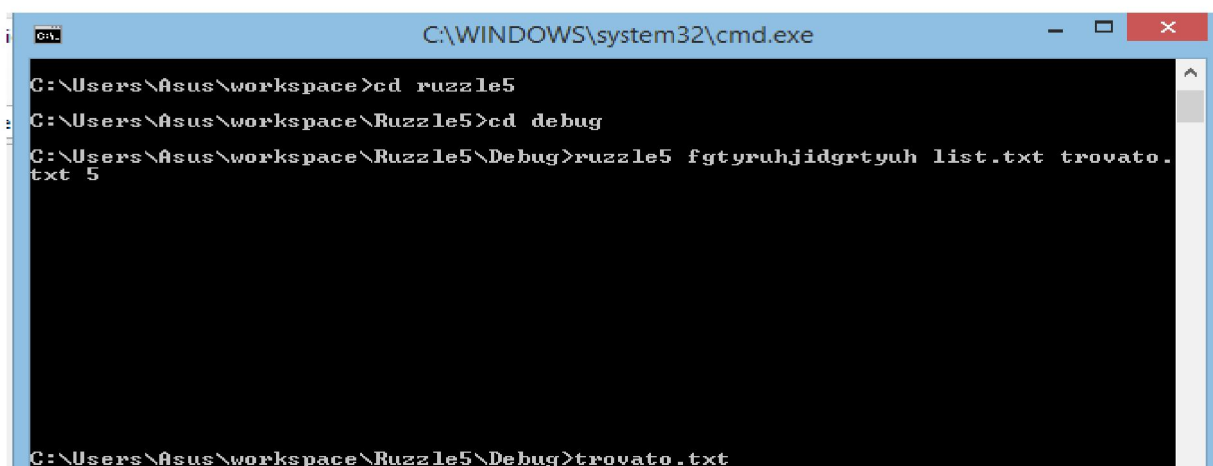
In questa sezione vengono inserite le videate che dimostrano il funzionamento de codice



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versione 6.3.9600]
(c) 2013 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\Asus>cd workspace
C:\Users\Asus\workspace>cd ruzzle5
C:\Users\Asus\workspace\Ruzzle5>cd debug
C:\Users\Asus\workspace\Ruzzle5\Debug>ruzzle5 fgtyruhjidgrtyuh list.txt trovato.
txt 5
```

Viene invocata la funzione da terminale



```
C:\WINDOWS\system32\cmd.exe

C:\Users\Asus\workspace>cd ruzzle5
C:\Users\Asus\workspace\Ruzzle5>cd debug
C:\Users\Asus\workspace\Ruzzle5\Debug>ruzzle5 fgtyruhjidgrtyuh list.txt trovato.
txt 5

C:\Users\Asus\workspace\Ruzzle5\Debug>trovato.txt
```

Dopo aver fatto eseguire il programma , chiamo tramite comando da terminale il file esterno con il risultato.

MATRICE

```
f   g   t   y
r   u   h   j
i   d   g   r
t   y   u   h
```

N°	Parola	Punteggio	Coordinate
1	thug	- 24	- (0,2);(1,2);(1,1);(0,1);
2	thug	- 24	- (0,2);(1,2);(1,1);(2,2);
3	duri	- 20	- (2,1);(1,1);(1,0);(2,0);
4	rudl	- 20	- (1,0);(1,1);(2,1);(2,0);
5	rudl	- 20	- (2,3);(3,2);(2,1);(2,0);

Risultati file esterno.

I programmatori
Michele Carnovale
Manan Tariq