Relazione progetto Assembly RISC-V
-A.A. 2021/2022-
Messaggi in Codice

Matricola 7049722
Pistolesi Boni Gabriele
gabriele.pistolesi1@stud.unifi.it

26/06/2022

Indice

1 Descrizione del funzionamento generale

Il progetto implementa cinque diversi algoritmi di cifratura che possono essere richiamati in ordine arbitrario tramite la stringa *mycypher*. Questi sono:

- Cifrario_a_Sostituzione : Le lettere della stringa di partenza vengono cifrate tramite l'utilizzo di una chiave *sostK*
- Cifrario_a_Blocchi: I caratteri della stringa *blocKey* vengono usati per cifrare quelli della stringa di partenza.
- Cifratura_Occorrenze: La stringa cifrata è ottenuta scrivendo per ogni carattere la posizione delle sue occorrenze nella stringa di partenza.
- Dizionario: Le lettere minuscole, maiuscole ed i numeri vengono cifrati secondo le regole specificate dal progetto.
- Inversione: Specchia la stringa di partenza per ottenere quella cifrata.

Il progetto inizializza le seguenti variabili globali:

- s0 puntatore alla stringa *myplaintext*.
- s1 puntatore alla stringa *mycypher*.
- s2 chiave di conversione *sostK*.
- s3 puntatore alla stringa *blocKey*.
- s4 puntatore ad una stringa ausiliaria.
- s5 variabile che comunica se il programma sta cifrando o decifrando.

2 Funzioni Principali

2.1 Lettura_mycypher

La funzione Lettura_mycypher è responsabile della lettura della stringa *mycypher* e le conseguenti chiamate alle funzioni di cifratura e decifratura.
Per ogni carattere della stringa si controlla che corrisponda ad una lettera chiamante: 'A','B','C','D' o 'E', se il confronto ha esito positivo viene effettuata la cifratura della stringa *myplaintext* con la funzione opportuna, in ordine:
Cifrario_a_Sostituzione, Cifrario_a_Blocchi, Cifratura_Occorrenze, Dizionario o Inversione.
Una volta esaminata tutta la stringa *mycypher* viene invertito il valore di *sostK* e assengato 1 ad s5 segnalando che il programma è in fase di decifratura, quindi si ripercorre la stringa partendo dalla fine verso l'inizio chiamando opportunamente le funzioni di decifratura.

2.2 Cifrario a Sostituzione

Il Cifrario a Sostituzione scorre la stringa *myplaintext* con un ciclo a_loop, dove viene identificato l'intervallo di appartenenza del carattere corrente. È la funzione ausiliara "interval_check", richiamata con una jal, che comunica col valore di ritorno a3 se il carattere in esame nel registro a2 è una lettera minuscola, maiuscola o un carattere speciale/numero (0,1,2).
I caratteri speciali e i numeri vengono ignorati, mentre le lettere vengono cifrate sommandovi la costante *sostK*. È necessario assicurarsi che non vengano superati i limiti dell'intervallo: [97,122] per le lettere minuscole e [65,90] per le maiuscole. Questo obbiettivo si raggiunge sottraendo al carattere in esame l'estremo inferiore dell'intervallo di appartenenza e facendo il mod(26) del risultato utilizzando l'istruzione "rem", sommando nuovamente l'estremo inferiore per rientrare nell'intervallo.
In ultimo è necessario specificare che l'istruzione rem t1 t1 t2 rimodula il valore passato nell'intervallo [-(t2) + 1,t2 - 1], quindi potrebbe restituire un valore negativo, in questo caso è necessario aggiungere nuovamente t2 per rientrare nell'intervallo correto [1,t2 - 1].

2.3 Cifrario a Blocchi

La funzione per la cifratura a blocchi scorre in contemporanea la stringa *myplaintext* e quella *blocKey*. Ogni carattere della stringa iniziale viene cifrato sommandovi quello corrente della stringa di cifratura *blocKey*, facendo il mod(96) del valore ottenuto. Si conclude la cifratura del carattere sommandovi 32.
È opportuno specificare che una volta finito di scorrere la stringa *blocKey*, è necessario resettare il puntatore alla testa della stringa per continuare la cifratura di *myplaintext*.

2.4 Cifratura Occorenze

La cifratura a occorrenze compone delle sotto-stringe per ogni carattere della stringa iniziale una volta sola, ovvero se sono presenti due caratteri uguali, la sottostringa relativa a quel carattere sarà una sola. Per facilitare questa operazione invece che sovrascrivere la stringa iniziale si utilizza una stringa ausiliaria con testa in s4 dove comporre il codice cifrato.
Viene usato quindi un puntatore per avanzare il ciclo c_loop che scorre la stringa di partenza per scrivere nella stringa ausiliaria il carattere di cui comporre la sottostringa. Un secondo puntatore avanza il ciclo annidato "c inner loop" necessario per trovare le occorrenze del carattere in esame, una volta trovata l'occorrenza la si sovrascrive con un valore fittizio (27) così da poter ignorare la posizione col primo puntatore. Successivamente viene chiamata di nuovo "write auxstr" per aggiungere la posizione dell'occorrenza alla sottostringa in composizione. La sottostringa è completa quando il secondo puntatore finisce di scorrere la stringa iniziale, a questo punto si scrive il carattere di spazio (ASCII 32) nella posizione successiva alla sottostringa.
Nel momento in cui il primo puntatore scorre tutta la stringa di partenza e l'ultima sottostringa è stata completata si conclude saltando alla funzione ausiliaria "swap head" che inserisce il carattere di fine stringa e scambia il puntatore s0 alla stringa iniziale con quello s4 della stringa ausiliaria, in questo modo si può continuare ad operare su s0 con eventuali altre funzioni di cifratura.

2.5 Dizionario

Il dizionario sfrutta un solo ciclo "d loop" che determina il tipo di carattere, caricato in a2, attualmente in esame dalla stringa di partenza. Per farlo viene usata la funzione interval_check, richiamandola con una jal, in questo modo il registro a3 comunica l'intervallo di appartenza del carattere contenuto in a2.
L'algoritmo implementato dal dizionario richiede anche la cifratura dei numeri, quindi è necessario un ulteriore controllo utile proprio a questo scopo.
Una volta accertato l'intervallo di appartenza di a2 si può procedere con la cifratura del carattere:
Per le lettere minuscole è necessario invertire il carattere nel suo intervallo di appartenenza, questo lo si esegue sottraendo ad a2 97 (ASCII per a), ottenendo così la sua posizione relativa alla lettera minima, successivamente a2 viene sottratto a 122 (ASCII per z) in modo da effettivamente invertire la posizione nell'intervallo [97,122]. A questo punto basterà sottrare 32 ad a2 per convertire il carattere in una lettera maiuscola. Si noti che il metodo per cifrare le lettere maiuscole è quasi identico, è sufficiente cambiare le costanti 97 e 122 con 65 e 90 e somare 32.
Infine, se il carattere è un numero, serve sottrarre 57 (ASCII per 9) ad a2 e sommarvi 48 (ASCII per 0).

2.6 Inversione

L'inversione utilizza un ciclo fill_stack che carica nella pila tutti i caratteri della stringa di partenza in ordine per poi resettare il puntatore usato per avanzare il ciclo alla testa s0 della stringa. A questo punto si passa al secondo ciclo empty_stack che estrae gli elementi dalla pila, che saranno quindi in ordine inverso, e li scrive uno ad uno nella stringa con testa s0.

3 Decifratura

3.1 Decifratura a Blocchi

La decifratura a blocchi consiste nell'eseguire l'operazione inversa di quella utilizzata per cifrare la stringa. Nel ciclo db_loop si sottrae la chiave al carattere corrente, si sottrae ulteriormente 32 e si correggono eventuali valore esterni all'intervallo [32,127] sommando 96 fino a che non si ottiene un valore maggiore o uguale a 32.

3.2 Decifratura Occorrenze

Per decifrare una stringa cifrata ad occorrenze servono due cicli annidati, il primo carica il carattere da decifrare, mentre il secondo a scrivere nella stringa ausiliaria con testa in s4, tutte le occorenze del carattere in esame, individuando le posizioni esplicitate nella sottostringa corrente. Una volta trovata la sezione che indica la posizione è necessario convertire il carattere ASCII in un numero intero così da poter facilmente scrivere il carattere nella giusta posizione.
Per farlo viene chiamata la funzione ausiliaria convert_to_int che appunto converte i caratteri che compongono la posizione in numeri interi. Una volta effettuata la conversione si può quindi scrivere il carattere in pos-1, poiché la numerazione delle posizioni parte da 1 e non da 0. Infine si scambiano i puntatori s0 ed s4 come per la cifratura chiamando swap_head.

3.3 Decifrature semplici

Le funzioni di cifratura Dizionario e Inversione non richiedono nessuna funzione di decifratura poiché basta richiamarle sulla stringa cifrata per ottenere la stringa originale. Anche per il Cifrario a Sostituzione non serve una funzione specifica di decifratura, basta invertire nella Lettura_*mycypher* la chiave *sostK* e si può richiamare la funzione stessa.

4 Funzioni Ausiliarie

4.1 to_string

La funzione to string stampa la stringa con testa in s0 a video ed esegue un salto a loop in Lettura_*mycypher* se si è in fase di cifratura (s5 = 0), se invece si è in fase di decifratura (s5 = 1) il salto è a rev_loop sempre nel Lettura_*mycypher*.

4.2 interval_check

Questa funzione assegna al registro di ritorno a3 il valore corrispondente all'intervallo di appartenenza del registro a2, che contiene il carattere in esame della stringa in analisi. Se a2∈[97,122] allora è una lettera minuscola e a3 = 0, se a2∈[65,90] è una maiuscola, se a2 non appartiene a nessuno dei due sottointervalli allora a2 è un carattere speciale o un numero e quinid a3 = 2.

4.3 write_auxstr

La funzione ausliaria write_auxstr si occupa di scrivere la sezione della sottostringa del carattere occorrente relativa alla posizione.
Inizialmente viene scritto il carattere "-", successivamente è necessario convertire la posizione dell'occorrenza in uno o più caratteri, per fare ciò è utilizzato un ciclo dividing che inserisce nello stack i resti delle divisioni per 10 del valore intero che rappresenta la posizione. Questi saranno poi estratti, nel ciclo write_loop, dall'ultimo inserito al primo e inseriti in quest'ordine nella sottostringa.

4.4 convert_to_int

Questa funzione ha il compito opposto alla write_auxstr: i caratteri della sottostringa relativi alla posizione vengono inseriti nella pila,utile anche per contare il numero di cifre da convertire, e quindi le loro relative potenze di 10. Quando i caratteri vengono estratti sono convertiti in interi sottraendovi 48, successivamente sono moltiplicati per la relativa potenza di 10 (10,100,...): Il risultato viene di volta in volta sommato in a3 che è il valore convertito di ritorno dalla funzione.
In ultimo è necessario specificare che se nella pila è presente un solo elemento la moltiplicazione per potenze di 10 viene saltata.

4.5 swap_head

La swap_head si occupa di scrivere il carattere di fine stringa dopo il completamento della cifratura ad occorrenze o della sua decifratura, scambiando poi fra loro i puntatori s0 ed s4 relativi rispettivamente alle teste della stringa di partenza e della stringa ausiliaria. Permettendo così di continuare a cifrare la stringa utilizzando il puntatore s0.

5 Esempi di corretto funzionamento

Input

myplaintext = "Jack O' Lantern"
mycypher = "AEDB"
sostK = -1
blocKey = "-test"

Output:

- Izbj N' Kzmsdqm
- mqdsmzK 'N jbzI
- NJWHNAp 'm QYAr
- ;~|{".DEZAm%~tF
- NJWHNAp 'm QYAr
- mqdsmzK 'N jbzI
- Izbj N' Kzmsdqm
- Jack O' Lantern

Input

myplaintext = "Superstrada Fi-Pi-Li"
mycypher = "CBCA"
sostK = -1
blocKey = "Chi4ve"

Output:

- S-1 u-2 p-3 e-4 r-5-8 s-6 t-7 a-9-11 d-10 -12 F-13 i-14-17-20 –15-18 P-16 L-19
- VUZtKR5H9!iEhU]tHR8UatIR9H=!mEdUb!gV#,V%fE#UZ&Vk0Yt?R4V%mR5XI!cV8UZ,Vu0Y_t"R4a
- V-1-36-39-47-57-66-71 U-2-14-20-32-44-68 Z-3-45-69 t-4-16-22-52-76 K-5 R-6-18-24-54-60-78 5-7-61 H-8-17-26 9-9-25 !-10-28-34-64 i-11 E-12-30-42 h-13]-15 8-19-67 a-21-80 I-23-63 =-27 m-29-59 d-31 b-33 g-35#-37-43 ,-38-70%-40-58 f-41&-46 k-48 0-49-73 Y-50-74\-51-56 ?-53 4-55-79 X-62 c-65 u-72_-75 "-77
- U-1-36-39-47-57-66-71 T-2-14-20-32-44-68 Y-3-45-69 s-4-16-22-52-76 J-5 Q-6-18-24-54-60-78 5-7-61 G-8-17-26 9-9-25 !-10-28-34-64 h-11 D-12-30-42 g-13]-15 8-19-67 z-21-80 H-23-63 =-27 l-29-59 c-31 a-33 f-35#-37-43 ,-38-70%-40-58 e-41&-46 j-48 0-49-73 X-50-74\-51-56 ?-53 4-55-79 W-62 b-65 t-72_-75 "-77
- V-1-36-39-47-57-66-71 U-2-14-20-32-44-68 Z-3-45-69 t-4-16-22-52-76 K-5 R-6-18-24-54-60-78 5-7-61 H-8-17-26 9-9-25 !-10-28-34-64 i-11 E-12-30-42 h-13]-15 8-19-67 a-21-80 I-23-63 =-27 m-29-59 d-31 b-33 g-35#-37-43 ,-38-70%-40-58 f-41&-46 k-48 0-49-73 Y-50-74\-51-56 ?-53 4-55-79 X-62 c-65 u-72_-75 "-77
- VUZtKR5H9!iEhU]tHR8UatIR9H=!mEdUb!gV#,V%fE#UZ&Vk0Yt?R4V%mR5XI!cV8UZ,Vu0Y_t"R4a
- S-1 u-2 p-3 e-4 r-5-8 s-6 t-7 a-9-11 d-10 -12 F-13 i-14-17-20 –15-18 P-16 L-19
- Superstrada Fi-Pi-Li