

E10 – Esercizi sulle stringhe in C

Esercizio 1. Scrivere una funzione che riceve come parametro una stringa s e un carattere c e restituisce la *frequenza* di c in s , cioè il numero di occorrenze di c in s .

Esercizio 2. Utilizzando la funzione dell'esercizio precedente, scrivere una funzione che riceve come parametro una stringa s e stampa la frequenza in s di ciascun carattere dalla 'a' alla 'z'.

Scrivere un programma che legge una stringa dall'utente e stampa la frequenza in s di ciascun carattere dalla 'a' alla 'z'.

Esercizio 3. Scrivere una funzione che riceve come parametro una stringa s e la modifica sostituendo ogni carattere minuscolo con il corrispondente maiuscolo.

Scrivere poi un programma che usa la funzione scritta.

Esercizio 4. Scrivere una funzione che riceve una stringa s come parametro e la modifica invertendola. Ad esempio, se la stringa fosse "ciao", dopo la modifica dovrebbe essere "oaic".

Scrivere poi un programma che usa la funzione scritta.

Esercizio 5. Scrivere una funzione che riceve come parametri due stringhe $s1$ e $s2$ e due indici i e j , e copia in $s2$ la sottostringa di $s1$ che va dal carattere i -esimo al carattere j -esimo. Ad esempio, se $s1$ fosse "altalena", i fosse 2 e j fosse 5, in $s2$ dovrebbe essere inserita la stringa "tale".

Scrivere poi un programma che usa la funzione scritta.

Nota: nello scrivere la funzione si assuma che $s2$ sia lunga almeno quanto $s1$ e che gli indici siano validi.

Esercizio 6. Scrivere una funzione che realizza un semplice algoritmo crittografico. La funzione riceve come parametri una stringa $s1$ e un intero n e modifica $s1$ sostituendola con la sua versione criptata utilizzando il seguente semplice algoritmo di cifratura. Ogni carattere da 'a' a 'z' viene rimpiazzato dal carattere che si trova n posizioni più avanti nell'alfabeto. L'alfabeto va gestito circolarmente, cioè il carattere 'a' è successivo al carattere 'z'. Tutti i caratteri che non rientrano nell'intervallo da 'a' a 'z' vanno lasciati inalterati.

Scrivere anche una funzione che decodifica una stringa criptata (tale funzione deve cioè funzionare in maniera inversa rispetto alla precedente).

Scrivere infine un programma che permette di criptare e decriptare una stringa inserita dall'utente.

Soluzioni

Esercizio 1 - svolgimento.

```
int frequenza(char *s, char c){
    int count=0;
    for(int i=0;s[i]!='\0';i++)
        if(s[i]==c)
            count++;
    return count;
}
```

Esercizio 2 - svolgimento.

```
#include <stdio.h>

int frequenza(char *s, char c);

void frequenzaChars(char *s);

int main(){
    char s[40];
    printf("Inserisci una stringa\n");
    gets(s);
    frequenzaChars(s);
}

int frequenza(char *s, char c){
    int count=0;
    for(int i=0;s[i]!='\0';i++)
        if(s[i]==c)
            count++;
    return count;
}

void frequenzaChars(char *s){
    for(char c='a'; c<='z'; c++){
        int count=frequenza(s,c);
        printf("Frequenza di %c in %s: %d\n", c, s, count);
    }
}
```

Esercizio 3 - svolgimento.

```
#include <stdio.h>
#include <ctype.h>

int maiuscolo(char *s);

int main(){
    char s[40];
    printf("Inserisci una stringa\n");
    gets(s);
    maiuscolo(s);
    printf("Stringa in maiuscolo %s\n", s);
}

int maiuscolo(char *s){
    for(int i=0;s[i]!='\0';i++)
```

```

        if(islower(s[i]))
            s[i]=toupper(s[i]);
    }

```

Esercizio 4 - svolgimento.

```

#include <stdio.h>
#include <string.h>

int inverti(char *s);

int main(){
    char s[40];
    printf("Inserisci una stringa\n");
    gets(s);
    inverti(s);
    printf("Stringa invertita %s\n", s);
}

int inverti(char *s){
    int n=strlen(s);
    for(int i=0;i<n/2;i++){
        char tmp=s[i];
        s[i]=s[n-1-i];
        s[n-1-i]=tmp;
    }
}

```

Esercizio 5 - svolgimento.

```

#include <stdio.h>
#include <string.h>

int sottostringa(char *s1, char *s2, int i, int j);

int main(){
    char s1[40], s2[40];

    printf("Inserisci una stringa\n");
    gets(s1);

    int i,j;
    printf("Inserisci due indici\n");
    scanf("%d%d", &i, &j);

    sottostringa(s1, s2, i, j);
    printf("Sottostringa di %s da %d a %d: %s\n", s1, i, j, s2);
}

int sottostringa(char *s1, char *s2, int i, int j){
    int h;
    for(h=i;h<=j;h++){
        s2[h-i]=s1[h];
    }
    s2[h-i]='\0';
}

```

Esercizio 5 - svolgimento.

```
#include <stdio.h>
#include <string.h>

int cripta(char *s, int n);
int decripta(char *s, int n);

int main(){
    char s[40];

    printf("Inserisci una stringa\n");
    gets(s);
    printf("Inserisci la chiave di codifica\n");
    int n;
    scanf("%d",&n);

    cripta(s,n);
    printf("Stringa criptata: %s\n",s);
    decripta(s,n);
    printf("Stringa decriptata: %s\n", s);
}

int cripta(char *s, int n){
    for(int i=0;s[i]!='\0';i++){
        if(s[i]>='a' && s[i]<='z'){
            char c=s[i]; // carattere i-esimo
            c=c-'a'; // posizione rispetto alla 'a'
            c=c+n; // posizione traslata di n
            c=c%26; // gestione circolare dell'alfabeto
            s[i]='a'+c; // carattere codificato
        }
    }
}

int decripta(char *s, int n){
    for(int i=0;s[i]!='\0';i++){
        if(s[i]>='a' && s[i]<='z'){
            char c=s[i]; // carattere i-esimo
            c=c-'a'; // posizione rispetto alla 'a'
            c=c-n; // posizione traslata indietro di n
            c=c%26; // gestione circolare dell'alfabeto
            s[i]='a'+c; // carattere codificato
        }
    }
}
```