# Esercizi sulle stringhe

La *cifratura* è un processo di sostituzione, permutazione manuale ed automatica di simboli. Lo scopo di un criptosistema è di cifrare un testo intellegibile (in chiaro) producendo un testo cifrato(criptogramma). La *decifratura* rappresenta il processo inverso ovvero è l' operazione messa in atto dal destinatario per riottenere il testo in chiaro.

Dato un valore intero k, scrivere un programma per la cifratura e decifratura di stringhe lunghe al più 50 caratteri dove ogni carattere della stringa viene fatto shiftare di k caratteri.

Applicare l'algoritmo alla stringa costante "nelmezzodelcammindinostravita"

# Esercizio 1 (cont.)

```
#include <stdio.h>
  #define N 51
main(){
  char
  s[N]="nelmezzodelcammindinostravita";
  int i; int k;
  scanf("%d", &k);
  for (i=0; s[i]!='\setminus0'; i++) /*cifratura
      s[i]=s[i]+k;
  printf("Stringa cifrata: %s",s);
  for (i=0; s[i]!='\setminus 0'; i++) /*decifr.
      s[i]=s[i]-k;
  printf("Stringa decifrata: %s",s);
```

### **Esercizio 1 Variante**

Dato un *carattere* k, scrivere un programma per la cifratura e decifratura di stringhe lunghe al più 50 caratteri. Ogni carattere della stringa viene fatto shiftare di un numero di caratteri equivalente alla posizione nell'alfabeto del carattere k. Il risultato deve essere una stringa di caratteri dell'alfabeto 'a'-'z'.

Applicare l'algoritmo ad una stringa inserita da tastiera.

#### **NOTA**

È' necessario prevedere gli opportuni meccanismi di shifting con "rotazione" dei caratteri. Ad esempio, se ho il carattere 'z' e lo devo shiftare di 3 posizioni allora il risultato è 'c'.

# **Esercizio 1 Variante (cont.)**

```
#include <stdio.h>
#define N 51
main(){
  char s[N];
  int i;
  char k;
  scanf("%s",s);
  scanf("%c", &k);
  k=k-'a';
  for (i=0; s[i]!='\setminus 0'; i++) {
     s[i]=s[i]+k;
     if(s[i]>'z')
         s[i] = a' + (s[i] % z') -1;
  for (i=0; s[i]!='\setminus 0'; i++) {
     s[i]=s[i]-k;
     if(s[i]<'a')
         s[i]='z'-(('a'%s[i])-1);}
```

Modificare l'esercizio precedente affinché richieda la chiave di cifratura e poi consenta la cifratura/ decifratura di una sequenza di stringhe fornita da tastiera. Ogni stringa inserita dall' utente deve essere prima cifrata (con stampa del risultato) e poi decifrata. Scrivere quindi due varianti:

- 1. Viene prima richiesto all' utente il numero delle stringhe inserite e poi la sequenza di stringhe
- 2.La lunghezza della sequenza non è nota a priori

Scrivere un programma per la cifratura e decifratura con parola chiave di stringhe minuscole lunghe al più 50 caratteri.

Per ogni carattere c della stringa, si considera il carattere k di eguale posizione (modulo la lunghezza della parola chiave) nella parola chiave, e si fa shiftare c di un numero di caratteri equivalente alla posizione nell' alfabeto del carattere k. Il risultato deve essere una stringa di caratteri dell' alfabeto 'a' - 'z'.

Ad esempio data la stringa "ciao" e la parola chiave "no", la corrispondente parola cifrata è "pwnc".

Applicare l'algortimo alla stringa costante "nelmezzodelcammindinostravita" con parola chiave "dante".

# Esercizio 2 (cont.)

```
#include <stdio.h>
#include <string.h>
#define N 51
void main(){
  char s[N]="nelmezzodelcammindinostravita",
  crit[] = "dante";
  int i,k;
  k=strlen(crit);
/*cifratura*/
  for (i=0; s[i]!='\setminus 0'; i++) {
       c2=crit[i%k]-'a';
       s[i]=s[i]+c2;
       if(s[i]>'z') s[i]='a'+(s[i]%'z')-1;
/*decifratura*/
  for (i=0; s[i]!='\setminus 0'; i++) {
       c2=crit[i%k]-'a';
       s[i]=s[i]-c2;
       if(s[i]<'a') s[i]='z'-(('a'%s[i])-1);}
```

Utilizzando le funzioni della libreria string.h e date due stringhe sta e stb lunghe al più 99 caratteri inserite da tastiera scrivere un programma che

- 1.concatena la stringa sta alla stringa " " e memorizza il risultato in sta
- 2.concatena sta a stb e memorizza il risultato in sta,
- 3.stampa i caratteri della stringa risultante sta in ordine inverso.

#### **NOTA**

Le funzioni di string.h coinvolte sono strcat e strlen

# Esercizio 3 (cont.)

```
#include <stdio.h>
#include <string.h>
void main ()
char sta[100], stb[100]; int i;
printf("introdurre una stringa: ");
scanf("%s", &sta[0]);
strcat(sta," ");
printf("introdurre una stringa: ");
scanf("%s", &stb[0]);
strcat(sta, stb);
for(i=strlen(sta)-1; i>=0;i--)
    printf("%c",sta[i]);
```

#### Dati:

- un vettore contenente al più 100 nomi di persona (lunghezza <=20) forniti da tastiera</li>
- un nome

scrivere un programma che conta quante volte il nome compare nel vettore.

#### **NOTA**

```
Un vettore di stringhe di lunghezza <= 20 è definito come:
    char nomi[100][21];
In alternativa è possibile definire il tipo stringa:
    typedef char stringa[21];
E quinidi definire nomi come vettore di tipo stringa:
    stringa nomi[100];</pre>
```

# Esercizio 4 (cont.)

```
#include <stdio.h>
#include <string.h>
typedef char stringa[21];
int main ( )
stringa NOMI[100], NOMECERCATO;
int I, N, CONTA;
printf("QUANTI NOMI NEL VETTORE ? ");
scanf("%d", &N);
for (I=0; I<N; I++) {
 printf("introdurre il nome %d:", I+1);
 scanf("%s", NOMI[I]);
 /* oppure scanf("%s", &NOMI[I][0]); */}
```

# Esercizio 4 (cont.)

```
printf("introdurre il nome da cercare:");
scanf("%s", NOMECERCATO);

for (CONTA=0,I=0; I<N; I++) {
  if (strcmp(NOMI[I] , NOMECERCATO)==0)
  CONTA = CONTA + 1;}

printf("il nome compare N.volte= %d:", CONTA);
}</pre>
```

# Definizione di tipi in C: typedef

- Usare typedef per definire il tipo stringa fa sì che il compilatore aggiunga stringa alla lista dei nomi di tipi che è in grado di riconoscere
- Adesso stringa può essere utilizzata per dichiarare variabili, nelle espressioni di cast e in qualsiasi altro punto dove è richiesto un tipo

#### **ESEMPIO**

```
stringa NOMI[100], NOMECERCATO;
```

• Per il compilatore stringa è sinonimo di un char [21]

#### **ESEMPIO: Definizione del tipo Bool**

```
typedef int Bool;
Bool flag;
```

- Rivedere i programmi sui vettori ordinati assumendo che i vettori siano vettori di stringhe
- Scrivere un programma che trasforma una stringa lunga al più 20 caratteri sostituendo caratteri minuscoli con caratteri maiuscoli
- Scrivere un programma che data una stringa s lunga al più 40 caratteri, produce una stringa e che rappresenta il reverse della stringa s.
- Scrivere un programma che dato un vettore V di stringhe lunghe al più 20 caratteri e un carattere c, copi in un altro vettore tutte e sole le parole del vettore V che iniziano con il carattere c