Esercizio

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase è terminata dall'introduzione del carattere di invio e contiene complessivamente al più 100 caratteri.

Il programma deve svolgere le seguenti operazioni:

- visualizzare la frase inserita
- costruire una nuova frase in cui tutte le occorrenze del carattere '.' sono sostituite con il carattere di ritorno di linea '\n' . Il programma deve memorizzare la nuova frase in una opportuna variabile (stringa)
- visualizzare la nuova frase.

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

int main(void)
{
    const int MAXDIM = 100 ;           /* dimensione max stringa di caratteri */

    char frase[MAXDIM + 1] ;           /* stringa di caratteri inserita */
    char frasemodificata[MAXDIM + 1] ; /* nuova stringa modificata */
    int lung_stringa ;                 /* lunghezza della stringa inserita */
    int i ;                            /* indice dei cicli */

    /* LEGGI LA FRASE INSERITA DA TASTIERA */
    printf ("Inserisci una frase di al massimo %d caratteri: ", MAXDIM) ;
    gets(frase) ;

    /* CALCOLA LA LUNGHEZZA DELLA FRASE */
    lung_stringa = strlen(frase) ;

    /* STAMPA LA FRASE INSERITA */
    printf("La frase inserita e': ") ;
    puts(frase) ;
    printf("La frase contiene %d caratteri (inclusi gli spazi)\n", lung_stringa) ;

    /* ANALIZZA LA FRASE INSERITA CARATTERE PER CARATTERE. RICOPIA LA FRASE
    NELLA STRINGA "frase modificata". SE LA STRINGA INSERITA CONTIENE IL
    CARATTERE ".", SOSTITUISCILO CON IL CARATTERE DI RITORNO DI LINEA "\n" */
    for ( i=0; i<lung_stringa; i=i+1 )
    {
        if ( frase[i] == '.' )
            frasemodificata[i] = '\n' ;
        else
            frasemodificata[i] = frase[i] ;
    }
    frasemodificata[lung_stringa] = '\0' ;

    /* STAMPA LA FRASE MODIFICATA */
    printf("La frase modificata e': \n") ;
    puts(frasemodificata) ;
    return(0)
}
```

Esercizio

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase è terminata dall'introduzione del carattere di invio. La frase contiene sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al più 100 caratteri.

Il programma deve svolgere le seguenti operazioni:

- visualizzare la frase inserita;
- costruire una nuova frase tale che ogni lettera vocale presente nella frase di partenza sia seguita dalla lettera **'f'** (se la vocale è minuscola) o dalla lettera **'F'** (se la vocale è maiuscola) nella nuova frase. Il programma deve memorizzare la nuova frase in una opportuna variabile (stringa).
- visualizzare la nuova frase.

Ad esempio, la frase **VacAnze di NaTAle** diviene **VafcAFnzef dif NafTAFlef**

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

int main(void)
{
    const int MAXDIM = 100 ;           /* dimensione max stringa di caratteri */

    char frase[MAXDIM + 1] ;             /* stringa di caratteri inserita */
    char frasemodificata[2*MAXDIM + 1] ; /* nuova stringa modificata */
    int lung_stringa ;                   /* lunghezza della stringa inserita */
    int i, j ;                           /* indici dei cicli */

    /* LEGGI LA FRASE INSERITA DA TASTIERA */
    printf ("Inserisci una frase di al massimo %d caratteri: ", MAXDIM) ;
    gets(frase) ;

    /* CALCOLA LA LUNGHEZZA DELLA FRASE */
    lung_stringa = strlen(frase) ;

    /* STAMPA LA FRASE INSERITA */
    printf("La frase inserita e': ") ;
    puts(frase) ;
    printf("La frase contiene %d caratteri (inclusi gli spazi)\n", lung_stringa) ;

    /* COSTRUISCI LA NUOVA FRASE */
    /* L'INDICE "i" E' USATO PER SCORRERE LA STRINGA "frase". L'INDICE "j" E'
    USATO PER SCORRERE LA STRINGA "frasemodificata" */
    for ( i=0, j=0; i<lung_stringa; i++ )
    {
        /* RICOPIA IL CARATTERE IN "frase[i]" nella cella "frasemodificata[j]" */
        /* INCREMENTA IL CONTATORE "j" PER ACCEDERE ALLA CELLA SUCCESSIVA
        NELLA STRINGA "frasemodificata" */
        frasemodificata[j] = frase[i] ;
        j = j + 1 ;

        /* SE "frase[i]" CONTIENE UNA VOCALE MINUSCOLA,
        INSERISCI IL CARATTERE "f" NELLA CELLA "frasemodificata[j]" */
        /* INCREMENTA IL CONTATORE "j" PER ACCEDERE ALLA CELLA SUCCESSIVA
        NELLA STRINGA "frasemodificata" */
        if ( frase[i] == 'a' || frase[i] == 'e' || frase[i] == 'i'
                || frase[i] == 'o' || frase[i] == 'u')
        {
            frasemodificata[j] = 'f' ;
            j = j + 1 ;
        }
```

```
        else
        {
            /* SE "frase[i]" CONTIENE UNA LETTERA VOCALE IN CARATTERE MAIUSCOLO,
            INSERISCI IL CARATTERE "F" NELLA CELLA "frasemodificata[j]" */
            /* INCREMENTA IL CONTATORE "j" PER ACCEDERE ALLA CELLA SUCCESSIVA
            NELLA STRINGA "frasemodificata" */
            if ( frase[i] == 'A' || frase[i] == 'E' || frase[i] == 'I'
                    || frase[i] == 'O' || frase[i] == 'U' )
            {
                frasemodificata[j] = 'F' ;
                j = j + 1 ;
            }
        }
    }
    frasemodificata[j] = '\0' ;

    /* STAMPA LA FRASE MODIFICATA */
    printf("La frase modificata e': ") ;
    puts(frasemodificata) ;
    return(0)
}
```

Esercizio

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase è terminata dall'introduzione del carattere di invio. La frase contiene sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al più 100 caratteri. Il programma deve svolgere le seguenti operazioni:

- visualizzare la frase inserita
- costruire una nuova frase in cui il primo carattere di ciascuna parola nella frase di partenza è stato reso maiuscolo. Tutti gli altri caratteri devono essere resi minuscoli. Il programma deve memorizzare la nuova frase in una opportuna variabile
- visualizzare la nuova frase.

Ad esempio la frase cHe bEILA glOrnaTa diviene Che Bella Giornata .

```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int main(void)
{
    const int MAXDIM = 100 ;        /* dimensione massima stringa di caratteri */

    char frase[MAXDIM +1] ;         /* stringa di caratteri inserita */
    char nuovafrase[MAXDIM +1] ;    /* stringa di caratteri modificata */
    int lung_stringa ;              /* lunghezza della stringa inserita */
    int i ;                         /* indice dei cicli */

    /* LEGGI LA FRASE INSERITA DA TASTIERA */
    printf ("Inserisci una frase di al massimo %d caratteri: ", MAXDIM) ;
    gets(frase) ;

    /* CALCOLA LA LUNGHEZZA DELLA FRASE */
    lung_stringa = strlen(frase) ;

    /* STAMPA LA FRASE INSERITA */
    printf("La frase inserita e': ") ;
    puts(frase) ;
    printf("La frase contiene %d caratteri (inclusi gli spazi)\n", lung_stringa) ;
```

```
    /* COSTRUISCI LA NUOVA FRASE */
    for ( i=0; i<lung_stringa; i++ )
    {
        /* IL CARATTERE "frase[i]" E' LA PRIMA LETTERA DI UNA PAROLA SE IL
        CARATTERE PRECEDENTE ("frase[i-1]") ERA UNO SPAZIO OPPURE SE E' IL PRIMO
        CARATTERE DELLA FRASE (OSSIA i==0). IN QUESTO CASO IL CARATTERE "frase[i]"
        E' CONVERTITO IN CARATTERE MAIUSCOLO. IN TUTTI GLI ALTRI CASI IL CARATTERE
        "frase[i]" E' CONVERTITO IN CARATTERE MINUSCOLO */
        if ( (i==0) || isspace(frase[i-1]) )
            nuovafrase[i] = toupper(frase[i]) ;
        else
            nuovafrase[i] = tolower(frase[i]) ;
    }
    nuovafrase[lung_stringa] = '\0' ;

    /* STAMPA LA FRASE MODIFICATA */
    printf("La␣frase␣modificata␣e':␣") ;
    puts(nuovafrase) ;
    return(0)
}
```

Esercizio

Scrivere un programma in linguaggio C che legga da tastiera un numero binario puro sotto forma di una stringa di caratteri (0 o 1) lunga al massimo 24 bit.

Il programma deve:

- controllare che la stringa inserita sia corretta, vale a dire composta solo da caratteri 0 e 1
- convertire il numero binario inserito nell'equivalente valore decimale
- stampare sul video il valore decimale.

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

int main(void)
{
    const int MAXDIM = 24 ;        /* dimensione massima stringa di caratteri */

    char binario[MAXDIM + 1] ;     /* stringa contenente il numero binario */

    int num_cifre ;                /* numero di cifre nel numero binario */
    int decimale ;                 /* numero decimale risultante */
    int corretto ;                 /* flag per la ricerca */
    int i ;                        /* indice dei cicli */

    /* LEGGI IL NUMERO BINARIO */
    printf("Inserisci un numero binario puro di al massimo %d cifre: ", MAXDIM) ;
    gets(binario) ;

    /* CALCOLA IL NUMERO DI CIFRE DEL NUMERO BINARIO */
    num_cifre = strlen(binario) ;

    /* VISUALIZZA IL NUMERO INSERITO */
```

```
    printf("Il numero binario inserito e' %s e contiene %d cifre\n",
           binario, num_cifre);

    /* VERIFICA SE IL NUMERO INSERITO CONTIENE SOLO CARATTERI 0 E 1 */
    /* IL NUMERO BINARIO NON E' CORRETTO SE CONTIENE ALMENO UNA CIFRA DIVERSA
    SIA DA 0 CHE DA 1 */
    corretto = 1 ;
    for ( i=0 ; i<num_cifre; i++ )
        if ( binario[i]!='0' && binario[i]!='1' )
            corretto = 0 ;

    if ( corretto == 0 )
        printf("Il numero binario inserito non e' valido\n") ;
    else
    {
        /* CONVERTI IL NUMERO BINARIO NEL NUMERO DECIMALE CORRISPONDENTE */
        decimale = 0 ;
        for ( i=0; i<num_cifre; i++)
        {
            if ( binario[i] == '1' )
                decimale = 2*decimale + 1 ;
            else
                decimale = 2*decimale ;
        }

        /* STAMPA IL RISULTATO */
        printf("Il valore decimale e': %d\n", decimale) ;
    }
    return(0)
}
```

Esercizio

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase è terminata dall'introduzione del carattere di invio.

La frase contiene sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al più 100 caratteri.

Il programma dovrà stampare su schermo le seguenti informazioni:

- per ognuna delle lettere dell'alfabeto, il numero di volte che la lettera compare nella stringa
- il numero di consonanti presenti nella stringa
- il numero di vocali presenti nella stringa.

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

int main(void)
{
    const int MAXDIM = 100 ;        /* dimensione massima stringa di caratteri */
    const int NUMLETTERE = 26 ;     /* numero di lettere dell'alfabeto */

    char frase[MAXDIM +1] ;         /* stringa di caratteri inserita */
    int lung_stringa ;              /* lunghezza della stringa inserita */
    int vocali, consonanti ;        /* contatori numero di vocali e di consonanti */
    int contatori[NUMLETTERE];      /* memorizza il numero di occorrenze per
                                       ogni lettera */
    int posizione_alfabeto ;        /* posizione nell'alfabeto di una lettera */
    int i ;                         /* indice dei cicli */


    /* LEGGI LA FRASE INSERITA DA TASTIERA */
    printf ("Inserisci una frase di al massimo %d caratteri: ", MAXDIM) ;
    gets(frase) ;

    /* CALCOLA LA LUNGHEZZA DELLA FRASE */
    lung_stringa = strlen(frase) ;

    /* STAMPA LA FRASE INSERITA */
    printf("La frase inserita e': ") ;
```

```
puts(frase) ;
printf("La frase contiene %d caratteri (inclusi gli spazi)\n", lung_stringa) ;

/* AZZERA IL VETTORE DEI CONTATORI. OGNI CELLA DI QUESTO VETTORE E'
ASSOCIATA A UNA LETTERA DELL'ALFABETO. LA CELLA 0 ALLA LETTERA A,
LA CELLA 1 ALLA B E COSI' VIA */

for ( i=0; i<NUMLETTERE; i++ )
    contatori[i] = 0 ;

/* ANALIZZA LA FRASE LETTERA PER LETTERA E AGGIORNA IL VETTORE DEI CONTATORI */
for ( i=0; i<lung_stringa; i++ )
{
    if ( frase[i] >= 'A' && frase[i] <= 'Z' )
    {
        /* IL CARATTERE ESAMINATO E' UNA LETTERA MAIUSCOLA POICHE'
        IL SUO CODICE ASCII E' COMPRESO TRA QUELLI DELLE LETTERE A E Z.
        PER RICAVARE LA CELLA DEL VETTORE "contatori" DA INCREMENTARE
        DEVI IDENTIFICARE LA POSIZIONE DELLA LETTERA NELL'ALFABETO.
        POICHE' I CODICI ASCII DELLE LETTERE MAIUSCOLE SONO CONSECUTIVI,
        BASTERA' SOTTRARRE AL CARATTERE ESAMINATO IL CODICE ASCII DELLA
        PRIMA LETTERA DELL'ALFABETO ('A') */

        posizione_alfabeto = frase[i] - 'A' ;
        contatori[posizione_alfabeto] ++ ;
    }
    else
    {
```

```
        if ( frase[i] >= 'a' && frase[i] <= 'z' )
        {
            /* IL CARATTERE ESAMINATO E' UNA LETTERA MINUSCOLA POICHE'
            IL SUO CODICE ASCII E' COMPRESO TRA QUELLI DELLE LETTERE a E z.
            PER RICAVARE LA CELLA DEL VETTORE "contatori" DA INCREMENTARE
            DEVI IDENTIFICARE LA POSIZIONE DELLA LETTERA NELL'ALFABETO.
            POICHE' I CODICI ASCII DELLE LETTERE MINUSCOLE SONO CONSECUTIVI,
             BASTERA' SOTTRARRE AL CARATTERE ESAMINATO IL CODICE ASCII DELLA
            PRIMA LETTERA DELL'ALFABETO ('a') */

            posizione_alfabeto = frase[i] - 'a' ;
            contatori[posizione_alfabeto] ++ ;
        }
    }
}

/* STAMPA I CONTATORI DELLE VARIE LETTERE */
for ( i=0; i<NUMLETTERE; i=i+1 )
    printf ("La␣lettera␣%c␣compare␣%d␣volte␣\n",
            'A'+i , contatori[i]) ;

/* CALCOLA IL NUMERO DI VOCALI */
/* SOMMA IL NUMERO DI OCCORRENZE PRESENTI NEL VETTORE "contatori"
NELLE CELLE ASSOCIATE ALLE LETTERE A, E, I, O, U, Y */
vocali = contatori['A'-'A'] + contatori['E'-'A'] + contatori['I'-'A'] +
         contatori['O'-'A'] + contatori['U'-'A'] + contatori['Y'-'A'] ;
```

```
    /* CALCOLA IL NUMERO DI CONSONANTI */
    /* IL NUMERO DI CONSONANTI SI OTTIENE SOTTRAENDO DAL NUMERO COMPLESSIVO
    DI OCCORRENZE DI TUTTE LE LETTERE, IL NUMERO COMPLESSIVO DI VOCALI */

    consonanti = 0 ;
    for ( i=0; i<NUMLETTERE; i=i+1 )
        consonanti = consonanti + contatori[i] ;

    consonanti = consonanti - vocali ;

    /* STAMPA IL NUMERO DI VOCALI E CONSONANTI */
    printf ("Il numero di vocali e': %d\n", vocali) ;
    printf ("Il numero di consonanti e': %d\n", consonanti) ;
    return(0)
}
```

Esercizio

Si scriva un programma in linguaggio C che riceva in ingresso due parole inserite da tastiera.

Si consideri che ciascuna parola può contenere al massimo 30 caratteri. Il programma deve verificare se la seconda parola inserita è contenuta almeno una volta all'interno della prima parola (ossia se la seconda parola è una sottostringa della prima parola).

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

int main(void)
{
    const int MAXDIM = 30 ;             /* dimensione massima stringa di caratteri */

    char parola1[MAXDIM + 1] ;          /* prima stringa di caratteri inserita */
    char parola2[MAXDIM + 1] ;          /* seconda stringa di caratteri inserita */
    int lung_stringa1, lung_stringa2 ; /* lunghezza delle due stringhe inserite */

    int contenuto, finito ;             /* flag per la ricerca */
    int i, j ;                          /* indici dei cicli */

    /* LEGGI LA PRIMA PAROLA INSERITA DA TASTIERA */
    printf ("Inserisci una parola di al massimo %d caratteri: ", MAXDIM) ;
    gets(parola1) ;
    /* CALCOLA LA LUNGHEZZA DELLA PAROLA */
    lung_stringa1 = strlen(parola1) ;

    /* STAMPA LA PAROLA INSERITA */
    printf("La parola %s contiene %d lettere\n", parola1, lung_stringa1) ;

    /* LEGGI LA SECONDA PAROLA INSERITA DA TASTIERA */
    printf ("Inserisci una parola di al massimo %d caratteri: ", MAXDIM) ;
    gets(parola2) ;

    /* CALCOLA LA LUNGHEZZA DELLA PAROLA */
    lung_stringa2 = strlen(parola2) ;

    /* STAMPA LA PAROLA INSERITA */
    printf("La parola %s contiene %d lettere\n", parola2, lung_stringa2) ;

    /* VERIFICA SE "parola2" E' CONTENUTA IN "parola1" */
    if ( lung_stringa1 < lung_stringa2 )
        printf("La seconda parola e' piu' lunga della prima parola \n") ;
    else
    {
```

```
    /* IL CICLO FOR ESTERNO SCORRE LA STRINGA "parola1".
    PER OGNI CARATTERE "parola1[i]" IL CICLO FOR INTERNO ANALIZZA LA
    LA SOTTOSTRINGA CONTENENTE I CARATTERI COMPRESI TRA "parola1[i]"
    E "parola1[i+lung_stringa2-1]", E VERIFICA SE TALE SOTTOSTRINGA
    E' UGUALE A "parola2" */

    /* IL FLAG "finito==1" INDICA LA CONDIZIONE DI FINE RICERCA.
    IL FLAG E' INIZIALIZZATO A 0 E VIENE ASSEGNATO A 1 SE "parola2" E'
    CONTENUTA IN "parola1" */
    finito = 0 ;
    for ( i=0; i+(lung_stringa2-1)<lung_stringa1 && finito==0; i++ )
    {
        /* "j" E' L'INDICE DEL CICLO FOR INTERNO. VIENE UTILIZZATO PER
        SCORRERE I CARATTERI DELLA SOTTOSTRINGA "parola2" E DELLA
        SOTTOSTRINGA CONTENENTE I CARATTERI COMPRESI TRA "parola1[i]"
        E "parola1[i+lung_stringa2-1]" */

        /* IL FLAG "contenuto==1" INDICA CHE LE DUE SOTTOSTRINGHE SONO
        UGUALI. IL FLAG E' INIZIALIZZATO A 1 E VIENE ASSEGNATO A 0 SE
        ALMENO UN CARATTERE "parola1[i+j]" NELLA SOTTOSTRINGA E' DIVERSO
        DAL CORRISPONDENTE CARATTERE "parola2[j]" */
        contenuto = 1 ;
        for ( j=0; j<lung_stringa2 && contenuto==1; j++ )
        {
            if ( parola1[i+j] != parola2[j] )
                contenuto = 0 ;
        }

        /* SE AL TERMINE DEL CONFRONTO TRA LE DUE STRINGHE "contenuto" E'
        ANCORA UGUALE A 1, ALLORA "parola2" E' CONTENUTA IN "parola1".
        IL FLAG "finito" VIENE AGGIORNATO, E SI CONCLUDE LA RICERCA */
        if ( contenuto==1 )
            finito = 1 ;
    }
}

/* STAMPA IL RISULTATO */
if ( contenuto == 1 )
    printf("La␣seconda␣parola␣e'␣contenuta␣nella␣prima␣\n") ;
else
    printf("La␣seconda␣parola␣non␣e'␣contenuta␣nella␣prima␣\n") ;

return(0)
}
```

```c
#include <stdlib.h>
#include <string.h>

int main(void)
{
    const int MAXDIM = 30 ;             /* dimensione max stringa di caratteri */

    char parola1[MAXDIM + 1] ;          /* prima stringa di caratteri */
    char parola2[MAXDIM + 1] ;          /* seconda stringa di caratteri */
    int lung_stringa1, lung_stringa2 ; /* lunghezza delle due stringhe */

    /* LEGGI LA PRIMA PAROLA INSERITA DA TASTIERA */
    printf ("Inserisci una parola di al massimo %d caratteri: ", MAXDIM) ;
    gets(parola1) ;

    /* CALCOLA LA LUNGHEZZA DELLA PAROLA */
    lung_stringa1 = strlen(parola1) ;

    /* STAMPA LA PAROLA INSERITA */
    printf("La parola %s contiene %d lettere\n", parola1, lung_stringa1) ;

    /* LEGGI LA SECONDA PAROLA INSERITA DA TASTIERA */
    printf ("Inserisci una parola di al massimo %d caratteri: ", MAXDIM) ;
    gets(parola2) ;

    /* CALCOLA LA LUNGHEZZA DELLA PAROLA */
    lung_stringa2 = strlen(parola2) ;

    /* STAMPA LA PAROLA INSERITA */
    printf("La parola %s contiene %d lettere\n", parola2, lung_stringa2) ;

    /* VERIFICA SE "parola2" E' CONTENUTA IN "parola1" */
    if ( lung_stringa1 < lung_stringa2 )
        printf("La seconda parola e' piu' lunga della prima parola \n") ;
    else
    {
        if ( strstr(parola1, parola2) != NULL )
            printf("La seconda parola e' contenuta nella prima \n") ;
        else
            printf("La seconda parola non e' contenuta nella prima \n") ;
    }
    return(0)
}
```