

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 1: STRINGHE - VOCALI CONSECUTIVE
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* scrivere un programma che data una stringa di lunghezza massima 100
* stampi le occorrenze di coppie di vocali identiche consecutive
*/

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

#define MAXLEN 100

int main(){

    char stringa[MAXLEN];
    int len, i, cont = 0;

    typedef enum {false, true} boolean;
    boolean vocale;

    printf("inserire una stringa\n");
    scanf("%s", stringa);
    len = strlen(stringa);

    for(i=0; i<len-1; i++){

        vocale = false;
        if(stringa[i] == 'a' || stringa[i] == 'e' || stringa[i] == 'i' || stringa[i] == 'o'
|| stringa[i] == 'u'){
            vocale = true;
        }

        if(vocale == true && stringa[i+1] == stringa[i]){
            cont++;
            printf("la vocale ripetuta e' %c\nil numero di coppie e' %d\n", stringa[i],
cont);
        }

    }

    if(cont == 0)
        printf("non ci sono vocali ripetute consecutivamente\n");

    system("PAUSE");
    return 0;

}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 2: STRINGHE - ARRAY
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* scrivere un programma che prenda stringhe di lunghezza massima 100
  (quando l'utente inserisce la stringa "stop" il programma smette di chiedere
inserimenti)
  calcolare quante parole di lunghezza l (definita dall'utente), l+1 e l+2 sono state
inserite
*/

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

#define MAXLEN 100

int main(){

    char stringa[100];
    int l, len, i, count[3]={0,0,0};

    printf("inserire la lunghezza desiderata\n");
    scanf("%d", &l);

    do{

        printf("inserire una stringa\ninserire stop per terminare\n");
        scanf("%s", stringa);
        len = strlen(stringa);

        if(len >= l && len<=l+2 && strcmp(stringa, "stop") != 0){
            printf("%d\n", len-l);
            count[len-l]++;
        }

    }while(strcmp(stringa,"stop") != 0);

    for(i=0; i<3; i++){
        printf("il numero di parole di lunghezza %d inserite e': %d\n", l+i, count[i]);
    }

    system("PAUSE");
    return 0;

}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 3: STRINGHE - LETTERE
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* il programma acquisisce una sequenza di caratteri in input
* tale sequenza e' terminata dal carattere #
* conteggiare quante volte ogni lettera minuscola e' inserita
*/

#include <stdio.h>

#define costante 97

int main(){

    int lettere[26], pos, i;
    char c;

    for(i = 0; i <26; i++){
        lettere[i] = 0;
    }

    do{

        printf("inserire un carattere\n");
        scanf("\n%c", &c);

        if((c >= 'a') && (c <= 'z')){
            pos = c - costante;
            printf("posizione dell'array %d\n", pos);
            lettere[pos] = lettere[pos] + 1;
        }

    }while(c!='#');

    for(i = 0; i < 26; i++){
        printf("il numero di %c insierite e' %d \n", i + costante ,lettere[i]);
    }

    return 0;
}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 4: STRINGA PALINDROMA
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* il programma acquisisce una sequenza di caratteri di lunghezza massima 100 (terminata
da #)
determinare se la parola composta da tali caratter (considerati nell'ordine inserito)
e' palindroma
*/

#include <stdio.h>

#define MAXLEN 100

int main(){

    char lettere[MAXLEN], max = 0;
    char c;
    int i, j;
    typedef enum {false, true} boolean;
    boolean found = true;

    do{

        printf("inserire un carattere, #per terminare\n");
        scanf("\n%c", &c);

        if((c >= 'a') && (c <= 'z')){
            lettere[max] = c;
            max++;
        }

    }while(c!='#' && max < MAXLEN);

    i=0;
    j=max-1;
    while(found && i<j){
        if(lettere[i] != lettere[j]){
            found = false;
        }
        i++;
        j--;
    }

    if(found)
        printf("la stringa e' palindroma\n");
    else
        printf("la stringa non e' palindroma\n");

    return 0;
}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 5: STRINGHE
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* scrivere un programma che legge due stringhe (di lunghezza massima 100) e verifica se
sono uguali
se le due stringhe sono diverse le concatena in ordine alfabetico
*/

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

#define MAXLEN 100

int main(){

    char s1[MAXLEN], s2[MAXLEN];

    printf("inserire la prima stringa\n");
    scanf("%s", s1);
    printf("inserire la seconda stringa\n");
    scanf("%s", s2);

    if(strcmp(s1,s2) == 0){
        printf("le stringhe sono uguali\n");
    }else if(strcmp(s1,s2)>0){
        strcat(s2,s1);
        printf("la stringa concatenata e' %s\n", s2);
    }else{
        strcat(s1,s2);
        printf("la stringa concatenata e' %s\n", s1);
    }

    system("PAUSE");
    return 0;
}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 6: MATRICE
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* scrivere un programma che legge una matrice quadrata di dimensioni specificate
dall'utente (al massimo 10 righe e 10 colonne)
    calcolare la somma dei valori sulla diagonale principale
    calcolare la somma dei valori sopra la diagonale principale
    calcolare la somma dei valori sotto la diagonale principale
*/

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

#define MAXDIM 10

int main(){

    int matrice[MAXDIM][MAXDIM];
    int sup = 0, inf = 0, diag = 0;
    int dim, i, j;

    printf("inserire il numero di righe della matrice (massimo 10)\n");
    scanf("%d", &dim);

    for(i=0; i<dim; i++){
        for(j=0; j<dim; j++){
            printf("inserire un numero\n");
            scanf("%d", &matrice[i][j]);

            if(i==j)
                diag = diag + matrice[i][j];
            else if (i < j)
                sup = sup + matrice[i][j];
            else
                inf = inf + matrice[i][j];

        }
    }

    printf("la somma delle diagonale e': %d\n", diag);
    printf("la somma della parte superiore alla diagonale e': %d\n", sup);
    printf("la somma della parte inferiore alla diagonale e': %d\n", inf);

    system("PAUSE");
    return 0;

}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 7: MATRICE
* Tratto da: sofiaceppi.com/?page_id=226
* -----
* Si costruisca una matrice con 4 righe 5 colonne e valori dati
  dall'utente. La si stampi a video e si riporti la somma dei valori per
  ogni riga e per ogni colonna.
*/

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
    int mat[4][5];
    int righe[4];
    int colonne[5];
    int i, j;

    for(i=0; i<4; i++)
        righe[i] = 0;

    for(j=0; j<5; j++)
        colonne[j] = 0;

    /** inserimento dati **/
    for(i=0; i<4; i++){
        for(j=0; j<5; j++){
            printf("inserire un numero intero\n");
            scanf("%d", &mat[i][j]);
        }
    }

    /** stampare la matrice a video e calcolare somme di righe e colonne **/
    for(i=0; i<4; i++){
        for(j=0; j<5; j++){
            printf("%d ", mat[i][j]);
            righe[i] = righe[i] + mat[i][j];
            colonne[j] = colonne[j] + mat[i][j];
        }
        printf("\n");
    }

    /** stampare le somme delle righe**/
    for(i=0; i<4; i++){
        printf("somma riga %d = %d\n", i+1, righe[i]);
    }

    /** stampare le somme delle colonne**/
    for(j=0; j<5; j++){
        printf("somma colonna %d = %d\n", j+1, colonne[j]);
    }

    system("PAUSE");
    return 0;
}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 4
* ESERCIZIO 8: STRUCT
* -----
* Scrivere un programma per la gestione di un autosalone
* che memorizzi per ogni auto modello, targa, colore, prezzo.
* Dopo l'inserimento permettere all'utente di cercare
* un'auto inserita tramite la targa.
*/

```

```

//===== ESEMPIO A =====

```

```

#include <stdio.h>
#define MAX_AUTO 100

int main(){

    typedef struct {

        char nome[16];
        char targa[7];
        char colore[7];
        float prezzo;

    } automobile;

    automobile autosalone[MAX_AUTO];
    int numero_auto,i;

    do {
        printf("\nQuante auto ci sono nell'autosalone? ");
        scanf("%d",&numero_auto);

        if (numero_auto <= 0 || numero_auto >= MAX_AUTO)
            printf("Numero inserito non valido\n");

    } while(numero_auto <= 0 || numero_auto >= MAX_AUTO);

    for (i = 0; i < numero_auto; i++){
        fflush(stdin);

        printf("Auto %d -- NOME: ", i);
        scanf("%s",autosalone[i].nome);

        printf("Auto %d -- TARGA: ", i);
        scanf("%s",autosalone[i].targa);

        printf("Auto %d -- COLORE: ", i);
        scanf("%s",autosalone[i].colore);

        printf("Auto %d -- PREZZO: eur. ", i);
        scanf("%f",&autosalone[i].prezzo);

        printf("\n\n\n");
    }
}

```



```

char targa_ricerca[7];
typedef enum{false,true} bool;
bool auto_trovata = false;

printf("-----\n");
printf("Cerca auto tramite targa...\n");
printf("TARGA: ");
scanf("%s",targa_ricerca);

printf("\nCerco la targa %s\n", targa_ricerca);

for (i = 0; i<numero_auto; i++){
    if (strcmp(targa_ricerca,autosalone[i].targa) == 0){

        auto_trovata = true;

        printf("Trovata auto con targa %s\n", autosalone[i].targa);
        printf(" - NOME: %s\n", autosalone[i].nome);
        printf(" - COLORE: %s\n", autosalone[i].colore);
        printf(" - PREZZO: %f eur\n", autosalone[i].prezzo);

        break;
    }
}

if (!auto_trovata)
    printf("Nessuna auto con targa %s e' stata trovata\n", targa_ricerca);

return 0;
}

```