

ESERCIZIO n. 1**(Uso dei vettori)**

Scrivere un programma in linguaggio C che verifichi se una parola inserita dallo standard input è palindroma e in caso contrario stampa a video la parola inserita partendo dall'ultimo carattere sino al primo.

Le parole palindrome sono quelle che possono essere lette indifferentemente da sinistra a destra e da destra a sinistra.

Esempio:

"anna", "abba", "onorarono", "radar", "a" (parola di una lettera).

Soluzione:

```
#include <iostream.h>
#include <stdlib.h>
#define MAXLEN 100

int main( ) {

    char parola[MAXLEN] ;
    int i, n;

    cout << "\nparola = ";
    cin >> parola;

    for (n=0; parola[n] !='\0'; n++);

    i = 0;
    while ((i < n/2)&&(parola[i] == parola[n-i-1])) {
        i = i+1;
    }

    if (i < n / 2) {
        cout << "\nNO palindroma!\n";
        cout << "\nInversa = ";
        for (i = n-1; i >=0; i--)
            cout << parola[i];
    }
    else cout << "\npalindroma";

    system("PAUSE");
    return 0;
}
```

ESERCIZIO 2 (Definizione funzioni – passaggio parametri per copia)

Scrivere una funzione per definire se un numero è primo

```
#include <stdlib.h>
#include <iostream.h>

bool primo(int numero);

int main() {
    int numero;
    int j;

    cout << "Inserire il numero ";
    cin >> numero;

    if ( primo(numero) )
        cout << "Numero primo ";
    else
        cout << "Numero non primo ";

    system("PAUSE");
    return 0;
}

bool primo(int a) {
    int j;
    bool varprimo; /* esempio d'uso di una variabile booleana con */
                  /* funzionalità di marcatore di una condizione. */

    varprimo = true;
    for (j=2;j<=a/2;j++) {
        if (a%j == 0)
            varprimo = false;
    }
    return varprimo;
}
```

ESERCIZIO 3 (Definizione funzioni – passaggio parametri per reference)

Scrivere una funzione che permette di scambiare il valore di due variabili

```

#include <iostream.h>
#include <stdlib.h>

void scambia(int& a, int& b);

int main() {
    int numero1 , numero2;
    int j;

    cout << "Inserire il numero 1 ";
    cin >> numero1;
    cout << "Inserire il numero 2 ";
    cin >> numero2;

    scambia(numero1 , numero2);

    cout << "\nIn main--Numero 1 ora vale : " << numero1 << endl;
    cout << "\nIn main--Numero 2 ora vale : " << numero2 << endl;

    system("PAUSE");
    return 0;
}

void scambia(int &a , int &b) {

    int c;
    c = a ;
    a = b ;
    b = c ;
    cout << "\nIn scambia--a ora vale : " << a << endl;
    cout << "\nIn scambia--b ora vale : " << b << endl;
}

/*Provare a riscrivere il programma dichiarando e definendo la
funzione scambia con il seguente prototipo (parametri per copia):
        void scambia(int a , int b)
*/

```

ESERCIZIO n. 4 (Uso di funzioni per la manipolazione di stringhe)

Facendo uso dei comandi messi a disposizione dalla libreria standard del C definiti in <string.h>, risolvere il seguente problema.

Leggere due stringhe str1, str2 dallo standard input, stampare a video lunghezza di str1 e str2, stampare il risultato della concatenazione delle due stringhe: str1 concatenata str2, confrontare le due stringhe e stamparle in ordine alfabetico.

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 100

int main( ) {

char str1[MAX_LEN], str2[MAX_LEN],str3[2*MAX_LEN];
int len1,len2,cmp;

cout << "\nInserire la prima stringa, str1:";
cin >> str1;
cout << "\nInserire la prima stringa, str2:";
cin >> str2;

len1 = strlen(str1);
len2 = strlen(str2);
cout << "\nla lunghezza della 1-ma stringa e': " << len1;
cout << "\nla lunghezza della 2-da stringa e': " << len2;

strcpy(str3,str1);
strcat(str3,str2);
cout << "\nIl risultato della concatenazione e': " << str3;

cmp = strcmp(str1,str2);
if (cmp > 0) {

    cout << "\n str1 > str2!";
    cout << "\n 1. " << str2 << " \n 2. " << str1;

} else if (cmp < 0) {
    cout << "\n str2 > str1!";
    cout << "\n 1. " << str1 << " \n 2. " << str2;
}
else
    cout << "\n str1 == str2 : " << str1;
return 0;
}
```

Esercizio 5 (Definizione funzioni con parametri di tipo vettore + algoritmi)

Si progetti e codifichi una funzione C che riceve come parametri due stringhe che rappresentano due parole e restituisce un valore intero, da interpretarsi come valore di verità, che indichi se le due parole sono **anagrammi**, cioè se sono ottenibili l'una dall'altra tramite una permutazione delle lettere che le compongono.

Ad esempio le parole POLENTA e PENTOLA sono anagrammi. Si presti attenzione al fatto che parole come TAPPO e PATTO non sono anagrammi, anche se ogni lettera dell'una è contenuta nell'altra.

```
// Restituisce 1 se le due parole sono una l'anagramma dell'altra,
// 0 altrimenti
int anagrammi(char a[], char b[])
{
    int len = strlen(a),          // lunghezza di a
        contA, contB,            // contatori di occorrenze di una lettera
        i, k;                    // indici di servizio

    if ( len != strlen(b) ) // se non sono lunghe uguali è banale
        return 0;

    for ( i = 0 ; i < len ; i++ ) { // per ogni char a[i] in a (escluso \0)
        contA = contB = 0;
        for ( k = 0 ; k < len ; k++ ) { // scandisco entrambe le stringhe
            if ( a[k] == a[i] )
                ++contA;                // conto le occorrenze di a[i] in a
            if ( b[k] == a[i] )
                ++contB;                // e anche le occorrenze di a[i] in b
        }
        if ( contA != contB ) // se un conteggio non corrisponde -> 0
            return 0;
    }
    return 1;                    // se corrispondono tutti -> 1
}
```

ESERCIZIO 6 (Scomposizione in sottoprogrammi e passaggio parametri)

Scrivere un programma che acquisita una frase permetta a scelta dell'utente di

1. Stamparla
2. Stamparla invertita
3. Stampare la stringa in maiuscolo
4. Stamparla eliminando gli spazi
5. Contare il numero di parole presenti
6. Permetta di sostituire un carattere con un altro
7. Calcoli la ricorrenza di una determinata lettera dell'alfabeto sia essa minuscola o maiuscola
8. Uscire dal programma

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

int menu();
void lettura(char a[]);
void stampa(char a[] );
void inverti(char source[] , char dest[]);
void maiuscolo(char source[] , char dest[]);
void nospazi(char source[] , char dest[]);
int numparole(char a[] );
void cambiacar(char a[] , char oldc, char newc);
int ricorrenza(char a[] , char x);

int main() {
    int scelta;
    char frase[100] ;
    char frasel[100] ;
    char vecchio , nuovo ;

    lettura(frase);
    do {
        scelta=menu();

        switch (scelta) {
            case 1 : stampa(frase) ;
                    break ;
            case 2 : inverti(frase , frasel) ;
                    stampa(frasel) ;
                    break ;
            case 3 : maiuscolo(frase , frasel) ;
                    stampa(frasel) ;
                    break ;
```

```
        case 4 : nospazi(frase , frasel) ;
                stampa(frasel) ;
                break ;
        case 5 : cout << "Numero di parole: ";
                cout << numparole(frase) << endl;
                system("PAUSE");
                break ;
        case 6 : cout << "Carattere da sostituire : ";
                cin >> vecchio ;
                cout << "Nuovo carattere : ";
                cin >> nuovo;
                cambiacar(frase,vecchio,nuovo) ;
                break ;
        case 7 : cout << "Carattere da esaminare : ";
                cin >> vecchio;
                cout << "Il carattere compare ";
                cout << ricorrenza(frase , vecchio);
                cout << " volte" << endl ;
                system("PAUSE");
                break ;
    } // end switch

} while(scelta != 0);
return 0;
} // end main

int menu() {
    int s;
    do {
        system("cls");
        cout << " - 1 - Stampa" << endl ;
        cout << " - 2 - Inverti" << endl ;
        cout << " - 3 - Maiuscolo" << endl ;
        cout << " - 4 - Senza Spazi" << endl ;
        cout << " - 5 - Numero Parole" << endl ;
        cout << " - 6 - Cambia carattere" << endl ;
        cout << " - 7 - Ricorrenza di una lettera" << endl ;
        cout << " - 0 - Esci" << endl ;
        cout << "Inserire la scelta ";
        cin >> s;
    } while(s<0 || s>7);
    return(s);
}
```

```
void lettura(char f[]) {
    cout << "Inserire la frase : " ;
    cin.getline(f, 100, '\n') ;
}

void stampa(char f[]) {
    cout << f ;
}

void inverti(char f[] , char f1[]) {
    int j ;
    int len;
    len= strlen(f);

    for (j=0;j<len;j++) {
        f1[j] = f[len-1-j];
    }
    f1[len] = '\0';
}

void maiuscolo(char f[], char f1[]) {
    int len;
    int j;
    len= strlen(f);

    for (j=0;j<len;j++) {
        if (f[j] >= 'a' && f[j] <= 'z')
            f1[j] = f[j] - (char)((int)'a'-(int)'A');
        else
            f1[j] = f[j];
    }
    f1[j] = '\0';
}

void nospazi(char f[] , char f1[]) {
    int len;
    int j;
    len= strlen(f);

    for (j=0;j<len;j++) {
        if (f[j] != ' ') {
            f1[j] = f[j];
        }
    }
    f1[j] = '\0';
}
```



```
int numparole(char f[]) {
    int nump=0;
    int len,j;
    len = strlen(f);

    j = 0;
    while (f[j] != '\0') {
        while(f[j] != ' ' && f[j] != '\0')
            j++;
        nump++;
        while (f[j] == ' ')
            j++;
    }
    cout << "Numero di parole: " << nump << endl;
    return (nump) ;
}

void cambiacar(char f[] , char v , char n) {
    int j = 0;

    while (f[j] != '\0') {
        if ( f[j] == v )
            f[j] = n ;
        j++;
    }
}

int ricorrenza(char f[] , char c) {
    int conta = 0, j = 0 ;
    while (f[j] != '\0') {
        if (f[j] == c ) conta++ ;
        j++;
    }
    return (conta) ;
}
```

Esercizio 7 (Funzioni, Algoritmi, passaggio parametri con Matrici)

Si definisca una funzione C di prototipo

```
int controlla(char m[][N])
```

che (1) riceve come unico parametro una matrice quadrata NxN (dove N è una costante positiva già definita), (2) controlla se la parola leggibile sulla diagonale principale è leggibile anche in una delle righe o colonne della matrice stessa, e (3) restituisce 1 in caso affermativo, 0 altrimenti.

Esempi (per N = 5):

E	N	E	A	S
E	S	S	O	R
X	I	A	A	D
G	O	M	M	A
B	R	E	V	E

⇓
1

A	R	C	A	A
I	N	A	N	E
P	A	S	S	A
A	N	S	I	A
D	A	Z	A	A

⇓
1

P	E	N	S	O
P	A	U	R	A
A	C	U	T	A
C	U	O	R	E
M	A	M	M	A

⇓
1

C	C	Y	A	M
P	A	U	R	A
D	O	L	O	R
E	S	A	M	E
A	N	S	I	A

⇓
0

Per i che va da 0 a $N-1$ scandisco dapprima la i -esima riga e poi la i -esima colonna, controllando (un carattere alla volta) l'identità tra la parola in diagonale e la parola via via scandita.

Quando trovo due caratteri diversi in posizioni corrispondenti posso interrompere subito la scansione corrente e passare alla riga o colonna "successiva".

La prima volta che trovo interamente la parola cercata posso terminare la funzione restituendo 1, mentre solo dopo aver controllato anche l'ultima colonna posso restituire 0.

```
#include <iostream.h>
#include <stdlib.h>
#define N 5

int controlla(char m[][N]);

int main( ) {
    int matrix[N][N], esito;
    /*..... riempimento della matrice.....*/
    esito = controlla(matrix);

    if (esito ==1) cout << "\nCONTROLLO POSITIVO!";
    else cout << "\nCONTROLLO NEGATIVO!";

    system("PAUSE");
    return 0;
}
```

```
int controlla(char m[][N]) {
    int i, j, trovato;
    for( i=0; i<N; i++ ) /* Ogni iterazione controlla */
    { /* una riga e una colonna */
        j = 0;
        trovato = 1;
        while ( (trovato==1) && (j<N) ){/* Scansione della i-esima riga */
            if ( m[j][j] != m[i][j] )
                trovato = 0; /* carattere diverso -> parola non trovata */
            j++;
        }
        if ((j==N)&&(trovato==1)) /* Può restituire 1 appena trova la parola */
            return 1;
        j = 0; /* reset per controllare la colonna */
        trovato = 1;
        while ( (trovato==1) && (j<N) ){/*Scansione della i-esima colonna */
            if ( m[j][j] != m[j][i] )
                trovato = 0; // idem...
            j++;
        }
        if ( j==N && trovato )
            return 1;
    }
    return 0;
}
```

Esercizio 8 (Funzioni, Algoritmi, passaggio parametri con Matrici)

Si progetti e codifichi una funzione C che avendo come parametri d'ingresso una matrice di interi e l'ordine di tale matrice, riempia le celle della matrice con i valori corrispondenti di un quadrato magico di dimensione n , con n dispari.

Un quadrato magico di ordine n contiene i primi n numeri naturali (1, 2, 3, ..., n^2) disposti in modo tale che la somma dei numeri su ogni riga, su ogni colonna e sulle due diagonali principali sia sempre la stessa.

Es: $n = 3$

```

4 9 2  0 0 0  0 0 2  0 0 2  4 0 2  4 0 2  4 0 2  4 0 2  4 0 2  4 0 2  4 9 2
3 5 7 → 0 0 0 → 0 0 0 → 3 0 0 → 3 0 0 → 3 5 0 → 3 5 0 → 3 5 7 → 3 5 7 → 3 5 7
8 1 6  0 1 0  0 1 0  0 1 0  0 1 0  0 1 0  0 1 0  0 1 6  0 1 6  8 1 6  8 1 6

```

Esiste una regola molto semplice per percorrere la matrice disponendo i numeri interi in ordine crescente. Partendo col posizionare un 1 nella posizione centrale sull'ultima riga, si percorre la matrice incrementando di una unità il numero di riga e il numero di colonna dell'elemento attuale, avendo cura di considerare i bordi opposti della matrice come adiacenti. Se durante questa operazione si individua una cella vuota si scrive il numero successivo; altrimenti, il numero successivo, viene posizionato nella cella avente riga immediatamente superiore a quella dell'ultimo numero inserito. (vedi esempio)

```

#include <iostream.h>
#include <stdlib.h>
#define MAX_DIM 51

void quadratoMagico(int mat[][MAX_DIM], int n) ;

int main( ) {
    int matrix[MAX_DIM][MAX_DIM];
    int i,j,lim,sum;

    do {
        cout << "\ndim. quadrato (dispari e <= " << MAX_DIM;
        cin >> lim;
    } while (( lim > MAX_DIM ) || (lim % 2 == 0));

    quadratoMagico(matrix, lim) ;

    sum = 0;
    for (j = 0; j < lim; j++)
        sum+=matrix[0][j];

    cout << "\nIl quadrato magico di ordine " << lim << " e':\n";
    cout << "\nLa somma su ogni linea e' " << sum << "\n";

    for (i = 0; i < lim; i++) {
        cout << "\n";
        for (j = 0; j < lim; j++)
            cout << matrix[i][j];
    }
    return 0;
}

void quadratoMagico(int mat[][MAX_DIM], int n) {

    int i, j, k ;

    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            mat[i][j] = 0;

    i = n - 1;
    j = n / 2;
    for (k = 0; k < n*n; k++) {

        if (mat[i][j] != 0) i = (i-1) % n;
        mat[i][j] = k+1;
        i = (i+1) % n;
        j = (j+1) % n;
    }
}

```