



# IEIM 2016-2018

## Esercitazione VIII *“Il Gioco dell’impiccato”, Esercizi su Array E Stringhe*

Alessandro A. Nacci

[alessandro.nacci@polimi.it](mailto:alessandro.nacci@polimi.it) - [www.alessandronacci.it](http://www.alessandronacci.it)



# Il gioco dell'impiccato

- Scrivere un programma che permetta di giocare al gioco dell'impiccato



-----  
A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z

```
bash
c o _ _ _ _ _
Inserisci una lettera
m
Hai a disposizione ancora 8 tentativi
c o m _ _ _ _
Inserisci una lettera
r
Hai a disposizione ancora 8 tentativi
c o m _ _ _ r
Inserisci una lettera
t
Hai a disposizione ancora 8 tentativi
c o m _ _ t _ r
Inserisci una lettera
p
Hai a disposizione ancora 8 tentativi
c o m p _ t _ r
Inserisci una lettera
u
Hai a disposizione ancora 8 tentativi
c o m p u t _ r
Inserisci una lettera
e
Hai a disposizione ancora 8 tentativi
c o m p u t e r
Complimenti!
MacBook-Pro-di-Alessandro-Nacci:es5 alessandronacci$
```



# Vediamo cosa ci serve...

- L'idea è quella di avere
  - un dizionario di parole da indovinare
  - un numero massimo di tentativi
  - lo stato delle parola (quali lettere sono state indovinate?)



# Dichiarazione ed inizializzazione variabili

```
//Dizionario di parole
char dizionario[5][20]={"computer", "proiettore", "aula", "ciao", "telecomando"};

int i;
int tentativi=0;
int found=0;

//Puntatore alla parola da indovinare, scelta casualmente
char* parola_da_indovinare;
parola_da_indovinare = dizionario[scegliParola()];

//Campo di gioco, stato della parola
char stato_parola[20];

//Lunghezza della parola da indovinare
int len = strlen(parola_da_indovinare);

char lettera;

for(i=0;i<len;i++)
    stato_parola[i]='_'; //Inizializzo il campo da gioco con '_'
```



# Dichiarazione ed inizializzazione variabili

```
//Dizionario di parole
char dizionario[5][20]={"computer", "proiettore", "aula", "ciao", "telecomando"};

int i;
int tentativi=0;
int found=0;

//Puntatore alla parola da indovinare, scelta casualmente
char* parola_da_indovinare;
parola_da_indovinare = dizionario[scegliParola()];

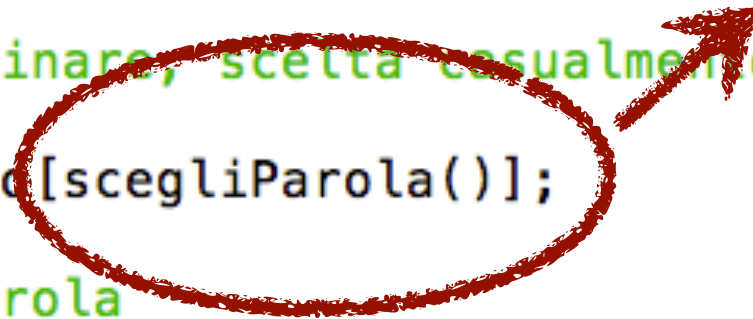
//Campo di gioco, stato della parola
char stato_parola[20];

//Lunghezza della parola da indovinare
int len = strlen(parola_da_indovinare);

char lettera;

for(i=0;i<len;i++)
    stato_parola[i]='_'; //Inizializzo il campo da gioco con '_'
```

come la implemento?





# Scegli parola

```
int scegliParola(){  
    int num;  
    srand(time(0));  
    //Estraggo un numero tra 0 e "numero di parole del dizionario"  
    num=rand()%5;  
    return num;  
}
```



# Ragioniamo sulle funzioni necessarie...

- Ora ci servono ancora alcune funzionalità:
  - Controllare se una data lettera è corretta o meno
  - Sostituire le lettere indovinate al posto dei trattini ‘\_’
  - Stampare a schermo lo stato attuale della parola
  - Controllare se il giocatore ha vinto



# Controllo e sostituzione lettera

- Ora ci servono ancora alcune funzionalità:
- Controllare se una data lettera è corretta o meno
- Sostituire le lettere indovinate al posto dei trattini ‘\_’

```
int selezionaLettere(char * parola_da_indovinare,  
                    char* stato_parola, char lettera)
```

- Controllare se il giocatore ha vinto





# Controllo e sostituzione lettera

```
int selezionaLettere(char * parola_da_indovinare,
                    char* stato_parola, char lettera)
{

    int i;
    int count=0;

    //Segno nel campo da gioco, tutte le posizioni
    //corrispondenti alla lettera scelta dall'utente
    for(i=0;i<strlen(parola_da_indovinare);i++)
        if(parola_da_indovinare[i]==lettera)
        {
            stato_parola[i]=lettera;
            count++;
        }

    if(count==0)
        return 0;
    else
        return 1;
}
```



# Controllo vincita & stampa a schermo



# Controllo vincita & stampa a schermo

```
//Controllo se la parola è stata completata
int check(char* stato_parola, int len){
    int i;
    for(i=0;i<len;i++)
        if(stato_parola[i]=='_')
            return 0;

    return 1;
}
```



# Controllo vincita & stampa a schermo

```
//Controllo se la parola è stata completata
int check(char* stato_parola, int len){
    int i;
    for(i=0;i<len;i++)
        if(stato_parola[i]=='_')
            return 0;

    return 1;
}
```

```
void stampa(char* parola, int len, int tentativi){

    int i;
    printf("Hai a disposizione ancora %d tentativi\n",
           |TENTATIVI - tentativi);
    for(i=0;i<len;i++)
        printf("%c ", parola[i]);
    printf("\n");
}
```



# Gestione del gioco



# Gestione del gioco

```
while(!found && tentativi<TENTATIVI){
//Chiedo lettere, finchè non indovina la parola o esaurisce i tentativi

    stampa(stato_parola, len, tentativi);
    printf("Inserisci una lettera\n");
    scanf("%c", &lettera);
    getchar();

    int result = selezionaLettere(parola_da_indovinare, stato_parola, lettera);
    if(result==0)
        tentativi++;
    else if(check(stato_parola, len))
        found=1;
}

stampa(stato_parola, len, tentativi);
if(found==1)
    printf("Complimenti!\n");
else
    printf("Hai superato i tentativi a disposizione\n");

return 0;
```



# **Esercizi sulle stringhe e matrici**



# Esercizio 1: stringhe - vocali consecutive

Scrivere un programma che data una stringa di lunghezza massima 100 stampi le occorrenze di coppie di vocali identiche consecutive.





# Esercizio I: stringhe - vocali 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;

}
```



## Esercizio 2: stringhe & array

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



# Esercizio 2: stringhe & array

```
#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 >= 1 && len<=l+2 && strcmp(stringa, "stop") != 0){
            printf("%d\n", len-1);
            count[len-1]++;
        }

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

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

    system("PAUSE");
    return 0;

}
```



## Esercizio 3: matrice

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.



# Esercizio 3: matrice

```
#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]);
        }
    }
}
```



# Esercizio 3: matrice

```
/** 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;  
}
```



## Esercizio 4: matrice

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



# Esercizio 4: matrice

```
#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;
}
```





## Esercizio 6: Array Lettere

Il programma acquisisce una sequenza di caratteri in input  
tale sequenza e' terminata dal carattere #  
conteggiare quante volte ogni lettera minuscola e'  
inserita



# Esercizio 6: Array Lettere

```
#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;
}
```



## Esercizio 7: Stringa Palindroma

il programma acquisisce una sequenza di caratteri di lunghezza massima 100 (terminata da #)  
determinare se la parola composta da tali caratteri (considerati nell'ordine inserito) è palindroma



# Esercizio 7: Stringa 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 termianre\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;
    }
```



## Esercizio 8: Stringa Concatenate

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



# Esercizio 8: Stringa Concatenate

```
#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;
}
```

**Tutte il materiale sar   
disponibile sul mio sito  
internet!**

[alessandronacci.it](http://alessandronacci.it)

**See You Next Time!**

