



IEIM 2017-2018

Esercitazione VIII “Puntatori, Matrici, Enum”

Alessandro A. Nacci
alessandro.nacci@polimi.it - www.alessandronacci.it



Matrici e funzioni

Esercizio I



WARNING

MATRICI E FUNZIONI

Il passaggio di una matrice ad una funzione
e' sempre per indirizzo e mai per copia.

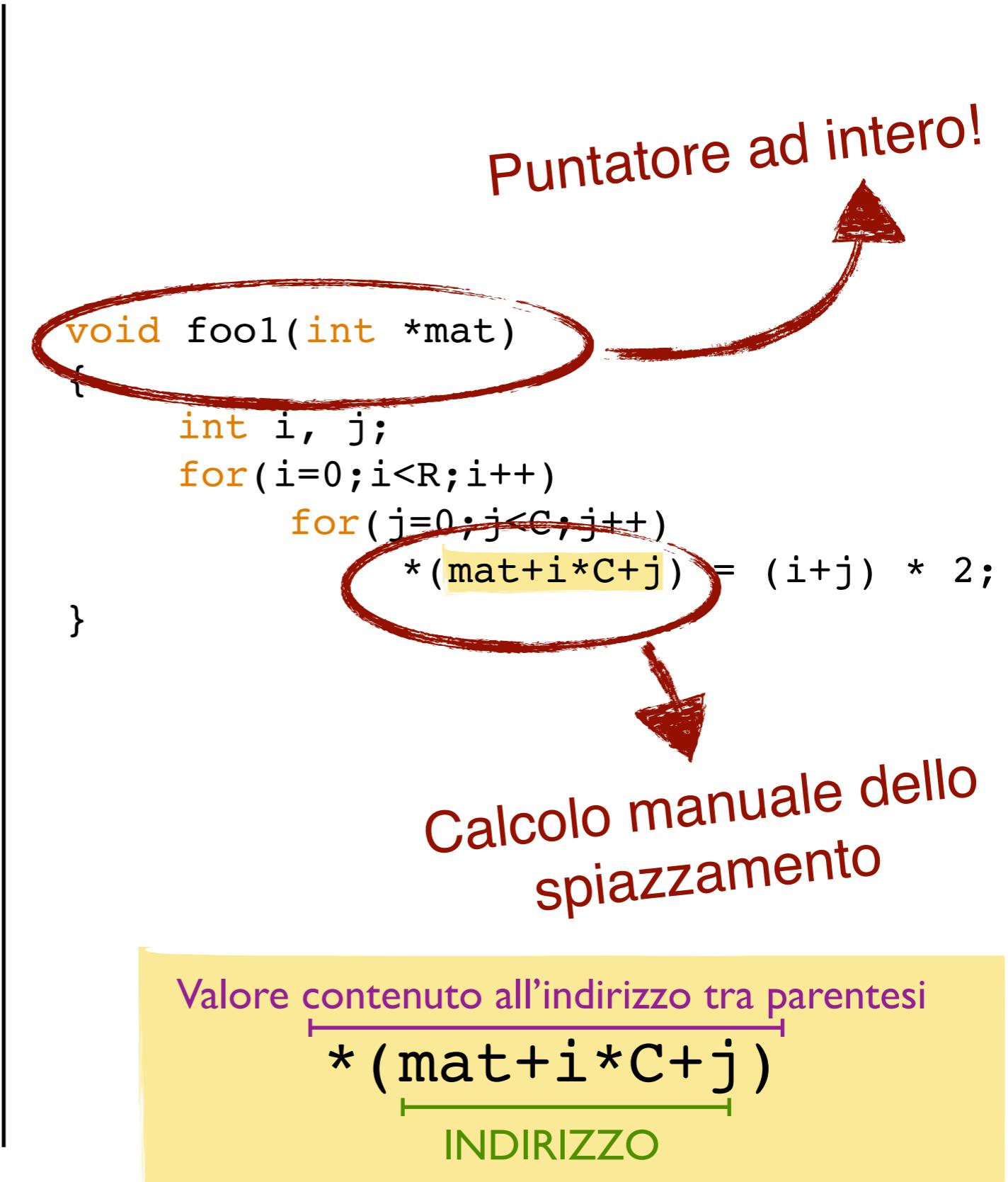
Esercizio I: Matrici e Funzioni

```
#include <stdio.h>

#define R 3
#define C 3

void foo1(int*);
void foo2(int mat[R][C]);
void foo3(int mat[][C]);

int main()
{
    int mat[R][C];
    foo1(mat);
    foo2(mat);
    foo3(mat);
}
```



Esercizio I: Matrici e Funzioni

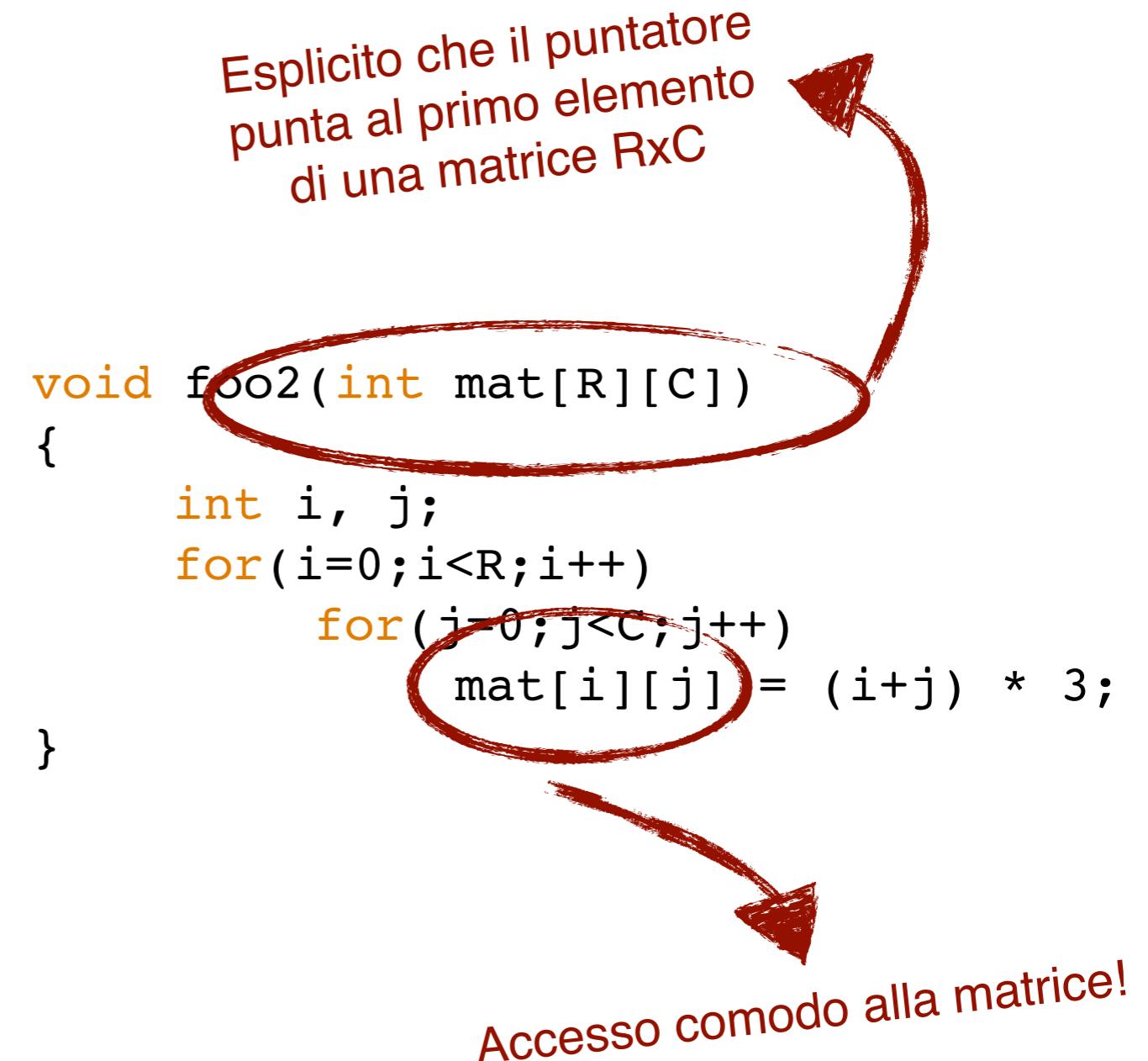
```
#include <stdio.h>

#define R 3
#define C 3

void foo1(int*);
void foo2(int mat[R][C]);
void foo3(int mat[][C]);

int main()
{
    int mat[R][C];
    foo1(mat);
    foo2(mat);
    foo3(mat);
}
```

Esplicito che il puntatore
punta al primo elemento
di una matrice RxC



```
void foo2(int mat[R][C])
{
    int i, j;
    for(i=0;i<R;i++)
        for(j=0;j<C;j++)
            mat[i][j] = (i+j) * 3;
}
```

Accesso comodo alla matrice!

Esercizio I: Matrici e Funzioni

```
#include <stdio.h>

#define R 3
#define C 3

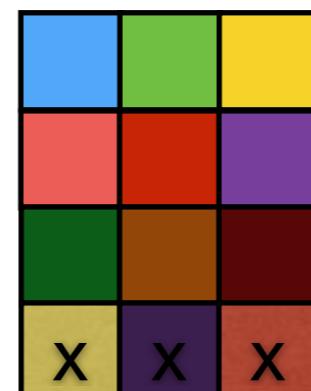
void foo1(int*);
void foo2(int mat[R][C]);
void foo3(int mat[][C]);

int main()
{
    int mat[R][C];
    foo1(mat);
    foo2(mat); → Il valore di mat cambia
    foo3(mat); → Il valore di mat cambia
}
```

Esplicito che il puntatore punta al primo elemento di una matrice con C colonne

```
void foo3(int mat[][C])
{
    int i, j;
    for(i=0;i<R;i++)
        for(j=0;j<C;j++)
            mat[i][j] = (i+j) * 4;
}
```

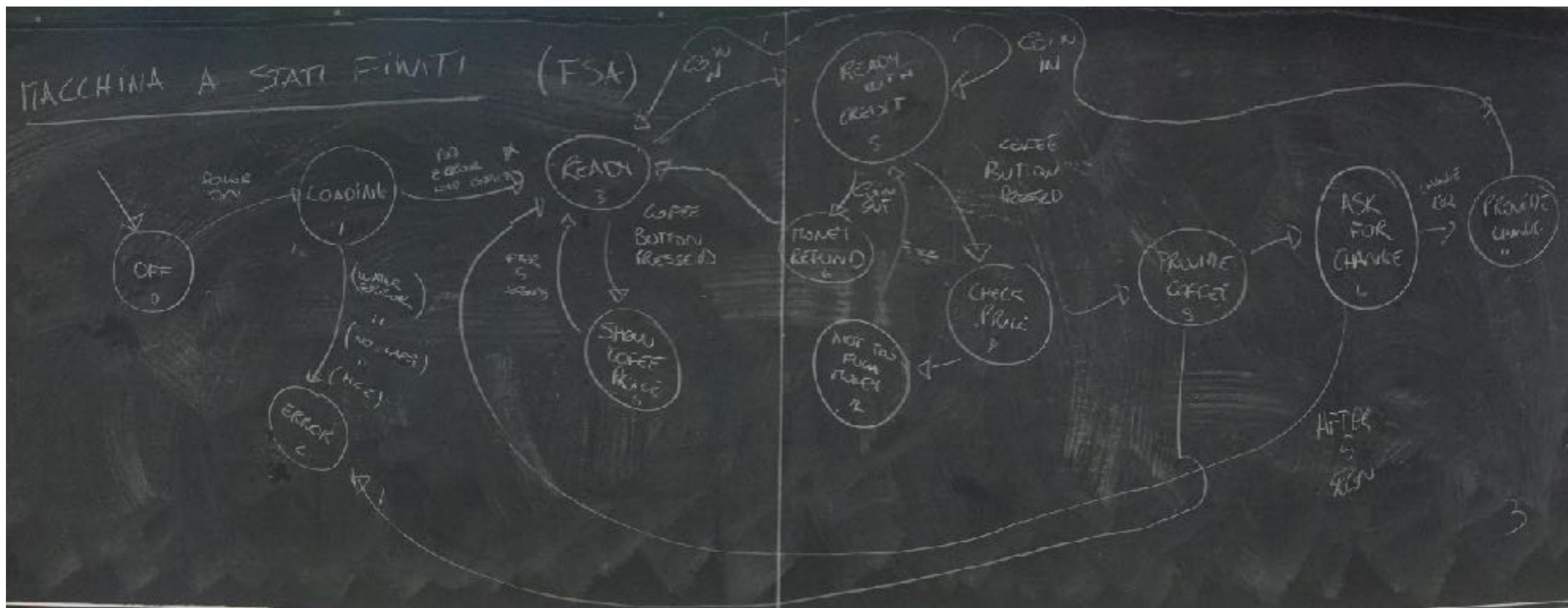
Accesso comodo alla matrice!



NOTA BENE

Ai fini del calcolo dello spiazzamento il numero di righe non è essenziale!

Macchina a Stati Finiti (Coffee Machine)





Macchina a Stati Finiti (Coffee Machine)

```
typedef enum {
    OFF,
    LOADING,
    ERROR,
    READY
    SHOW_COFFE_PRICE,
    READY_WITH_CREDIT,
    MONEY_REFUND,
    NOT_TOO MUCH MONEY,
    CHECK_PRICE,
    PROVIDE_COFFEE,
    ASK_FOR_CHANGE,
    PROVIDE_CHANGE
} stateNameType;

typedef struct
{
    stateNameType stateName;
    float floatValue;
}

stateType;
```



Macchina a Stati Finiti (Coffee Machine)

```
stateType f_off(stateType current_state);
stateType f_loading(stateType current_state);
stateType f_error(stateType current_state);
stateType f_ready(stateType current_state);
stateType f_show_coffe_price(stateType current_state);
stateType f_ready_with_credit(stateType current_state);
stateType f_money_refund(stateType current_state);
stateType f_not_too_much_money(stateType current_state);
stateType f_check_price(stateType current_state);
stateType f_provide_coffee(stateType current_state);
stateType f_ask_for_change(stateType current_state);
stateType f_provide_change(stateType current_state);
```



Macchina a Stati Finiti (Coffee Machine)

```
int main()
{
    stateType current_state;
    stateType next_state;

    current_state.stateName = OFF;
    nextState.stateName = OFF;

    while(1)
    {
        switch(current_state.stateName)
        {
            case OFF:
                next_state = f_off(current_state);
                break;
            case LOADING:
                next_state = f_loading(current_state);
                break;
            case ERROR:
                next_state = f_error(current_state);
                break;
            case READY:
                next_state = f_ready(current_state);
                break;
            case SHOW_COFFE_PRICE:
                next_state = f_show_coffe_price(current_state);
                break;
            case READY_WITH_CREDIT:
                next_state = f_ready_with_credit(current_state);
                break;
            case MONEY_REFUND:
                next_state = f_money_refund(current_state);
                break;
            case NOT_TOO MUCH MONEY:
                next_state = f_not_too_much_money(current_state);
                break;
            case CHECK PRICE:
                next_state = f_check_price(current_state);
                break;
            case PROVIDE_COFFEE:
                next_state = f_provide_coffee(current_state);
                break;
            case ASK_FOR_CHANGE:
                next_state = f_ask_for_change(current_state);
                break;
            case PROVIDE_CHANGE:
                next_state = f_provide_change(current_state);
                break;
            default:
                printf("error");
        }

        current_state = next_state;
    }
}
```



Macchina a Stati Finiti (Coffee Machine)

```
stateType f_ready(stateType current_state)
{
    stateType nextState;
    char value[10];
    float money;

    scanf ("%s", value);

    if (strcmp(value, "COFFEE") == 0)
    {
        nextState.stateName = SHOW_COFFE_PRICE;
    }
    else if (strcmp(value, "THE") == 0)
    {

    }
    else
    {
        money = string2float(value);
        nextState.stateName = READY_WITH_CREDIT;
        nextState.floatValue = money;
    }

    return nextState;
}
```



Macchina a Stati Finiti (Coffee Machine)

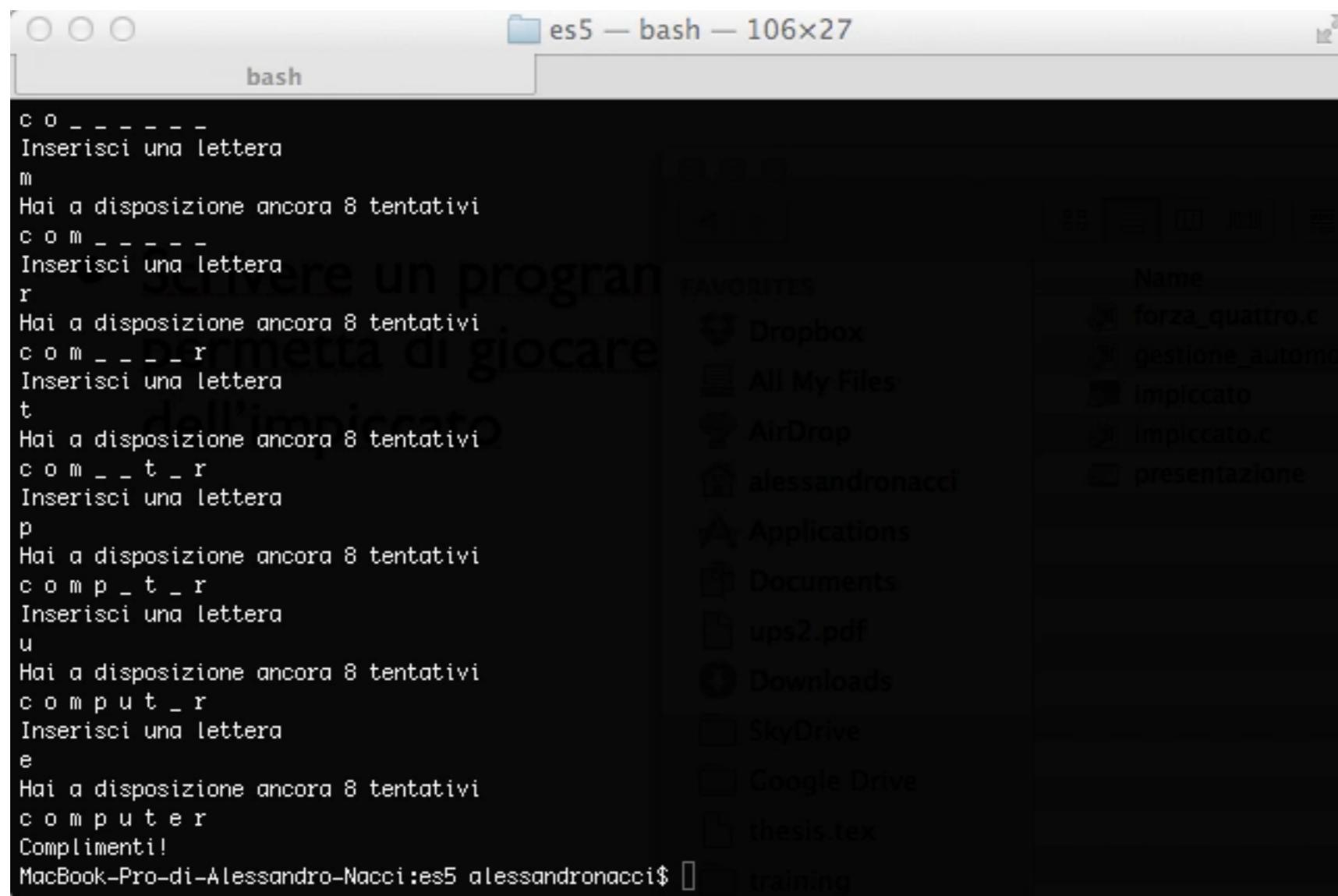
```
stateType f_show_coffe_price(stateType current_state)
{
    printf("Credito %f", current_state.floatValue);
    stateType nextState;
    nextState.stateName = READY;
    wait(5000);
    return nextState;
}
```

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



```
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$
```



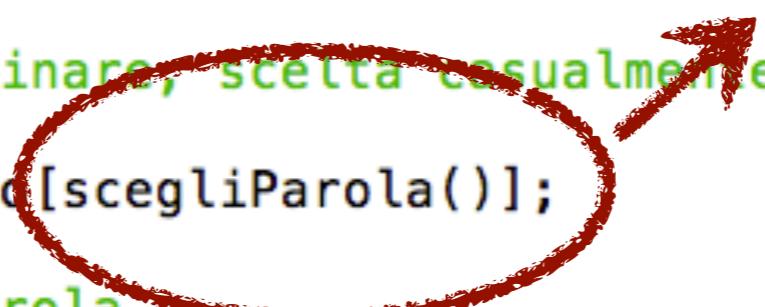
Vediamo cosa ci serve...

- 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 '_'
```

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

```
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 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

```
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_inovinare, 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;
```

**Tutte il materiale sarà
disponibile sul mio sito
internet!**

alessandronacci.it

See You Next Time!

