



IEIM 2015-2016

Esercitazione XI

“Funzioni, Punatori, Array e Ricorsione”

Alessandro A. Nacci

alessandro.nacci@polimi.it - www.alessandronacci.it

Tratto da:
Lab 7: 10 Giugno 2013
Gianluca Durelli – durelli@elet.polimi.it
Marco D. Santambrogio – marco.santambrogio@polimi.it
Ver. aggiornata al 09 Giugno 2013



- Mercato
- Tris 1
- Tris 2
- Ricorsione



Es I - Mercato

- Si scriva un programma per la gestione del carrello di un supermercato.
- Si definisca una struttura dati per rappresentare un prodotto acquistabile; tale struttura deve contenere:
 - Codice prodotto
 - Nome
 - Prezzo
- Si definisca una struttura dati per rappresentare il carrello:
 - Lista di prodotti
 - Numero di prodotti acquistati
 - Totale da pagare
- Viene chiesto all'utente di inserire i prodotti che devono essere acquistati. Tali prodotti vengono inseriti nel carrello e le informazioni del carrello vengono aggiornate. Alla fine si procede alla stampa dello scontrino e del totale da pagare.
- Vincolo:
 - Si strutturi il programma dividendolo in sottofunzioni ove possibile. In particolare si creino apposite funzioni per:
 - Leggere un prodotto acquistato
 - Inserire il prodotto nel carrello
 - Calcolare il totale da pagare
 - Stampare lo scontrino



Es I - Mercato - Strutture Dati

```
#include <stdio.h>

#define MAX_PRODOTTI 10
#define MAX_NOME 50

typedef struct{
    int codice;
    char nome[MAX_NOME];
    float prezzo;
} prodotto;

typedef struct{
    prodotto prodotti[MAX_PRODOTTI];
    int numeroProdotti;
    float totale;
} carrello;
```



Es I - Mercato - Prototipi Funzioni

```
prodotto leggiProdotto();  
void inserisciProdotto(carrello *c, prodotto p);  
void calcolaTotale(carrello *c);  
void stampaScontrino(carrello *c);
```



Es I - Mercato - Funzioni

```
prodotto leggiProdotto() {  
    prodotto p;  
  
    printf( "Codice: " );  
    scanf( "%d", &p.codice );  
    printf( "Nome: " );  
    scanf( "%s", p.nome );  
    printf( "Prezzo: " );  
    scanf( "%f", &p.prezzo );  
  
    return p;  
}
```



Es I - Mercato - Funzioni

```
void inserisciProdotto(carrello *c, prodotto p){  
    c->prodotti[c->numeroProdotti]=p;  
    c->numeroProdotti+=1;  
}
```



Es I - Mercato - Funzioni

```
void calcolaTotale(carrello *c){  
    int i;  
  
    c->totale=0;  
  
    for(i=0;i<c->numeroProdotti;i++)  
        c->totale+=c->prodotti[i].prezzo;  
}
```



Es I - Mercato - Funzioni

```
void stampaScontrino(carrello *c){
    int i;

    for(i=0;i<c->numeroProdotti;i++)
        printf("%d\t%s\t%f\n",c->prodotti[i].codice, c->prodotti[i].nome, c->prodotti[i].prezzo);

    printf("\n\tTOTALE:\t%f\n",c->totale);
}
```



Es I - Mercato - Main

```
int main(){
    carrello c;
    prodotto p;

    c.numeroProdotti=0;

    int i;
    char s;

    do{
        printf("Vuoi comprare un prodotto (s/n)? ");
        scanf(" %c",&s);

        if(s=='s' || s=='S'){
            p=leggiProdotto();
            inserisciProdotto(&c,p);
        }
    }while((s=='s' || s=='S') && c.numeroProdotti<MAX_PRODOTTI);

    calcolaTotale(&c);
    stampaScontrino(&c);

    return 0;
}
```



Es 2 - Tris

- Si scriva un programma che controlli la possibilità di vincere una partita a tris.
- Il programma chiede all'utente di inserire una matrice 3x3 riempita con i numeri 0,1,2:
 - 0 significa casella vuota
 - 1 è la pedina del primo giocatore
 - 2 è la pedina del secondo giocatore
- Viene chiesto per quale giocatore si vuole effettuare il controllo: 1 o 2.
- Si effettui la mossa per far vincere il giocatore scelto nel caso questa sia possibile.

- Vincolo:
 - Si strutturi il programma dividendolo in funzioni ove possibile. In particolare si creino apposite funzioni per:
 - Riempire la matrice
 - Controllare se è possibile fare tris, e nel caso fare la mossa vincente.
 - Stampare la matrice

```
AlessandroMBP2:Lab7-Lun10Giugno2013 alessandronacci$ ./IEIM_Exe2_Lab7
Inserisci il valore (0/1/2) in posizione 0-0: 0
Inserisci il valore (0/1/2) in posizione 0-1: 0
Inserisci il valore (0/1/2) in posizione 0-2: 1
Inserisci il valore (0/1/2) in posizione 1-0: 2
Inserisci il valore (0/1/2) in posizione 1-1: 1
Inserisci il valore (0/1/2) in posizione 1-2: 0
Inserisci il valore (0/1/2) in posizione 2-0: 2
Inserisci il valore (0/1/2) in posizione 2-1: 1
Inserisci il valore (0/1/2) in posizione 2-2: 2
0 0 1
2 1 0
2 1 2
Quale giocatore vuoi controllare? 1
Il giocatore ha vinto
0 1 1
2 1 0
2 1 2
```



Es 2 - Tris - Prototipi Funzione

```
#include <stdio.h>
```

```
void riempiMatrice(char mat[3][3]);
```

```
char checkGiocatore(char mat[3][3], char player);
```

```
void faiMossa(char mat[3][3], char x, char y, char player);
```

```
void stampaMatrice(char mat[3][3]);
```



Es 2 - Tris - Funzioni

```
void riempiMatrice(char mat[3][3]){
    int i,j;
    int n;

    for(i=0;i<3;i++)
        for(j=0;j<3;j++){
            printf("Inserisci il valore (0/1/2) in posizione %d-%d: ",i,j);
            scanf("%d", &n);
            mat[i][j]=n;
        }
}
```



Es 2 - Tris - Funzioni

```
char checkGiocatore(char mat[3][3], char player){
    char riga=0;
    char colonna=0;
    char d1=0;
    char d2=0;

    int i,j;

    for(i=0;i<3;i++){
        riga=0;
        colonna=0;
        for(j=0;j<3;j++){
            if(mat[i][j]==player)
                riga+=1;

            if(mat[j][i]==player)
                colonna+=1;

            if(mat[i][j]==2-player+1)
                riga=-1;

            if(mat[j][i]==2-player+1)
                colonna=-1;
        }

        if(riga==2){
            for(j=0;j<3;j++)
                faiMossa(mat,i,j,player);
            return 1;
        }
        if(colonna==2){
            for(j=0;j<3;j++)
                faiMossa(mat,j,i,player);
            return 1;
        }
    }
}
```

```
for(i=0;i<3;i++){
    if(mat[i][i]==player)
        d1+=1;
    if(mat[i][2-i]==player)
        d2+=1;

    if(mat[i][i]==3-player)
        d1=-1;

    if(mat[i][2-i]==3-player)
        d2=-1;
}

if(d1==2){
    for(j=0;j<3;j++)
        faiMossa(mat,j,j,player);
    return 1;
}

if(d2==2){
    for(j=0;j<3;j++)
        faiMossa(mat,j,2-j,player);
    return 1;
}

return 0;
}
```



Es 2 - Tris - Funzioni

```
void faiMossa(char mat[3][3], char x, char y, char player){  
    mat[x][y]=player;  
}
```



Es 2 - Tris - Funzioni

```
void stampaMatrice(char mat[3][3]) {  
    int i, j;  
  
    for(i=0; i<3; i++) {  
        for(j=0; j<3; j++)  
            printf("%d ", mat[i][j]);  
        printf("\n");  
    }  
}
```



Es 2 - Tris - Main

```
int main(){
    char mat[3][3];
    int p;
    char check;

    riempiMatrice(mat);
    stampaMatrice(mat);

    printf("Quale giocatore vuoi controllare? ");
    scanf("%d", &p);

    check=checkGiocatore(mat, p);

    if(check==1){
        printf("Il giocatore ha vinto\n");
        stampaMatrice(mat);
    }
    else
        printf("Il giocatore non puo' vincere\n");

    return 0;
}
```



Es 3 - Tris (v2)

- Si scriva un programma che controlli la possibilità di perdere una partita a tris alla mossa successiva.
- Il programma chiede all'utente di inserire una matrice 3x3 riempita con i numeri 0,1,2:
 - 0 significa casella vuota
 - 1 è la pedina del primo giocatore
 - 2 è la pedina del secondo giocatore
- Viene chiesto per quale giocatore si vuole effettuare il controllo: 1 o 2.
- Il giocatore perde la partita alla mossa successiva se valgono simultaneamente le seguenti condizioni:
 - Non può fare tris con la sua mossa
 - L'avversario può fare più di un tris
- Si stampi a video il risultato del controllo
- Vincolo:
 - Si strutturi il programma dividendolo in funzioni ove possibile. In particolare si creino funzioni per:
 - Riempire la matrice
 - Controllare se si perderà la partita.
 - Stampare la matrice
- Facoltativo:
 - Si effettui una contromossa per non perdere la partita:
 - Fare tris
 - Impedire un tris dell'avversario (nel caso ne possa fare solo 1)



Es 2 - Tris - Prototipi Funzioni

```
#include <stdio.h>
```

```
void riempiMatrice(char mat[3][3]);
```

```
char checkGiocatore(char mat[3][3], char player, char curPlayer);
```

```
void faiMossa(char mat[3][3], char x, char y, char player);
```

```
void stampaMatrice(char mat[3][3]);
```

```
char checkVittoria(char mat[3][3], char player);
```



Es 2 - Tris - Funzioni

```
char checkVittoria(char mat[3][3], char player){
    char temp[3][3];
    int i,j;
    int count=0;
    int otherPlayer=3-player;

    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            temp[i][j]=mat[i][j];

    if(checkGiocatore(temp,player, player)==1){
        checkGiocatore(mat,player, player);
        return 0;
    }

    while(checkGiocatore(temp,otherPlayer, player)==1){
        //stampaMatrice(temp);
        count++;
    }

    if(count>1)
        return 1;

    if(count==1)
        checkGiocatore(mat,otherPlayer, player);

    return 0;
}
```



Es 2 - Tris - Main

```
int main(){
    char mat[3][3];
    int p,q;
    char check;

    riempiMatrice(mat);
    stampaMatrice(mat);

    printf("Quale giocatore vuoi controllare? ");
    scanf("%d",&p);
    q=3-p;

    check=checkVittoria(mat, p);

    if(check==1){
        printf("Il giocatore %d vincerà in due mosse\n", q);
        stampaMatrice(mat);
    }
    else
        printf("Il giocatore %d non può vincere in due mosse\n", q);

    return 0;
}
```



Es 4 - Ricorsione

- Si scriva un programma che, utilizzando funzioni ricorsive:
 - calcoli il numero di cifre di un numero N letto da tastiera.
 - Stampi il numero al contrario.
- Es.:
 - 1234
 - 4 cifre
 - 4321



Es 2 - Tris - Prototipi Funzioni

```
#include <stdio.h>
```

```
int contaCifre(int n);
```

```
void stampaCifre(int n);
```



Es 2 - Tris - Funzioni

```
int contaCifre(int n) {  
    if (n==0)  
        return 0;  
    else  
        return 1+contaCifre(n/10);  
}
```



Es 2 - Tris - Funzioni

```
void stampaCifre(int n) {  
    if (n==0)  
        printf(" ");  
    else {  
        printf("%d", n%10);  
        stampaCifre(n/10);  
    }  
}
```



Es 2 - Tris - Main

```
int main(){
    int n;

    printf("Inserisci un numero: ");
    scanf("%d",&n);

    printf("Il numero %d ha %d cifre\n",n,contaCifre(n));
    stampaCifre(n);
    printf("\n");

    return 0;
}
```

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

alessandronacci.it

See You Next Time!

