



IEIM 2017-2018

Esercitazione XI ***“Ricorsione e Ripasso”***

Alessandro A. Nacci

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

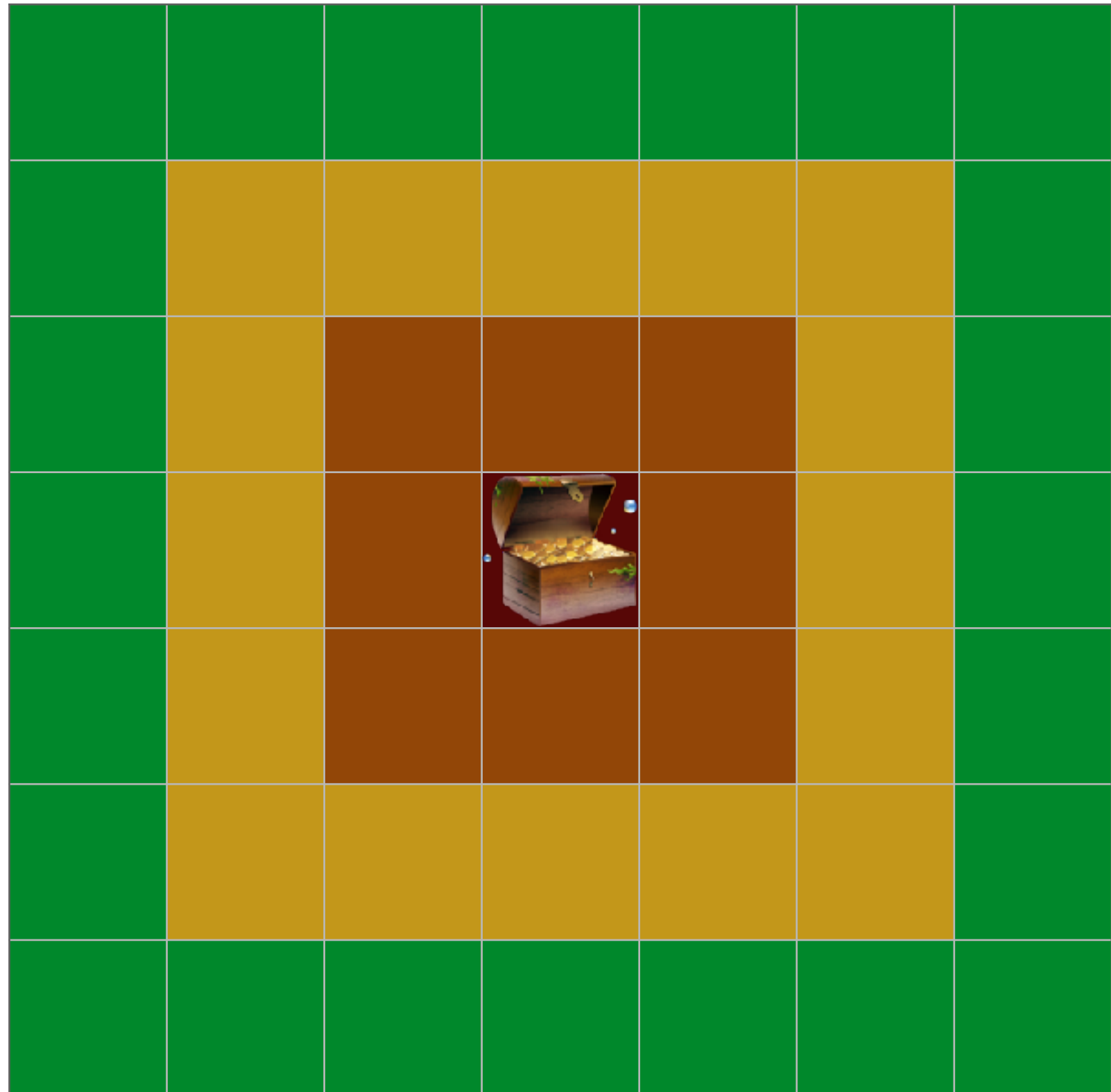
Mappa del Tesoro

Matrici e Ricorsione

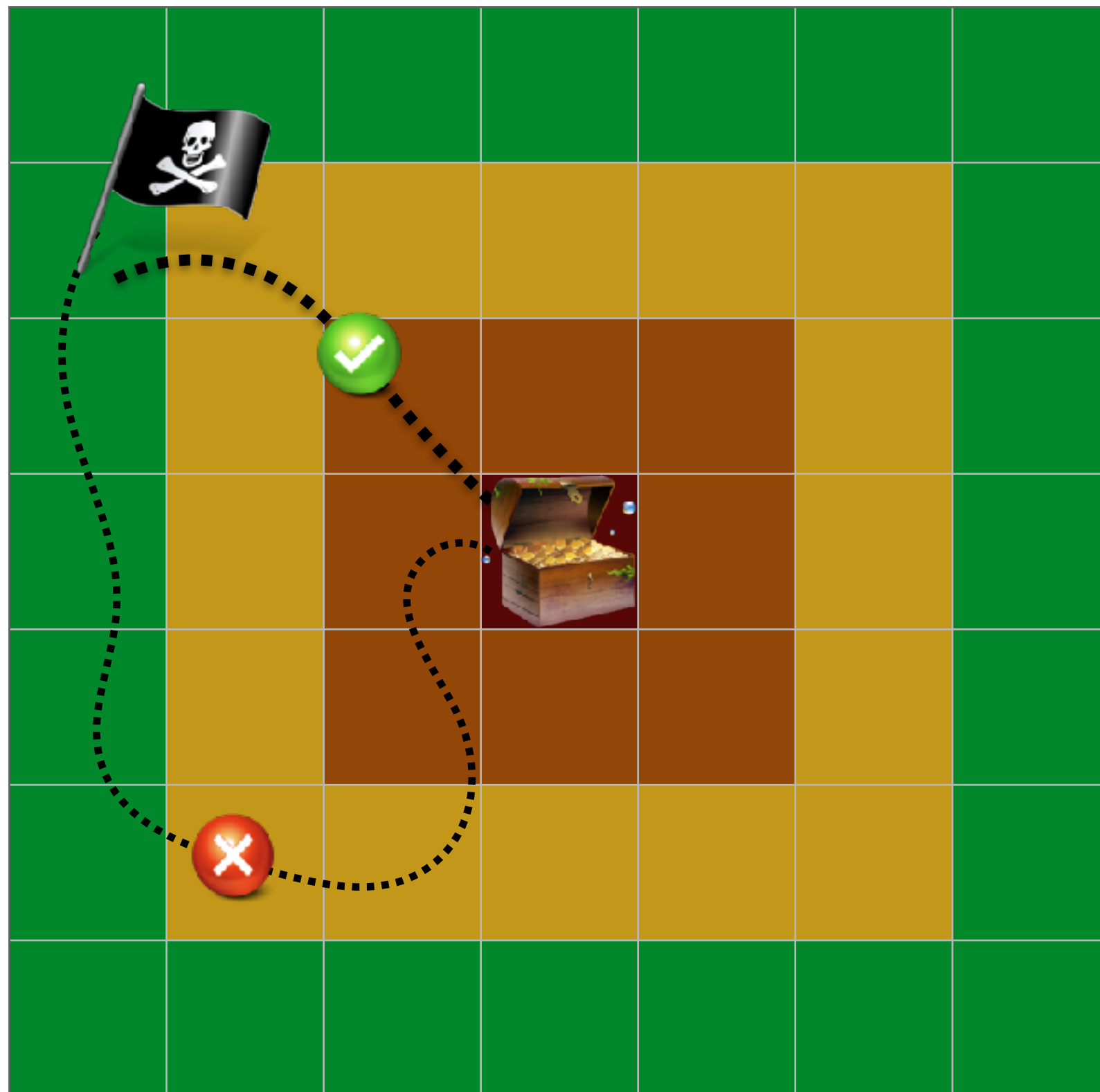
LA MAPPA DEL TESORO



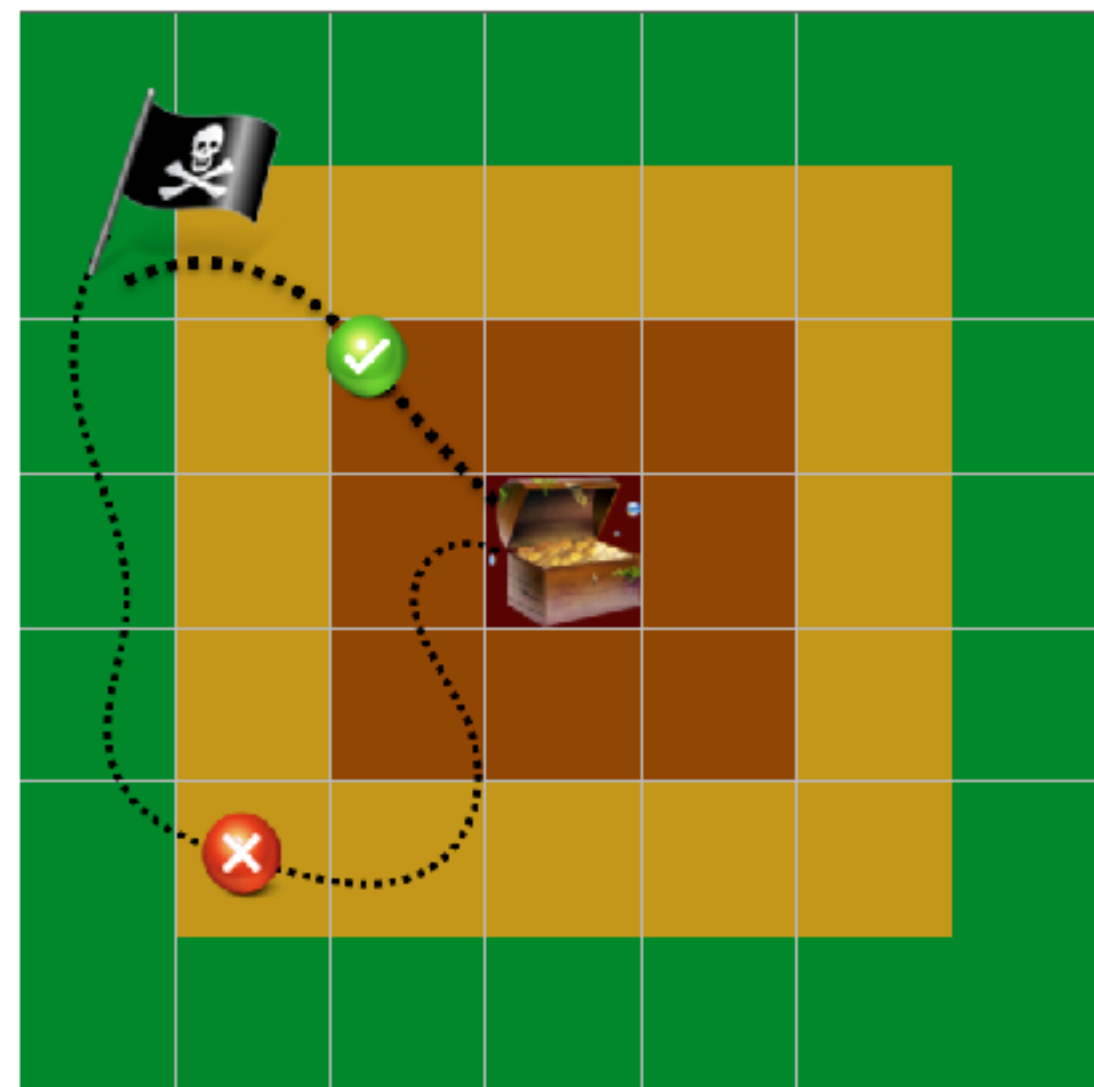
- Immaginiamo di avere una mappa del tesoro... un po particolare!
- E' una sorta di mappa termica **radiale** della “vicinanza” al tesoro



- Un pirata che vuole muoversi su questa mappa per cercare il tesoro, si sposterà **sempre** dalle celle con colori più freddi verso celle con colori più caldi



- Scrivere un programma C che sia in grado di
- gestire la mappa appena introdotta
- date le coordinate del tesoro, creare la *mappa termica di vicinanza*
- date delle coordinate di partenza, tracciare un percorso valido per arrivare al tesoro



- Come rappresentiamo la mappa termica?
 - Usiamo una matrice
 - I colori diventano numeri

11	11	11	11	11	11	11	11	11
11	12	12	12	12	12	12	12	11
11	12	13	13	13	13	13	12	11
11	12	13	14	14	14	13	12	11
11	12	13	14	15	14	13	12	11
11	12	13	14	14	14	13	12	11
11	12	13	13	13	13	13	12	11
11	12	12	12	12	12	12	12	11
11	11	11	11	11	11	11	11	11



Include, define e dichiarazione variabili

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

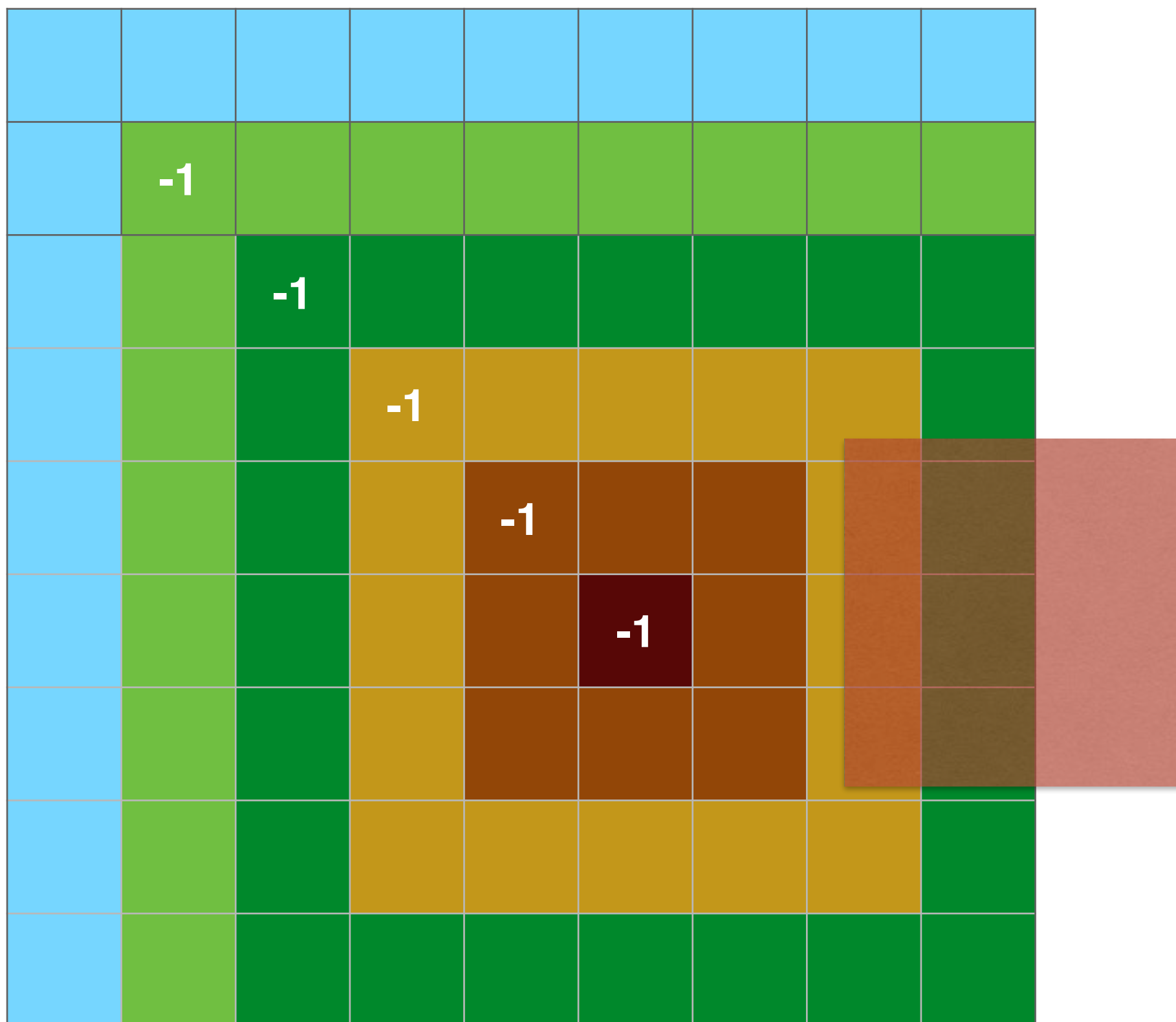
```
#define W 10
#define H 10
```

```
#define VALORE_TESORO 15
```

**VALORE_TESORO è legato alla
dimensione della mappa**



```
int main( )
{
    int mappa[W][H];
    int x,y;
    int trovato = 0;
```



Inizializziamo la mappa

```
void init_mappa(int mappa[W][H])  
{
```

Scrivere una funzione che riempia di '0' una matrice 'mappa' in ingresso.

```
}
```



Stampiamo la mappa

```
void stampa_mappa(int mappa[W][H])  
{
```

*Scrivere una funzione che visualizzi a
schermo una matrice 'mappa' in
ingresso*

```
}
```



Metti tesoro - corretto

```
void metti_tesoro(int mappa[W][H], int tes_x, int tes_y)
{
```

Scrivere una funzione che, date due coordinate tes_x e tes_y - che sono le coordinate del tesoro - crei la mappa termica radiale presentata in precedenza all'interno di una matrice 'mappa' passata in ingresso.

```
}
```



Cerchiamo il tesoro e tracciamo il percorso...

```
int cerca_tesoro(int mappa[W][H], int start_x, int start_y)
{
```

*Scrivere una funzione C che date due coordinate da cui il pirata parte ('start_x' e 'start_y'), cerchi un percorso **corretto** per arrivare al tesoro.*

Per creare un percorso corretto, dato un punto generico in cui il pirata si trova, il pirata può spostarsi solo in una cella adiacente che abbia un valore superiore a quello della cella corrente.

Una volta che una cella viene visitata, è necessario 'marcarla' per indicare che quella cella è parte del percorso scelto.

```
}
```



Mostriamo a schermo il percorso...

```
void stampa_percorso(int mappa[W][H])  
{
```

Scrivere una cella che visualizzi in modo chiaro quale è il percorso scelto dal pirata per raggiungere il tesoro.

```
}
```



E facciamo il main :)

```
int main()  
{  
    int mappa[W][H];  
    int x,y;  
    int trovato = 0;  
  
    init_mappa(mappa);  
    stampa_mappa(mappa);  
  
    metti_tesoro(mappa, 5,5);  
    stampa_mappa(mappa);  
  
    trovato = cerca_tesoro(mappa, 0,2);  
  
    if (trovato) printf("Ho trovato il tesoro!\n");  
  
    stampa_mappa(mappa);  
    stampa_percorso(mappa);  
}
```

```
- - - - -  
- # - - - - -  
- - # - - - - -  
- - - # - - - - -  
- - - - # - - - - -  
- - - - - - - - - -  
- - - - - - - - - -  
- - - - - - - - - -  
- - - - - - - - - -  
- - - - - - - - - -  
- - - - - - - - - -
```




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;
}
```

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

alessandronacci.it

See You Next Time!

