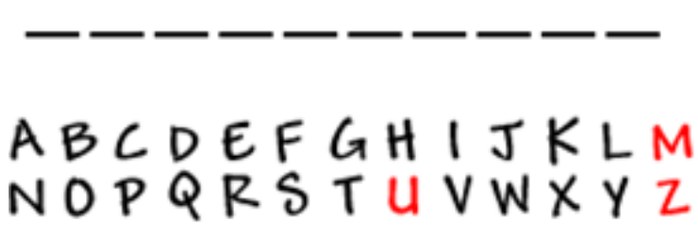
IEIM 2015-2016

Esercitazione VIII
"Gioco dell'impiccato e Mappa Del Tesoro"

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

Il gioco dell'impiccato

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



es5 — bash — 106×27

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

```c
//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 '_'
  - Controllare se il giocatore ha vinto

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

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

```c
//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

```c
//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;
}
```

```c
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;
```

Mappa del Tesoro

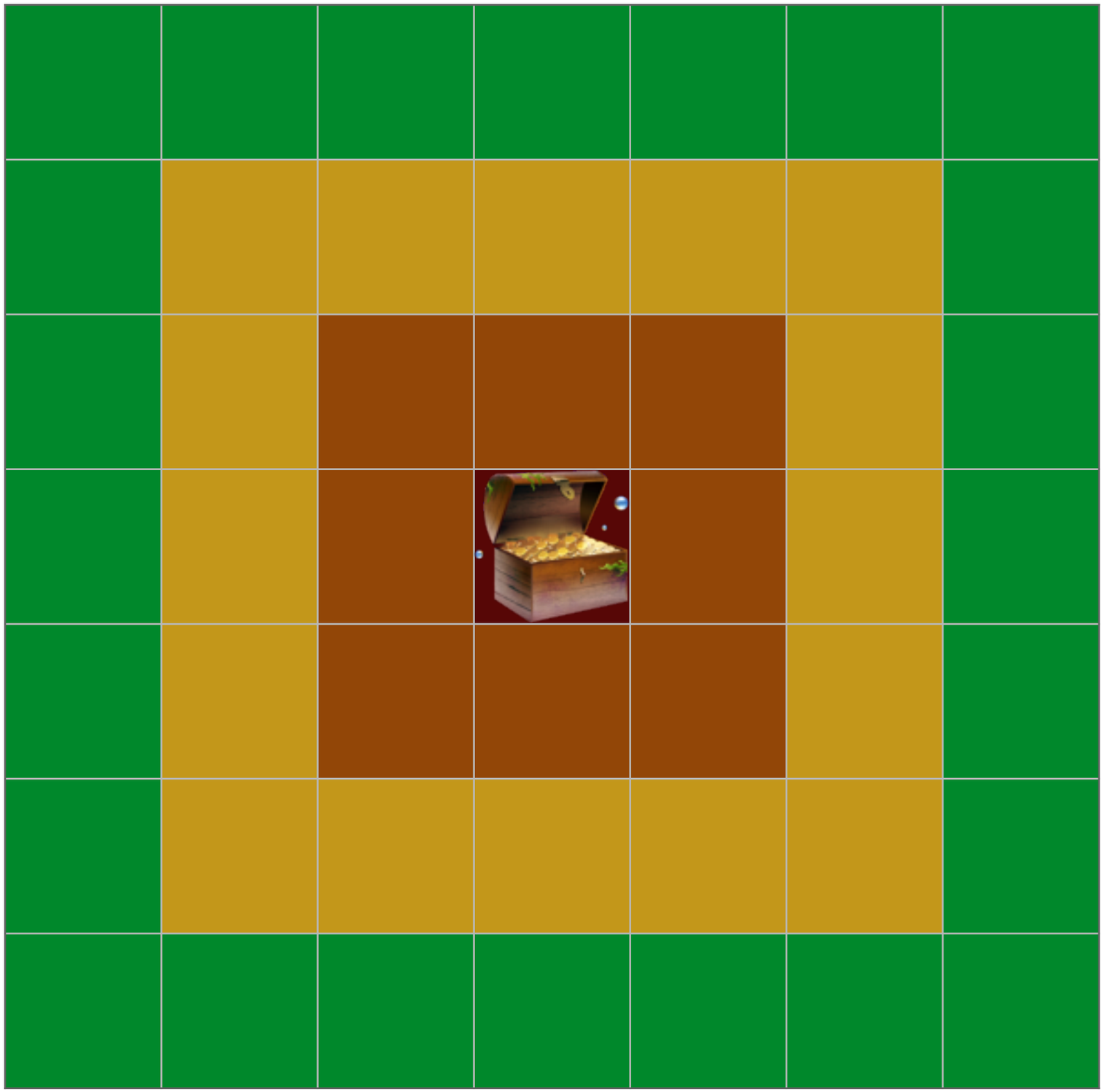
Matrici e Ricorsione

LA MAPPA DEL TESORO

Una mappa particolare!

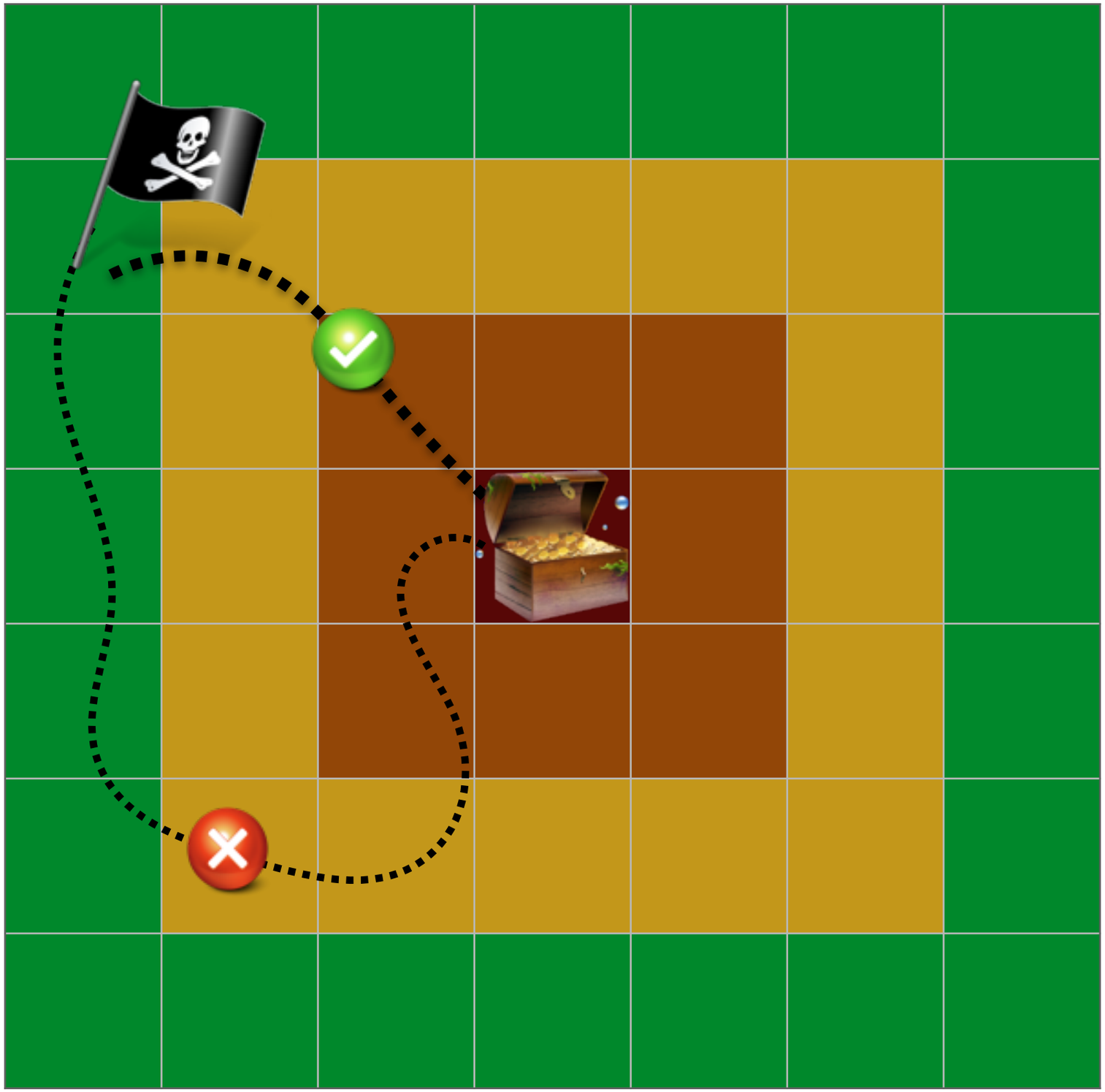
- Immaginiamo di avere una mappa del tesoro… un po particolare!

- E’ una sorta di mappa termica **radiale** della “vicinanza” al tesoro



La strategia del pirata

- 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



Traccia dell'esercizio

- 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

Gestione della mappa termica

- Come rappresentiamo la mappa termica?

Gestione della mappa termica

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

Gestione della mappa termica

- 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 funziona che riempia di '0' una matrice 'mappa' in ingresso.

```
}
```

Inizializziamo la mappa

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

      int x,y;

    for (x = 0; x < W; x++)
      {
          for (y = 0; y < H; y++)
          {
              mappa[x][y] = 0;
          }
      }

}
```

Stampiamo la mappa

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

}
```

Scrivere una funziona che visualizzi a schermo una matrice 'mappa' in ingresso

Stampiamo la mappa

```
void stampa_mappa(int mappa[W][H])
{
      int x,y;

      for (x = 0; x < W; x++)
      {
            for (y = 0; y < H; y++)
            {
                  printf("%d\t", mappa[x][y]);
            }
            printf("\n");
      }

      printf("\n\n\n");

}
```

Metti tesoro - corretto

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

Scrivere una funziona 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.

```
}
```

Metti tesoro - corretto

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

    int radius;
    int x,y;
    int start_x, start_y;
    int end_x, end_y;

    int val_tesoro = VALORE_TESORO;

    mappa[tes_x][tes_y] = val_tesoro;

    // attenzione alla dimensione massima del raggio!
    for (radius = 1; radius < W; radius++)
    {
        start_x = tes_x - radius;
        start_y = tes_y - radius;

        end_x = tes_x + radius;
        end_y = tes_y + radius;

        for (x = start_x; x <= end_x; x++)
            for (y = start_y; y <= end_y; y++)
                if ( ((x == start_x) || (x == end_x)) || ((y == start_y) || (y == end_y)))
                    if (x >= 0 && y >= 0 && x < W && y < H )
                        mappa[x][y] = val_tesoro - radius;
    }
}
```

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.

```
}
```

Cerchiamo il tesoro e tracciamo il percorso…

```
int cerca_tesoro(int mappa[W][H], int start_x, int start_y)
{
    if (mappa[start_x][start_y] == VALORE_TESORO) return 1;

    int x,y;

    for (x = start_x - 1; x <= start_x + 1; x++)
        for (y = start_y - 1; y <= start_y + 1; y++)
            if (x>0 && y >= 0 && x<W && y<H)
                if (mappa[x][y] > mappa[start_x][start_y])
                {
                    mappa[start_x][start_y] = -1;
                    return cerca_tesoro(mappa, x, y);
                }

    return 0;
}
```

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.

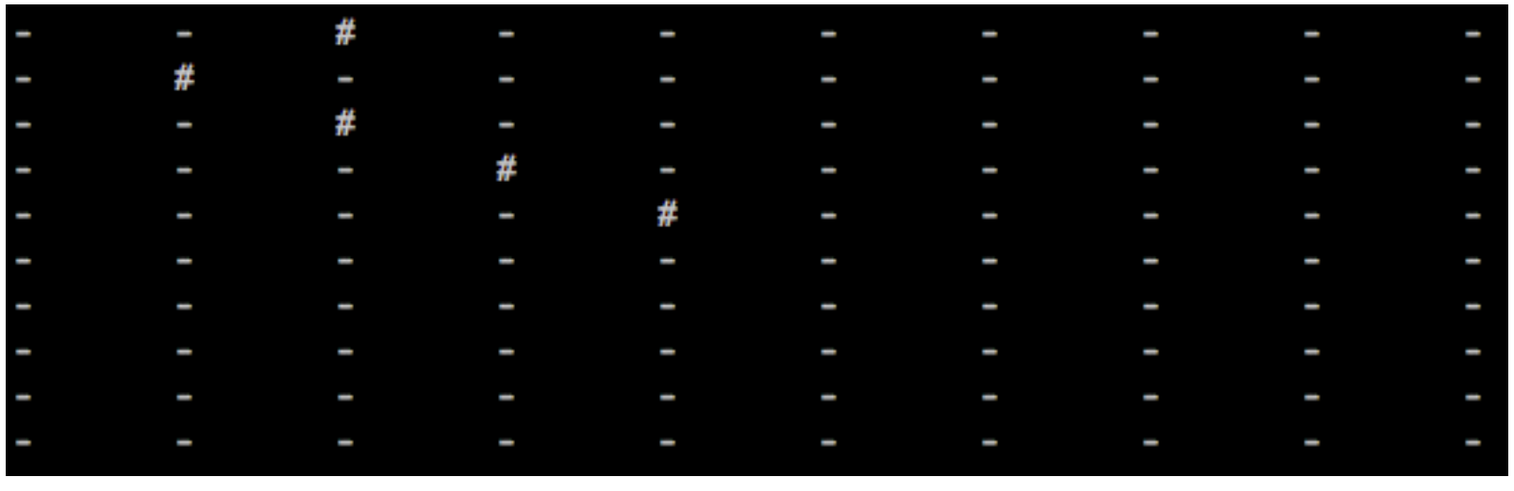
```
}
```

Mostriamo a schermo il percorso…

```
void stampa_percorso(int mappa[W][H])
{
     int x,y;

     for (x = 0; x < W; x++)
     {
          for (y = 0; y < H; y++)
          {
               if (mappa[x][y] == -1) printf("#\t");
               else printf("-\t");
          }
          printf("\n");
     }

     printf("\n\n\n");

}
```

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);

}
```



See You Next Time!

Tutte il materiale sarà disponibile sul mio sito internet!

alessandronacci.com