ALBERO BINARIO

#include <stdio.h>

#include <stdlib.h>

// Tipo info dell'albero

typedef char TipoInfoAlbero;

// Valore di errore

TipoInfoAlbero ERRORE\_InfoAlbBin='\*';

// Albero con strutture collegate

typedef struct StructAlbero {

TipoInfoAlbero info;

struct StructAlbero\* destro;

struct StructAlbero\* sinistro;

} TipoNodoAlbero;

typedef TipoNodoAlbero\* TipoAlbero;

ALBERO N-ARIO

#pragma once

#include <stdio.h>  
#include <stdlib.h>  
  
#define ERRORE\_InfoAlbero -99999  
typedef int TipoInfoAlbero;  
  
struct StructNodoFiglio;  
  
// Albero n-ario  
struct StructAlbero {  
  TipoInfoAlbero info;  
  struct StructNodoFiglio\* figli;  
};  
  
// SCL di alberi  
struct StructNodoFiglio {  
  struct StructAlbero\* albero;  
  struct StructNodoFiglio\* next;  
};  
  
typedef struct StructAlbero\* TipoAlbero;

typedef struct StructNodoFiglio\* TipoFigli;

GRAFO

#pragma once  
  
// Rappresentazione di grafi mediante doppia lista collegata.  
// StructNodo contiene informazione del grafo e puntatore a lista dei successori  
// StructNodoSucc contiene il puntatore al nodo successore e il puntatore  
// alla parte restante della lista di successori  
  
// Questa rappresentazione è efficiente in termini di uso di memoria e  
// di tempo di esecuzione delle operazioni.  
  
#define ERRORE\_InfoGrafo -99999  
typedef int TipoInfoGrafo;  
  
struct StructNodo;  
  
struct StructNodo {  
  TipoInfoGrafo info;  
  struct StructNodoSucc\* succ;  
};  
  
struct StructNodoSucc {  
  struct StructNodo\* nodo;  
  struct StructNodoSucc\* next;  
};  
  
typedef struct StructNodo\* TipoGrafo;  
typedef struct StructNodoSucc\* TipoSuccessori;