Domino - Tiago Martins 24440

Tiago Martins
Docente – Rui Moreira
Versão 1.0
Domingo, 27 de Novembro de 2016

Índice dos ficheiros

Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

Documentação do ficheiro

Referência ao ficheiro main.c

```
#include "projeto.h"
```

Funções

- int **main** (int argc, char *argv[])
  Função main.

Documentação das funções

int main (int *argc*, char * *argv*[])

Função main.

função que contém a funcao que chama o **main()** do meu projeto

Parâmetros:

int	argc não está a ser usado mas contém o numero de posicoes do array de strings argv[] usadas,
char	*argv[] não está a ser usado, mas contém o numero de posições do array de strings

Retorna:

0

Referência ao ficheiro projeto.c

```
#include "projeto.h"
```

Funções

- int **main_domino** (int argc, char *argv[])
  Função main.
- int **verificasequencia** (char **seq**[])
  Função verificasequencia.
- int * **eliminarep** (int *pv, int *psize)
  Função eliminarep.
- int **separarseqinvertidas** (char seqf[][**COLSEQ**], int numdeseq)
  Função separarseqinvertidas.
- void **criapecasint** (int pecasi[][**COL**], int size)
  Função criapecasint.
- void **criapecasstr** (char pecass[][**COLSTR**], int s)
  Função criapecasstr.
- int **entregarbaralhos** (char pecass[][**COLSTR**], char baralhoss[][**COLSTR**], int n)
  Função entregarbaralhos.
- void **printpecasint** (int pecasi[][**COL**], int l, int c, int inicio)
  Função printpecasint.
- void **printpecasstr** (char pecass[][**COLSTR**], int inicio, int fim)
  Função printpecasstr.
- void **printseqstr** (char seqss[][**COLSEQ**], int inicio, int fim)
  Função printseqstr.
- void **mostrarjogosstr** (char baralhoss[][**COLSTR**], int njogos)
  Função mostrarjogosstr.
- void **mostrarjogosint** (int baralhosi[][**COL**], int njogos)
  Função mostrarjogosint.
- int **preenchebaralhos** (char pecass[][**COLSTR**], char baralhoss[][**COLSTR**], int n)
  Função preenchebaralhos.
- void **esvaziabaralhoint** (int baralhosi[][**COL**], int lin, int col)
  Função esvaziabaralhoint.
- void **esvaziabaralhostr** (char baralhoss[][**COLSTR**], int size)
  Função esvaziabaralhostr.
- void **esvaziaseqstr** (char seqss[][**COLSEQ**], int size)
  Função esvaziaseqstr.
- int **convertestrtoint** (char baralhoss[][**COLSTR**], int baralhosi[][**COL**], int num)
  Função convertestrtoint.
- int **converteinttostr** (int baralhosi[][**COL**], char baralhoss[][**COLSTR**], int num)
  Função converteinttostr.
- int **remover** (char baralhoss[][**COLSTR**], char pecass[][**COLSTR**], int num)
  Função remover.
- int **modificar** (char baralhoss[][**COLSTR**], char pecass[][**COLSTR**], int num)
  Função modificar.
- int **seq** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int num)

Função seq.

- char * **inverterstr** (char str1[], char str2[])
  Função inverterstr.
- int **procsubseq_ausar** (char seqf[][**COLSEQ**], int size, char subs[])
  Função procsubseq_ausar.
- int **procsubseq** (char seqf[][**COLSEQ**], int size, char subs[])
  Função procsubseq.
- int **strtoque** (char stra[][**COLSEQ**], char str[], char car)
  Função strtoque.
- void **ordernarmatrizinteiros** (int m[][2], int size)
  Função ordernarmatrizinteiros.
- void **ordernarsequencias** (char seqf[][**COLSEQ**], int size)
  Função ordernarsequencias.
- int **procsubseq_trocapadrao** (char seqf[][**COLSEQ**], int size, char subs[**LIN**], int l)
  Função procsubseq_trocapadrao.
- void **trocapadrao** (char seqf[][**COLSEQ**], int size, char padrao[], char padraon[], char seqfpadrao[][**COLSEQ**], int *sizeseqfpadrao)
  Função trocapadrao.
- int **tirartracosinvertidos** (char seqf[][**COLSEQ**], int numdeseq)
  Função tirartracosinvertidos.
- int **seqcomseqincial** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int num, char seqinit[])
  Função seqcomseqincial.
- int **retiraseqinitrepetida** (char seqf[][**COLSEQ**], int size, char seqinit[])
  Função retiraseqinitrepetida.
- int **jogoadois** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int num, char seqinit[])
  Função jogoadois.
- int **baralhoausar** (char baralhoss[][**COLSTR**], char baralhoaux[][**COLSEQ**], int numero)
  Função baralhoausar.

Documentação das funções

int baralhoausar (char *baralhoss*[][COLSTR], char *baralhoaux*[][COLSEQ], int *numero*)

Função baralhoausar.

esta funcao recebe as pecas de um baralho e acrescenta as invertidas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	baralhoaux[][COLSEQ]baralhos do jogador em string auxiliar na qual guardo também as peças invertidas
int	numero contém o numero do baralho ao qual vamos acrescentar as pecas invertidas também

Retorna:

retorna o numero de pecas do baralho do jogador com as pecas invertidas incluidas e retirando as repetidas

int converteinttostr (int *baralhosi*[][COL], char *baralhoss*[][COLSTR], int *num*)

Função converteinttostr.

esta funcao converte as peças inteiras em string

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
int	baralhosi[][COL] baralho do jogador em inteiros
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

int convertestrtoint (char *baralhoss*[][COLSTR], int *baralhosi*[][COL], int *num*)

Função convertestrtoint.

esta funcao converte as pecas do jogador de string para inteiros

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
int	baralhosi[][COL] baralho do jogador em inteiros
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

void criapecasint (int *pecasi*[][COL], int *size*)

Função criapecasint.

void criapecasstr (char *pecass*[][COLSTR], int *s*)

Função criapecasstr.

função que cria as pecas em strings

Parâmetros:

char	pecass[][COLSTR] array que guarda as pecas em string
int	s contém o numero de pecas a ser guardadas

int* eliminarep (int * *pv*, int * *psize*)

Função eliminarep.

função eliminarep, serve para eliminar as sequencias repetidas, recebe os indices das sequencias e a quantidade de sequencias e remove as repetidas que são as que contém -1 e aloca espaco

para as que não sao repetidas e retorna essa quantidade para imprimir só as que não são repetidas

Parâmetros:

int	*pv apontador para um array de inteiros que contém os indices das sequencias
int	*psize quantidade de sequencias recebidas no array de inteiros

Retorna:

retorna o novo tamanho do array de inteiros para depois apenas imprimir as sequências que possuem os indices com sequencias que não são repetidas

int entregarbaralhos (char *pecass*[][COLSTR], char *baralhoss*[][COLSTR], int *n*)

Função entregarbaralhos.

função que entrega os baralhos aos jogadores aleatóriamente sem pecas repetidas para cada jogador (baralhos de 7 pecas e máximo de 4 jogadores)

Parâmetros:

pecass[][COLSTR]	array de string que contém as pecas todas
char	baralhoss[][COLSTR] array de strings que con´tem os jogos de cada jogador
int	n numero de jogos entregues

void esvaziabaralhoint (int *baralhosi*[][COL], int *lin*, int *col*)

Função esvaziabaralhoint.

esta funcao coloca o baralho de inteiros vazio

Parâmetros:

int	baralhosi[][COL] baralhos de pecas dos jogadores em inteiros
int	lin linhas do baralho a esvaziar
int	col colunas do baralho a esvaziar

void esvaziabaralhostr (char *baralhoss*[][COLSTR], int *size*)

Função esvaziabaralhostr.

esta funcao coloca o baralho de strings vazio

Parâmetros:

char	baralhoss[][COLSTR] baralhos de pecas dos jogadores em strings
int	size tamanho do baralho

void esvaziaseqstr (char *seqss*[][COLSEQ], int *size*)

Função esvaziaseqstr.

esta funcao coloca o array das sequencias vazias

Parâmetros:

char	seqss[][COLSEQ] array de strings das sequencias
int	size numero de sequencias inseridas

char* inverterstr (char *str1*[], char *str2*[])

Função inverterstr.

função para inverter uma sequencia

Parâmetros:

char	str1[] string a inverter
char	str2[] string invertida a retornar

Retorna:

retorna a string invertida

int jogoadois (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int *num*, char *seqinit*[])

Função jogoadois.

esta funcao é igual à função das sequencias, só que nestas posso começar inicialmente com uma sequencia on nao e os jogadores jogam à vez de cada baralho

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a calcular sequencias
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

VERIFICA QUANTIDADE DE CORRESPONDENCIAS

JUNTA AS CORRESPONDENCIAS

int main_domino (int *argc*, char * *argv*[])

Função main.

função que contém o menu do programa e as respetivas chamadas as funcoes, neste momento estou também a fazer algumas verificações para verificar se as subsequencias a procurar são válidas, se os padrões a substituir são válidos. Estas verificações passarão posteriormente a ser realizadas dentro de funções

Parâmetros:

int	argc não está a ser usado mas contém o numero de posicoes do array de strings argv[] usadas,
char	*argv[] não está a ser usado, mas contém o numero de posições do array de

	strings

Retorna:

0

MENU

int modificar (char *baralhoss*[][COLSTR], char *pecass*[][COLSTR], int *num*)

Função modificar.

esta funcao altera as peças depois de removidas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	pecass[][COLSTR] pecas todas do jogo em strings
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

void mostrarjogosint (int *baralhosi*[][COL], int *njogos*)

Função mostrarjogosint.

Funcao que recebe o numero de jogos a imprimir em inteiros e os jogos dos jogadores e imprime separadamente os jogos dos jogadores

Parâmetros:

int	baralhosi[][COL] array de inteiros que guarda os jogos dos jogadores
int	njogos numero de jogos a imprimir

void mostrarjogosstr (char *baralhoss*[][COLSTR], int *njogos*)

Função mostrarjogosstr.

Funcao que recebe o numero de jogos a imprimir em string e os jogos dos jogadores e imprime separadamente os jogos dos jogadores

Parâmetros:

char	baralhoss[][COLSTR] array de strings que guarda os jogos dos jogadores
int	njogos numero de jogos a imprimir

void ordernarmatrizinteiros (int *m*[][2], int *size*)

Função ordernarmatrizinteiros.

esta função recebe uma matriz e ordena crescente ou decrescente

Parâmetros:

int	m[][2] matriz a ordenar
int	size tamanho da matriz

void ordernarsequencias (char *seqf*[][COLSEQ], int *size*)

Função ordernarsequencias.

esta função ordena as sequenciass por ordem decrescente

Parâmetros:

seqf[][COLSEQ]	array de strings das sequencias finais a ordenar
int	size numero de sequencias finais

int preenchebaralhos (char *pecass*[][COLSTR], char *baralhoss*[][COLSTR], int *n*)

Função preenchebaralhos.

funcão para inserir os baralhos dos jogadores manualmente

Parâmetros:

int	pecass[][COLSTR] array que possui as pecas todas do jogo
char	baralhoss[][COLSTR] array que possui os jogos dos jogadores
int	n numero de jogos a preencher manualmente

void printpecasint (int *pecasi*[][COL], int *l*, int *c*, int *inicio*)

Função printpecasint.

função que imprime as pecas inteiras

Parâmetros:

int	pecasi[][COL] pecas total inteiras
int	l linhas da matriz
int	c colunas da matriz
int	inicio variavel que contem o numero a partir do qual queremos imprimir

void printpecasstr (char *pecass*[][COLSTR], int *inicio*, int *fim*)

Função printpecasstr.

função para imprimir peças string

Parâmetros:

int	inicio desde quando é que queremos imprimir
int	fim até onde queremos imprimir

void printseqstr (char *seqss*[][COLSEQ], int *inicio*, int *fim*)

Função printseqstr.

função para imprimir sequencias

Parâmetros:

int	inicio desde quando é que queremos imprimir
int	fim até onde queremos imprimir

int procsubseq (char *seqf*[][COLSEQ], int *size*, char *subs*[])

Função procsubseq.

função para procurar uma substring numa string

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	subs[] sub string a procurar

Retorna:

retorna a posicao em que encontrou a substring

int procsubseq_ausar (char *seqf*[][COLSEQ], int *size*, char *subs*[])

Função procsubseq_ausar.

função para procurar uma substring numa string

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	subs[] sub string a procurar

Retorna:

retorna a posicao em que encontrou a substring

int procsubseq_trocapadrao (char *seqf*[][COLSEQ], int *size*, char *subs*[LIN], int *l*)

Função procsubseq_trocapadrao.

função para procurar uma substring numa string (utilizei esta variante na torca de padrao)

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring

char	subs[] sub string a procurar
int	l numero da string que vamos procurar uma substring

Retorna:

retorna a posicao em que encontrou a substring

int remover (char *baralhoss*[][COLSTR], char *pecass*[][COLSTR], int *num*)

Função remover.

esta funcao remove as pecas inseridas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	pecass[][COLSTR] pecas todas do jogo em strings
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a remover

int retiraseqinitrepetida (char *seqf*[][COLSEQ], int *size*, char *seqinit*[])

Função retiraseqinitrepetida.

Nesta funcao verifico se encontro a sequencia inicial ou a inicial invertida repetida na mesma sequencia para poder eliminá-la

Parâmetros:

char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	size numero de jogos a retirar sequencias com sequencias iniciais repetidas
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

int separarseqinvertidas (char *seqf*[][COLSEQ], int *numdeseq*)

Função separarseqinvertidas.

função que distingue as sequencias iguais lidas da esquerda para a direita às lidas da direita para a esquerda e chama a funcao elimina rep para retirar as que sao iguais escritas ao contrário

Parâmetros:

char	seqf[][COLSEQ] array de sequencias
int	numdeseq guarda o numero de sequencias

Retorna:

retorna o numero de sequencias

dou indices ás sequencias invertidas e ás sequencias normais para dps eliminar os duplicados

int seq (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int *num*)

Função seq.

função para criar as sequencias de peças existentes

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a calcular sequencias

Retorna:

retorna o numero de sequencias que encontrou

int seqcomseqincial (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int *num*, char *seqinit*[])

Função seqcomseqincial.

esta funcao é igual à função das sequencias, só que nestas começo inicialmente com uma sequencia

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a converter
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

VERIFICA QUANTIDADE DE CORRESPONDENCIAS

JUNTA AS CORRESPONDENCIAS

int strtoque (char *stra*[][COLSEQ], char *str*[], char *car*)

Função strtoque.

esta função recebe um array de strings, uma string e o carater pelo qual vai partir e faz o strtok()

Parâmetros:

char	stra[][COLSEQ] array de strings na qual vamos guardar as strings partidas
char	str[] string que queremos partir
char	car carater pelo qual partimos

Retorna:

retorna o numero de pecas partidas

int tirartracosinvertidos (char *seqf*[][COLSEQ], int *numdeseq*)

Função tirartracosinvertidos.

esta função serve para eliminar as sequencias que ão repetidas mas que estão escritas da direita para a esquerda, nas quais coloquei um "-" e agora elimino-as e puxo as outras para ci

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	numdeseq numero de sequencias

Retorna:

retorna o numero de sequencias menos as que estavam escritas da esquerda para a direita

void trocapadrao (char *seqf*[][COLSEQ], int *size*, char *padrao*[], char *padraon*[], char *seqfpadrao*[][COLSEQ], int * *sizeseqfpadrao*)

Função trocapadrao.

função para procurar uma substring numa string (utilizei esta variante na torca de padrao)

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	padrao[] string a substituir
char	padraon[] string substituta
char	seqfpadrao[][COLSEQ]
int	*sizeseqfpadrao indice da sequencia padrao na qual vou trocar o padrao

CICLO DAS OCURRENCIAS DO PADRAO por indices da string

int verificasequencia (char *seq*[])

Função verificasequencia.

verifica se a sequência é possível juntar ex:2|3-3|4 estas pecas encaixam e esta funcao faz esta verificação

Parâmetros:

char	seq[] recebe a sequência de peças

Retorna:

Retorna 0 se for possível e 1 se não for

Referência ao ficheiro projeto.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

Macros

- #define **LIN** 28
- #define **COL** 2
- #define **COLSTR** 4
- #define **LINSEQ** 800
- #define **COLSEQ** 200

Funções

- void **criapecasint** (int pecasi[][**COL**], int)
  Função criapecasint.
- void **criapecasstr** (char pecass[][**COLSTR**], int)
  Função criapecasstr.
- void **printpecasint** (int pecasi[][**COL**], int, int, int)
  Função printpecasint.
- void **printpecasstr** (char pecass[][**COLSTR**], int, int)
  Função printpecasstr.
- void **printseqstr** (char seqss[][**COLSEQ**], int, int)
  Função printseqstr.
- int **entregarbaralhos** (char pecass[][**COLSTR**], char baralhoss[][**COLSTR**], int)
  Função entregarbaralhos.
- int **preenchebaralhos** (char pecass[][**COLSTR**], char baralhoss[][**COLSTR**], int)
  Função preenchebaralhos.
- void **mostrarjogosstr** (char baralhoss[][**COLSTR**], int)
  Função mostrarjogosstr.
- void **mostrarjogosint** (int baralhosint[][**COL**], int)
  Função mostrarjogosint.
- void **esvaziabaralhoint** (int baralhosi[][**COL**], int, int)
  Função esvaziabaralhoint.
- void **esvaziabaralhostr** (char baralhoss[][**COLSTR**], int)
  Função esvaziabaralhostr.
- void **esvaziaseqstr** (char seqss[][**COLSEQ**], int)
  Função esvaziaseqstr.
- int **convertestrtoint** (char baralhoss[][**COLSTR**], int baralhosi[][**COL**], int)
  Função convertestrtoint.
- int **converteinttostr** (int baralhosi[][**COL**], char baralhoss[][**COLSTR**], int)
  Função converteinttostr.
- int **remover** (char baralhoss[][**COLSTR**], char pecass[][**COLSTR**], int)
  Função remover.
- int **modificar** (char baralhoss[][**COLSTR**], char pecass[][**COLSTR**], int)
  Função modificar.

- int **seq** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int)
  Função seq.
- int **procsubseq** (char seqf[][**COLSEQ**], int size, char subs[])
  Função procsubseq.
- int **procsubseq_trocapadrao** (char seqf[][**COLSEQ**], int size, char subs[], int)
- void **ordernarsequencias** (char seqf[][**COLSEQ**], int)
  Função ordernarsequencias.
- void **ordernarmatrizinteiros** (int m[][2], int)
  Função ordernarmatrizinteiros.
- void **trocapadrao** (char seqf[][**COLSEQ**], int size, char padrao[], char padraon[], char seqfpadrao[][**COLSEQ**], int *sizeseqfpadrao)
  Função trocapadrao.
- int **tirartracosinvertidos** (char seqf[][**COLSEQ**], int numdeseq)
  Função tirartracosinvertidos.
- int **seqcomseqincial** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int, char seqinit[])
  Função seqcomseqincial.
- int **separarseqinvertidas** (char seqf[][**COLSEQ**], int)
  Função separarseqinvertidas.
- int **strtoque** (char stra[][**COLSEQ**], char str[], char)
  Função strtoque.
- int **retiraseqinitrepetida** (char seqf[][**COLSEQ**], int size, char seqinit[])
  Função retiraseqinitrepetida.
- int **jogoadois** (char baralhoss[][**COLSTR**], char seqf[][**COLSEQ**], int, char seqinit[])
  Função jogoadois.
- int * **eliminarep** (int *, int *)
  Função eliminarep.
- char * **inverterstr** (char str1[], char str2[])
  Função inverterstr.
- int **baralhoausar** (char baralhoss[][**COLSTR**], char baralhoaux[][**COLSEQ**], int)
  Função baralhoausar.
- int **verificasequencia** (char **seq**[])
  Função verificasequencia.
- int **procsubseq_ausar** (char seqf[][**COLSEQ**], int, char subs[])
  Função procsubseq_ausar.
- int **main_domino** (int argc, char *argv[])
  Função main.

Documentação das macros

#define COL 2

#define COLSEQ 200

#define COLSTR 4

#define LIN 28

#define LINSEQ 800

Documentação das funções

int baralhoausar (char *baralhoss*[][COLSTR], char *baralhoaux*[][COLSEQ], int)

Função baralhoausar.

esta funcao recebe as pecas de um baralho e acrescenta as invertidas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	baralhoaux[][COLSEQ]baralhos do jogador em string auxiliar na qual guardo também as peças invertidas
int	numero contém o numero do baralho ao qual vamos acrescentar as pecas invertidas também

Retorna:

retorna o numero de pecas do baralho do jogador com as pecas invertidas incluidas e retirando as repetidas

int converteinttostr (int *baralhosi*[][COL], char *baralhoss*[][COLSTR], int)

Função converteinttostr.

esta funcao converte as peças inteiras em string

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
int	baralhosi[][COL] baralho do jogador em inteiros
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

int convertestrtoint (char *baralhoss*[][COLSTR], int *baralhosi*[][COL], int)

Função convertestrtoint.

esta funcao converte as pecas do jogador de string para inteiros

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
int	baralhosi[][COL] baralho do jogador em inteiros
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

void criapecasint (int *pecasi*[][COL], int)

Função criapecasint.

void criapecasstr (char *pecass*[][COLSTR], int)

Função criapecasstr.

função que cria as pecas em strings

Parâmetros:

char	pecass[][COLSTR] array que guarda as pecas em string
int	s contém o numero de pecas a ser guardadas

int* eliminarep (int * , int *)

Função eliminarep.

função eliminarep, serve para eliminar as sequencias repetidas, recebe os indices das sequencias e a quantidade de sequencias e remove as repetidas que são as que contém -1 e aloca espaco para as que não sao repetidas e retorna essa quantidade para imprimir só as que não são repetidas

Parâmetros:

int	*pv apontador para um array de inteiros que contém os indices das sequencias
int	*psize quantidade de sequencias recebidas no array de inteiros

Retorna:

retorna o novo tamanho do array de inteiros para depois apenas imprimir as sequências que possuem os indices com sequencias que não são repetidas

int entregarbaralhos (char *pecass*[][COLSTR], char *baralhoss*[][COLSTR], int)

Função entregarbaralhos.

função que entrega os baralhos aos jogadores aleatóriamente sem pecas repetidas para cada jogador (baralhos de 7 pecas e máximo de 4 jogadores)

Parâmetros:

pecass[][COLSTR	array de string que contém as pecas todas

]	
char	baralhoss[][COLSTR] array de strings que con´tem os jogos de cada jogador
int	n numero de jogos entregues

void esvaziabaralhoint (int *baralhosi*[][COL], int , int)

Função esvaziabaralhoint.

esta funcao coloca o baralho de inteiros vazio

Parâmetros:

int	baralhosi[][COL] baralhos de pecas dos jogadores em inteiros
int	lin linhas do baralho a esvaziar
int	col colunas do baralho a esvaziar

void esvaziabaralhostr (char *baralhoss*[][COLSTR], int)

Função esvaziabaralhostr.

esta funcao coloca o baralho de strings vazio

Parâmetros:

char	baralhoss[][COLSTR] baralhos de pecas dos jogadores em strings
int	size tamanho do baralho

void esvaziaseqstr (char *seqss*[][COLSEQ], int)

Função esvaziaseqstr.

esta funcao coloca o array das sequencias vazias

Parâmetros:

char	seqss[][COLSEQ] array de strings das sequencias
int	size numero de sequencias inseridas

char* inverterstr (char *str1*[], char *str2*[])

Função inverterstr.

função para inverter uma sequencia

Parâmetros:

char	str1[] string a inverter
char	str2[] string invertida a retornar

Retorna:

retorna a string invertida

int jogoadois (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int , char *seqinit*[])

Função jogoadois.

esta funcao é igual à função das sequencias, só que nestas posso começar inicialmente com uma sequencia on nao e os jogadores jogam à vez de cada baralho

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a calcular sequencias
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

VERIFICA QUANTIDADE DE CORRESPONDENCIAS

JUNTA AS CORRESPONDENCIAS

int main_domino (int *argc*, char * *argv*[])

Função main.

função que contém o menu do programa e as respetivas chamadas as funcoes, neste momento estou também a fazer algumas verificações para verificar se as subsequencias a procurar são válidas, se os padrões a substituir são válidos. Estas verificações passarão posteriormente a ser realizadas dentro de funções

Parâmetros:

int	argc não está a ser usado mas contém o numero de posicoes do array de strings argv[] usadas,
char	*argv[] não está a ser usado, mas contém o numero de posições do array de strings

Retorna:

0

MENU

int modificar (char *baralhoss*[][COLSTR], char *pecass*[][COLSTR], int)

Função modificar.

esta funcao altera as peças depois de removidas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	pecass[][COLSTR] pecas todas do jogo em strings
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a imprimir

void mostrarjogosint (int *baralhosint*[][COL], int)

Função mostrarjogosint.

Funcao que recebe o numero de jogos a imprimir em inteiros e os jogos dos jogadores e imprime separadamente os jogos dos jogadores

Parâmetros:

int	baralhosi[][COL] array de inteiros que guarda os jogos dos jogadores
int	njogos numero de jogos a imprimir

void mostrarjogosstr (char *baralhoss*[][COLSTR], int)

Função mostrarjogosstr.

Funcao que recebe o numero de jogos a imprimir em string e os jogos dos jogadores e imprime separadamente os jogos dos jogadores

Parâmetros:

char	baralhoss[][COLSTR] array de strings que guarda os jogos dos jogadores
int	njogos numero de jogos a imprimir

void ordernarmatrizinteiros (int *m*[][2], int)

Função ordernarmatrizinteiros.

esta função recebe uma matriz e ordena crescente ou decrescente

Parâmetros:

int	m[][2] matriz a ordenar
int	size tamanho da matriz

void ordernarsequencias (char *seqf*[][COLSEQ], int)

Função ordernarsequencias.

esta função ordena as sequenciass por ordem decrescente

Parâmetros:

seqf[][COLSEQ]	array de strings das sequencias finais a ordenar
int	size numero de sequencias finais

int preenchebaralhos (char *pecass*[][COLSTR], char *baralhoss*[][COLSTR], int)

Função preenchebaralhos.

funcão para inserir os baralhos dos jogadores manualmente

Parâmetros:

int	pecass[][COLSTR] array que possui as pecas todas do jogo
char	baralhoss[][COLSTR] array que possui os jogos dos jogadores
int	n numero de jogos a preencher manualmente

void printpecasint (int *pecasi*[][COL], int , int , int)

Função printpecasint.

função que imprime as pecas inteiras

Parâmetros:

int	pecasi[][COL] pecas total inteiras
int	l linhas da matriz
int	c colunas da matriz
int	inicio variavel que contem o numero a partir do qual queremos imprimir

void printpecasstr (char *pecass*[][COLSTR], int , int)

Função printpecasstr.

função para imprimir peças string

Parâmetros:

int	inicio desde quando é que queremos imprimir
int	fim até onde queremos imprimir

void printseqstr (char *seqss*[][COLSEQ], int , int)

Função printseqstr.

função para imprimir sequencias

Parâmetros:

int	inicio desde quando é que queremos imprimir
int	fim até onde queremos imprimir

int procsubseq (char *seqf*[][COLSEQ], int *size*, char *subs*[])

Função procsubseq.

função para procurar uma substring numa string

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	subs[] sub string a procurar

Retorna:

retorna a posicao em que encontrou a substring

int procsubseq_ausar (char *seqf*[][COLSEQ], int , char *subs*[])

Função procsubseq_ausar.

função para procurar uma substring numa string

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	subs[] sub string a procurar

Retorna:

retorna a posicao em que encontrou a substring

int procsubseq_trocapadrao (char *seqf*[][COLSEQ], int *size*, char *subs*[], int)

int remover (char *baralhoss*[][COLSTR], char *pecass*[][COLSTR], int)

Função remover.

esta funcao remove as pecas inseridas

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	pecass[][COLSTR] pecas todas do jogo em strings
int	num numero de jogos a converter

Retorna:

retorna o numero de jogos a remover

int retiraseqinitrepetida (char *seqf*[][COLSEQ], int *size*, char *seqinit*[])

Função retiraseqinitrepetida.

Nesta funcao verifico se encontro a sequencia inicial ou a inicial invertida repetida na mesma sequencia para poder eliminá-la

Parâmetros:

char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	size numero de jogos a retirar sequencias com sequencias iniciais repetidas
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

int separarseqinvertidas (char *seqf*[][COLSEQ], int)

Função separarseqinvertidas.

função que distingue as sequencias iguais lidas da esquerda para a direita às lidas da direita para a esquerda e chama a funcao elimina rep para retirar as que sao iguais escritas ao contrário

Parâmetros:

char	seqf[][COLSEQ] array de sequencias
int	numdeseq guarda o numero de sequencias

Retorna:

retorna o numero de sequencias

dou indices ás sequencias invertidas e ás sequencias normais para dps eliminar os duplicados

int seq (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int)

Função seq.

função para criar as sequencias de peças existentes

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a calcular sequencias

Retorna:

retorna o numero de sequencias que encontrou

int seqcomseqincial (char *baralhoss*[][COLSTR], char *seqf*[][COLSEQ], int , char *seqinit*[])

Função seqcomseqincial.

esta funcao é igual à função das sequencias, só que nestas começo inicialmente com uma sequencia

Parâmetros:

char	baralhoss[][COLSTR] baralhos do jogador em string
char	seqf[][COLSEQ] array final onde vou guardar as sequencias todas possiveis
int	num numero de jogos a converter
char	seqinit[] sequencia inicial

Retorna:

retorna o numero de sequencias que encontrou

VERIFICA QUANTIDADE DE CORRESPONDENCIAS

JUNTA AS CORRESPONDENCIAS

int strtoque (char *stra*[][COLSEQ], char *str*[], char)

Função strtoque.

esta função recebe um array de strings, uma string e o carater pelo qual vai partir e faz o strtok()

Parâmetros:

char	stra[][COLSEQ] array de strings na qual vamos guardar as strings partidas
char	str[] string que queremos partir
char	car carater pelo qual partimos

Retorna:

retorna o numero de pecas partidas

int tirartracosinvertidos (char *seqf*[][COLSEQ], int *numdeseq*)

Função tirartracosinvertidos.

esta função serve para eliminar as sequencias que ão repetidas mas que estão escritas da direita para a esquerda, nas quais coloquei um "-" e agora elimino-as e puxo as outras para ci

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	numdeseq numero de sequencias

Retorna:

retorna o numero de sequencias menos as que estavam escritas da esquerda para a direita

void trocapadrao (char *seqf*[][COLSEQ], int *size*, char *padrao*[], char *padraon*[], char *seqfpadrao*[][COLSEQ], int * *sizeseqfpadrao*)

Função trocapadrao.

função para procurar uma substring numa string (utilizei esta variante na torca de padrao)

Parâmetros:

char	seqf[][COLSEQ] array final de strings
int	size numero da linha do array de strings final que vamos querer procurar uma substring
char	padrao[] string a substituir
char	padraon[] string substituta
char	seqfpadrao[][COLSEQ]
int	*sizeseqfpadrao indice da sequencia padrao na qual vou trocar o padrao

CICLO DAS OCURRENCIAS DO PADRAO por indices da string

int verificasequencia (char *seq*[])

Função verificasequencia.

verifica se a sequência é possível juntar ex:2|3-3|4 estas pecas encaixam e esta funcao faz esta verificação

Parâmetros:

char	seq[] recebe a sequência de peças

Retorna:

Retorna 0 se for possível e 1 se não for