#include <stdio.h>

#include <string.h>

#include <stdlib.h>

// baseado no jogo do 2048: https://2048game.com/pt/

// Num tabuleiro de 4x4 aparece um número 1 numa casa aleatória vazia.

// Um de 4 movimentos, cima, esquerda, direita, baixo, desloca todos os números para uma das bordas

// Os números que estiverem nessa direção iguais e juntos, somam-se.

// O jogo acaba quando não há mais movimentos, sendo o valor máximo a prestação do jogo

void MostraTab(int tabuleiro[4][4])

{

 int i, j;

 printf("\n+----+----+----+----+");

 for (i = 0; i < 4; i++) {

 printf("\n|");

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] > 0)

 printf("%4d|", tabuleiro[i][j]);

 else

 printf(" |");

 printf("\n+----+----+----+----+");

 }

}

// Erros comuns:

// - não mostrar as bordas

// - imprime 0 em vez de imprimir espaços

void InicializaTab(int tabuleiro[4][4])

{

 int i, j;

 for (i = 0; i < 4; i++)

 for (j = 0; j < 4; j++)

 tabuleiro[i][j] = 0;

}

void InsereNumeroTab(int tabuleiro[4][4])

{

 int i, j; // gerar uma posição aleatória, até que tenha 0

 i = rand() % 4;

 j = rand() % 4;

 while (tabuleiro[i][j] > 0) {

 i = rand() % 4;

 j = rand() % 4;

 }

 tabuleiro[i][j] = 1;

}

void InsereNumeroTab2(int tabuleiro[4][4])

{

 int i, j; // gerar uma posição aleatória, até que tenha 0

 do {

 i = rand() % 4;

 j = rand() % 4;

 } while (tabuleiro[i][j] > 0);

 tabuleiro[i][j] = 1;

}

void InsereNumeroTab3(int tabuleiro[4][4])

{

 int posicao, i, j;

 int livres = LivresTab(tabuleiro);

 if (livres == 0)

 return;

 // gerar uma posição aleatória, de entre as livres

 posicao = rand() % livres;

 // percorrer o tabuleiro, e saltar as posições anteriores com 0

 // chegando à posição escolihida, colocar 1

 for (i = 0; i < 4; i++)

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] == 0)

 if (posicao-- == 0) {

 tabuleiro[i][j] = 1;

 return;

 }

}

void InsereNumeroTab4(int tabuleiro[4][4])

{

 int i, j; // gerar uma posição aleatória, até que tenha 0

 i = rand() % 4;

 j = rand() % 4;

 if (tabuleiro[i][j] == 0)

 tabuleiro[i][j] = 1;

 else

 InsereNumeroTab4(tabuleiro);

}

int LivresTab(int tabuleiro[4][4])

{

 int i, j, livres = 0;

 for (i = 0; i < 4; i++)

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] == 0)

 livres++;

 return livres;

}

int MaiorNumeroTab(int tabuleiro[4][4])

{

 int i, j, maior = 0;

 for (i = 0; i < 4; i++)

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] > maior)

 maior = tabuleiro[i][j];

 return maior;

}

void MoveNumero(int tabuleiro[4][4], int i, int j, int di, int dj)

{

 while (i + di >= 0 && i + di < 4 && j + dj >= 0 && j + dj < 4) {

 if (tabuleiro[i + di][j + dj] == 0) { // deslocar número

 tabuleiro[i + di][j + dj] = tabuleiro[i][j];

 tabuleiro[i][j] = 0;

 }

 else if (tabuleiro[i + di][j + dj] == tabuleiro[i][j]) { // juntar número

 tabuleiro[i + di][j + dj] += tabuleiro[i][j];

 tabuleiro[i][j] = 0;

 return; // terminar, o número já foi junto com outro

 }

 else // caso contrário o número já não pode ser deslocado

 return;

 i += di;

 j += dj;

 }

}

// Erros comuns:

// - não fazer um ciclo, para o caso de movimento do número

// - não parar se o número se juntar a outro

// - não parar caso o exista um número diferente lá (e distinto de 0)

// - não atualizar uma das variáveis para a próxima iteração

void MoveNumero2(int tabuleiro[4][4], int i, int j, int di, int dj)

{

 int iProx = i + di, jProx = j + dj;

 if (iProx >= 0 && iProx < 4 && jProx >= 0 && jProx < 4) {

 if (tabuleiro[iProx][jProx] == 0) { // deslocar número

 tabuleiro[iProx][jProx] = tabuleiro[i][j];

 tabuleiro[i][j] = 0;

 MoveNumero2(tabuleiro, iProx, jProx, di, dj);

 }

 else if (tabuleiro[iProx][jProx] == tabuleiro[i][j]) { // juntar número

 tabuleiro[iProx][jProx] += tabuleiro[i][j];

 tabuleiro[i][j] = 0;

 }

 }

}

void JogarTab(int tabuleiro[4][4], char jogada)

{

 int i, j;

 if (jogada == 'C' || jogada == 'c') {

 // percorrer tabuleiro de cima para baixo

 for (i = 0; i < 4; i++)

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] > 0)

 MoveNumero(tabuleiro, i, j, -1, 0);

 }

 else if (jogada == 'B' || jogada == 'b') {

 // percorrer tabuleiro de baixo para cima

 for (i = 3; i >= 0; i--)

 for (j = 0; j < 4; j++)

 if (tabuleiro[i][j] > 0)

 MoveNumero(tabuleiro, i, j, 1, 0);

 }

 else if (jogada == 'E' || jogada == 'e') {

 // percorrer tabuleiro da direita para a esquerda

 for (j = 0; j < 4; j++)

 for (i = 0; i < 4; i++)

 if (tabuleiro[i][j] > 0)

 MoveNumero(tabuleiro, i, j, 0, -1);

 }

 else if (jogada == 'D' || jogada == 'd') {

 // percorrer tabuleiro da esquerda para a direita

 for (j = 3; j >= 0; j--)

 for (i = 0; i < 4; i++)

 if (tabuleiro[i][j] > 0)

 MoveNumero(tabuleiro, i, j, 0, 1);

 }

}

int main()

{

 int tabuleiro[4][4];

 char jogada[256];

 srand(0);

 InicializaTab(tabuleiro);

 while (LivresTab(tabuleiro) > 0) {

 InsereNumeroTab(tabuleiro);

 MostraTab(tabuleiro);

 printf("\nJogada (C/B/E/D): ");

 scanf("%s", &jogada);

 JogarTab(tabuleiro, jogada[0]);

 }

 printf("\nResultado: %d.", MaiorNumeroTab(tabuleiro));

}