#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "../../include\_jogos/jogos.h"

int jogo7Dezenas[8] = {0,0,0,0,0,0,0,0};

void ordenaJogo(){

int i, j, aux;

for( i=0; i<7; i++ ){

for( j=i+1; j<8; j++ ){

if( jogo7Dezenas[i] > jogo7Dezenas[j] ){

aux = jogo7Dezenas[i];

jogo7Dezenas[i] = jogo7Dezenas[j];

jogo7Dezenas[j] = aux;

}

}

}

}

int main()

{

int i, j, k=0, x, \_x = 0, y, MAX = 18, cont = 0, it=0;

FILE \*f;

int contGeral = 0;

int dezenasFixas[18];

int dezenas[18]; // Array de Dezenas tempor�rias

int gerados2[153][2];

int ignorados16[153][16];

int dezenasProibidas[6] = {0,0,0,0,0,0}, contDezenasProibidas;

int \_maisCont, \_comunsCont, \_menosCont;

int mais[15] = {10,5,53,23,4, 33,54,24,51,42, 17,37,27,52,28}, maisTemp[15];

int comuns[30] = {13,56,30,32,43, 6,34,36,16,2, 50,18,29,44,41,

35,8,38,12,11, 1,45,49,59,58, 40,47,20,46,31}, comunsTemp[30];

int menos[15] = {39,7,14,3,19, 60,25,48,57,15, 9,22,55,21,26}, menosTemp[15];

int distribuicao[60] = {10,5,53,23,4, 33,54,24,51,42, 17,37,27,52,28,

13,56,30,32,43, 6,34,36,16,2, 50,18,29,44,41,

35,8,38,12,11, 1,45,49,59,58, 40,47,20,46,31,

39,7,14,3,19, 60,25,48,57,15, 9,22,55,21,26}; // utilizado para verificar duplicidade em grupos 5 MCM

contGeral=0;

memset(dezenasProibidas, 0, 6);

contDezenasProibidas=0;

\_maisCont = \_comunsCont = \_menosCont = 0;

// + . - Par <31 proporcao

int param[12][6] = {{2 ,3 ,2 ,4 ,4 ,9}, // ex: 9%

{2 ,3 ,2 ,4 ,3 ,11}, // ex: 11%

{2 ,3 ,2 ,3 ,4 ,9},

{2 ,3 ,2 ,3 ,3 ,11},

{1 ,4 ,2 ,4 ,4 ,7},

{1 ,4 ,2 ,4 ,3 ,8},

{1 ,4 ,2 ,3 ,4 ,7},

{1 ,4 ,2 ,3 ,3 ,8},

{2 ,4 ,1 ,4 ,4 ,7},

{2 ,4 ,1 ,4 ,3 ,8},

{2 ,4 ,1 ,3 ,4 ,7},

{2 ,4 ,1 ,3 ,3 ,8}};

srand(time(NULL));

// Pega os TRES ultimos sorteios e preenche o vetor dezenasFixas;

for(i=0;i<3;i++){

for(j=0;j<6;j++){

dezenasFixas[k] = jogosMega[sizeof(jogosMega)/24-3+i][j];

k++;

}

}

// Eliminar Repetidas nas dezenas fixas

for(i=0;i<18;i++){

for(j=i+1;j<18;j++){

if(dezenasFixas[i] == dezenasFixas[j] && i>5 && j>6){ // Elimina a possibilidade de entrar no jogo

dezenasProibidas[contDezenasProibidas++]=dezenasFixas[i]; // as dezenas q se repetiram 2 vezes nos ultimos 2 jogos

dezenasFixas[i] = 0; // ja que isso eh raro.

dezenasFixas[j] = 0;

}

if(dezenasFixas[i] == dezenasFixas[j])

dezenasFixas[j] = 0;

}

}

for(i=0;i<153;i++){

for(j=0;j<2;j++) gerados2[i][j] = 0;

for(j=0;j<16;j++) ignorados16[i][j] = 0;

}

f = fopen("JogosGerados.txt", "w");

// Gera todas as combinacoes e divide-as entre GER e IGN

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

for(j=i;j<MAX;j++){

for(x=0;x<MAX;x++)

dezenas[x] = dezenasFixas[x];

if (dezenas[i] != dezenas[j] && dezenas[i]!=0 && dezenas[j]!=0){

gerados2[cont][0] = dezenas[i];

gerados2[cont][1] = dezenas[j];

dezenas[i] = 0;

dezenas[j] = 0;

x = 0;

\_x = 0;

while(x<MAX){

if(dezenas[x]!=0){

ignorados16[cont][\_x] = dezenas[x];

\_x++;

}

x++;

}

cont++;

}

}

}

int contZeros=0, contJogos=0;

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

for(k=0;k<2;k++){

if(gerados2[i][k] == 0){

contZeros++;

break;

}

}

if(contZeros==0) contJogos++;

contZeros = 0;

}

printf("\n----------- QUANTIDADE DE COMBINACOES 2 a 2 JOGOS: %d ------------\n\n", contJogos);

int preReqs = 0;

int indiceAleatorioComb=0;

int indicesNaoRepetidos[153];

int quantidadeDezenas = 50;

int contSeq=0;

int quant;

while(contSeq<12){

quant = 0;

while(quant < (int)(((float)param[contSeq][5])/100\*quantidadeDezenas)){

// COPIAR TABELA MCM PARA OS VETORES TEMPORARIOS

indiceAleatorioComb = rand()%contJogos; // Utiliza uma combinacao aleatoria das dezenas geradas; Somente 2 dezenas

//printf("combinacao %d \n", indiceAleatorioComb);

for(i=0;i<30;i++) {

if(i<15){

maisTemp[i] = mais[i];

menosTemp[i] = menos[i];

}

comunsTemp[i] = comuns[i];

}

// Eliminar as dezenas IGNORADAS das tabelas temporarias (Testado)

for(i=0;i<16;i++){

for(j=0;j<30;j++){

if(j<16){

if(ignorados16[indiceAleatorioComb][i] == maisTemp[j]) maisTemp[j] = 0;

if(ignorados16[indiceAleatorioComb][i] == menosTemp[j]) menosTemp[j] = 0;

}

if(ignorados16[indiceAleatorioComb][i] == comunsTemp[j]) comunsTemp[j] = 0;

}

}

// Eliminar as dezenas GERADAS das tabelas temporarias (Testado)

for(i=0;i<2;i++){

for(j=0;j<30;j++){

if(j<16){

if(gerados2[indiceAleatorioComb][i] == maisTemp[j]) maisTemp[j] = 0;

if(gerados2[indiceAleatorioComb][i] == menosTemp[j]) menosTemp[j] = 0;

}

if(gerados2[indiceAleatorioComb][i] == comunsTemp[j]) comunsTemp[j] = 0;

}

}

// Eliminar as dezenas PROIBIDAS das tabelas temporarias (Testado)

for(i=0;i<6;i++){

for(j=0;j<30;j++){

if(j<16){

if(dezenasProibidas[i] == maisTemp[j]) maisTemp[j] = 0;

if(dezenasProibidas[i] == menosTemp[j]) menosTemp[j] = 0;

}

if(dezenasProibidas[i] == comunsTemp[j]) comunsTemp[j] = 0;

}

}

// Verificar a quem as 2 dezenas Geradas pertence, quais faixas de valores (mais, comuns, menos)

// PARA a organizacao mais comum 2-3-2

\_maisCont = param[contSeq][0];

\_comunsCont = param[contSeq][1];

\_menosCont = param[contSeq][2];

int \_dezenasCont = 2; // o Vetor jogo ja contem 2 dezenas

for(i=0;i<2;i++)

jogo7Dezenas[i] = gerados2[indiceAleatorioComb][i]; // Carregando as combinacoes de 2 dezenas para o vetor jogo

for(i=0;i<2;i++){ // TESTADO -- OK

for(j=0;j<30;j++){

if(j<16){

if(jogo7Dezenas[i] == mais[j]) \_maisCont--;

if(jogo7Dezenas[i] == menos[j]) \_menosCont--;

}

if(jogo7Dezenas[i] == comuns[j]) \_comunsCont--;

}

}

// ------------------------ TESTADO OK -------------------------------

int indiceDezena=0;

int dezenaTemp;

while(\_maisCont>0){

indiceDezena = rand() % 15;

dezenaTemp = maisTemp[indiceDezena];

if(dezenaTemp!=0){

\_maisCont--;

jogo7Dezenas[\_dezenasCont] = dezenaTemp;

maisTemp[indiceDezena] = 0;

\_dezenasCont++;

}

}

while(\_menosCont>0){

indiceDezena = rand() % 15;

dezenaTemp = menosTemp[indiceDezena];

if(dezenaTemp!=0){

\_menosCont--;

jogo7Dezenas[\_dezenasCont] = dezenaTemp;

menosTemp[indiceDezena] = 0;

\_dezenasCont++;

}

}

while(\_comunsCont>0){

indiceDezena = rand() % 30;

dezenaTemp = comunsTemp[indiceDezena];

if(dezenaTemp!=0){

\_comunsCont--;

jogo7Dezenas[\_dezenasCont] = dezenaTemp;

comunsTemp[indiceDezena] = 0;

\_dezenasCont++;

}

}

// ------------------------ FIM, TESTADO OK -------------------------------

// Verificar se o jogo Atende aos requisitos

// 1 -------- Verificar a quantidade de pares 4-3 impares ----------

int quantPares = 0;

for(i=0;i<8;i++){

if(jogo7Dezenas[i]%2==0)

quantPares++;

}

if(quantPares==param[contSeq][3]) preReqs++; // param[contSeq][3] quantidade de pares

// ---------------------------------------------------------------------------------

// 2 -------------------------- Verificar 30 Divisor 4:3 -------------------------

int quantAbaixo30 = 0;

for(i=0;i<8;i++){

if(jogo7Dezenas[i]<=30)

quantAbaixo30++;

}

if(quantAbaixo30==param[contSeq][3]){

preReqs++;

}

// ---------------------------------------------------------------------------------

// 3 ------ 67,1% dos jogos contem duplicidade nos grupos de 5 dezenas MCM -------

int existeDupla=0;

for(j = 0; j < 60; j+=5){ // TESTADO OK: Mas em alguns casos acontece dupla duplicidade.. Verificar Estatisticas!

existeDupla = 0;

for(k = 0; k < 8; k++){

for(x = 0; x < 6; x++){

if(jogo7Dezenas[k] == distribuicao[j+x]) existeDupla++;

}

}

if(existeDupla == 2){break; break;}

}

if(existeDupla == 2) preReqs++;

// ---------------------------------------------------------------------------------

// 4 ------------ Permitir Apenas se nao aconteceu quinas e senas ----------------

int sorteioTemp[6];

int contDezenas, quinas=0, senas=0;

for(i = 0; i < (sizeof(jogosMega)/24); i++){

for(j=0;j<6;j++) sorteioTemp[j] = jogosMega[i][j];

contDezenas = 0;

for(j = 0; j < 6; j++){

for(k = 0; k < sizeof(jogo7Dezenas)/4; k++){

if(sorteioTemp[j] == jogo7Dezenas[k]){

contDezenas++;

break;

}

}

}

if(contDezenas == 5) quinas++;

if(contDezenas == 6) senas++;

}

if(quinas==0 && senas==0) preReqs++;

// ---------------------------------------------------------------------------------

// 5 --- nao Permitir jogos que comecem com 20 e terminem com 30 - sao raros -----

int contDezAte19 = 0;

int contDezAte40\_60 = 0;

for(i=0;i<8;i++){

if(jogo7Dezenas[i]<20) contDezAte19++;

if(jogo7Dezenas[i]>=40) contDezAte40\_60++;

}

if(contDezAte19>0 && contDezAte40\_60>0) preReqs++;

// ---------------------------------------------------------------------------------

// 6 ------ Permitir no maximo 2 Dezenas na mesma coluna por JOGO -------

int contDupla = 0;

int duques = 0;

int ternos = 0;

int tempJogo[7];

for(i=0;i<8;i++) tempJogo[i] = jogo7Dezenas[i];

for(i=0;i<8;i++){

contDupla = 0;

for(j=i+1;j<8;j++){

if(jogo7Dezenas[i]%10==jogo7Dezenas[j]%10 && jogo7Dezenas[i]!=0 && jogo7Dezenas[j]!=0){

contDupla++;

jogo7Dezenas[j]=0;

}

}

if(contDupla==1) duques++;

if(contDupla==2) ternos++;

}

if(duques==1 && ternos==0){

preReqs++;

for(i=0;i<8;i++) jogo7Dezenas[i] = tempJogo[i];

}

// ---------------------------------------------------------------------------------

// 7 ------ Permitir no maximo 3 numeros da mesma faixa de dezenas :ex, 22, 23, 27 -------

int cont09=0, cont1019=0, cont2029=0, cont3039=0, cont4049=0, cont5060=0;

for(i=0;i<8;i++){

if(jogo7Dezenas[i]<10) cont09++;

if(jogo7Dezenas[i]<20&&jogo7Dezenas[i]>=10) cont1019++;

if(jogo7Dezenas[i]<30&&jogo7Dezenas[i]>=20) cont2029++;

if(jogo7Dezenas[i]<40&&jogo7Dezenas[i]>=30) cont3039++;

if(jogo7Dezenas[i]<50&&jogo7Dezenas[i]>=40) cont4049++;

if(jogo7Dezenas[i]<=60&&jogo7Dezenas[i]>=50) cont5060++;

}

if(cont09<4 && cont1019<4 && cont2029<4 && cont3039<4 && cont4049<4 && cont5060<4) preReqs++;

// ------------------------------------------------------------------------------------

// --------- O Jogo só passa se todos os 'PRE-REQUISITOS FOREM ATENDIDOS' -----------

if(preReqs==7){

quant++;

contGeral++;

ordenaJogo();

printf("%03d - ", contGeral);

for(i=0;i<7;i++) {

printf("%02d ", jogo7Dezenas[i]);

fprintf(f,"%02d%c", jogo7Dezenas[i], 9);

}

printf("%02d\n", jogo7Dezenas[6]);

fprintf(f,"%02d\n", jogo7Dezenas[6]);

}

preReqs = 0;

it++;

}

contSeq++;

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

}

// ------------------------------------------------------------------------------------------------

// ------------------------- COMBINACOES COM APENAS (UMA) DEZENAS ---------------------------------

// ------------------------------------------------------------------------------------------------

printf("\nTOTAL DE ITERACOES NECESSARIAS: %d\n", it);

printf("INVESTIMENTO NECESSARIO: Quant: %d;R$ %1.2f\n", contGeral, (float)(contGeral\*98.0));

fclose(f);

}