#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#include<string.h>

#define r 44

#define c 32

int one(float time) {

char wan[8][7] = {

' ',' ',' ','\*',' ',' ',' ',

' ',' ','\*','\*',' ',' ',' ',

' ','\*',' ','\*',' ',' ',' ',

' ',' ',' ','\*',' ',' ',' ',

' ',' ',' ','\*',' ',' ',' ',

' ',' ',' ','\*',' ',' ',' ',

' ',' ',' ','\*',' ',' ',' ',

' ','\*','\*','\*','\*','\*',' ' };

for (int i = 0; i < 8; )

{

printf(" ");

for (int j = 0; j < 7; )

{

printf("%c", wan[i][j]); printf(" ");

if (counter\_seconds(time))

j++;

}

if (counter\_seconds(time))

i++;

printf("\n");

}

return 1;

}

int tow(float time) {

char tow[8][7] = {

' ',' ','\*','\*',' ',' ',' ',

' ','\*',' ',' ','\*',' ',' ',

' ',' ',' ',' ','\*',' ',' ',

' ',' ',' ',' ','\*',' ',' ',

' ',' ',' ','\*',' ',' ',' ',

' ',' ','\*',' ',' ',' ',' ',

' ','\*',' ',' ',' ',' ',' ',

' ','\*','\*','\*','\*',' ',' ' };

for (int i = 0; i < 8; )

{

printf(" ");

for (int j = 0; j < 7; )

{

printf("%c", tow[i][j]); printf(" ");

if (counter\_seconds(time))

j++;

}

if (counter\_seconds(time))

i++;

printf("\n");

}

return 1;

}

int three(float time) {

char three[8][7] = {

' ',' ',' ',' ',' ',' ',' ',

' ',' ','\*','\*',' ',' ',' ',

' ','\*',' ',' ','\*',' ',' ',

' ',' ',' ',' ','\*',' ',' ',

' ',' ','\*','\*',' ',' ',' ',

' ',' ',' ',' ','\*',' ',' ',

' ','\*',' ',' ','\*',' ',' ',

' ',' ','\*','\*',' ',' ',' ' };

for (int i = 0; i < 8; )

{

printf(" ");

for (int j = 0; j < 7; )

{

printf("%c", three[i][j]); printf(" ");

if (counter\_seconds(time))

j++;

}

if (counter\_seconds(time))

i++;

printf("\n");

}

return 1;

}

int delet(int size) {

for (int i = 0; i <size; i++)

{

fputs("\033[A\033[2K", stdout);

}

}

int counter\_seconds(float second)

{

double time\_spent, begin;

unsigned int i;

begin = clock();

for (i = 0; 1; i++)

{

time\_spent = (double)(clock() - begin) / CLOCKS\_PER\_SEC;

if (time\_spent >= second)

break;

}

return 1;

}

int random\_number(int numto, int numfrom) {

srand(time(NULL));

return rand() % (numto - numfrom + 1) + numfrom;

}

int start(char tract[r][c]) {

counter\_seconds(5);

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

{

delet(r - 3);

if (i == 2)

one(0);

else if (i == 1)

tow(0);

else if (i == 0)

three(0);

for (int i = 0; i < r-12; i++)

{

for (int j = 0; j < c; j++)

{

printf("%c ", tract[i][j]);

}

printf("\n");

}

counter\_seconds(2);

}

delet(r - 3);

return 1;

}

int tract\_(char tract[r][c]) {

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

{

tract[i][j] = ' ';

if (i % 8 < 4 && j == 15 || i % 8 < 4 && j == 22)

tract[i][j] = '|';

if (j == 7 || j == 9 || j == 30 || j == 28)

tract[i][j] = '|';

if (i % 3 == 0 && j == 8 || i % 3 == 0 && j == 29)

tract[i][j] = '^';

if(i%9==0&&j==5|| i % 9 == 0 && j == 31)

tract[i][j] = '\*';

}

}

return 1;

}

int print\_mat(char tract[r][c]){

delet(r+1);

for (int i = 0; i < r-1; i++)

{

for (int j = 0; j < c; j++)

{

printf("%c ", tract[i][j]);

}

printf("\n");

}

return;

}

int bubel\_tract(char tract[r][c]) {

char swap;

for (int i = r - 1; i >0 ; i--)

{

for (int j= 0; j < c; j++)

{

swap = tract[i][j];

tract[i][j] = tract[i-1][j];

tract[i-1][j] = swap;

}

}

}

int location\_car(char tract[r][c], char car[8][5],int loc) {

int i\_car = 0, j\_car = 0;

int i = 22, j = 16+loc;

if (loc<= -6)

loc = -6;

if ( loc >= 7)

loc = 7;

for (i = 22; i < 30; i++, i\_car++)

{

j\_car = 0;

for ( j = 16+ loc; j < 21+ loc; j++, j\_car++)

{

tract[i][j] = car[i\_car][j\_car];

}

}

}

int reset\_car(char tract[r][c] , int loc, char location){

for (int i = 0; i < c; i++)

{

tract[30][i] = tract[21][i];

}

for (int i = 22; i < 21 + 9; i++)

{

if (return\_location(location) == -1)

tract[i][21 + loc] = tract[i - 10][21 + loc];

if (return\_location(location) == 1)

tract[i][15 + loc] = tract[i - 10][15 + loc];

}

}

int return\_location(char location) {

if (location == 98)

return -1;

if (location == 109)

return 1;

else

return 0;

}

int chek\_location(int loc){

if (loc < -6)

return -6;

else if (loc > 7)

return 7;

else

return loc;

}

int boms(int i, char tract[r][c], int num) {

int ran;

if(i%num==0){

ran = random\_number(24, 9);

for (int i = ran; i < ran+4; i++)

{

tract[0][i] = '\*';

}

}

return 1;

}

int reset\_boms(char tract[r][c]) {

for (int i = 8; i <28; i++)

{

if (tract[r - 1][i] == '\*')

tract[r - 1][i] = tract[r - 10][i];

}

return 1;

}

int accident(char tract[r][c]) {

for (int i = 8; i <29 ; i++)

{

if ((tract[21][i] == '\*' && tract[22][i]=='\_')|| (tract[21][i] == '\*' && tract[23][i] == '/')|| (tract[21][i] == '\*' && tract[23][i] == '\\')) {

return 5;

}

}

return 1;

}

void main() {

char tract[r][c];

char car[8][5] = {

' ','\_','\_','\_',' ',

'/','^',' ','^','\\',

'|',' ','\_',' ','|',

'|','|',' ','|','|',

'|','|','^','|','|',

'|','|','\_','|','|',

'|',' ',' ',' ','|',

'|','\_','\_','\_','|',};

char location = 0;

int loc=0;

int sec = 1;

int duration = 0;

int i = 0;

float time = 0.09;

tract\_(tract);

start(tract);

while (1) {

i++;

if (i % 100 == 0)

time /= 2;

duration = 0;

while (!\_kbhit()) {

duration = (clock()) / CLOCKS\_PER\_SEC;

if (duration >= sec) {

break;

}

}

if (duration < sec) {

location = getch();

loc += return\_location(location);

}

else

boms(i, tract,18);

location\_car(tract, car, loc);

if (accident(tract) == 5)

break;

reset\_car(tract, loc, location);

print\_mat(tract);

bubel\_tract(tract);

counter\_seconds(time);

reset\_boms(tract);

}

return 1;

}