5.

```
#include <iostream>
#include <Windows.h>
using namespace std;

const int CENTER_PADDING = 5; // distance from middle to UP and DOWN

// Changes output device (in our case console) cursor position
void gotoxy(int x, int y, HANDLE outputDeviceHandle);
// Prints word in output device (console) on specified cursor position
void printWordOnConsole(HANDLE outputDeviceHandle, int x, int y, char* word);

void main ()
{
HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
CONSOLE_SCREEN_BUFFER_INFO csbi; // structure where we write console window size
	GetConsoleScreenBufferInfo(consoleHandle, &csbi); // and other information

int consoleColumns = csbi.srWindow.Right - csbi.srWindow.Left + 1; // get console length
int consoleRows = csbi.srWindow.Bottom - csbi.srWindow.Top + 1; // get console width
int middleColumn = consoleColumns / 2;
// move "UP" and "DOWN" words until they go to top and bottom
for (int i = 0, j = consoleRows - 1; j >= 0; i++, j--)
{
printWordOnConsole(consoleHandle, middleColumn - CENTER_PADDING, j, "UP");
printWordOnConsole(consoleHandle, middleColumn + CENTER_PADDING, i, "DOWN");
Sleep(1000);
system("cls");
}
}

void printWordOnConsole(HANDLE outputDeviceHandle, int x, int y, char* word)
{
gotoxy(x, y, outputDeviceHandle);
cout << word;
}

void gotoxy(int x, int y, HANDLE outputDeviceHandle)
{
COORD coord;
coord.X = x;
coord.Y = y;
SetConsoleCursorPosition(outputDeviceHandle, coord);
}
```

//
//

6.

```
#include <iostream>
#include <conio.h> // for getch()
#include <Windows.h>
using namespace std;

const int CENTER_PADDING = 5;

// Changes output device (in our case console) cursor position
void gotoxy(int x, int y, HANDLE outputDeviceHandle);
// Prints word in output device (console) on specified cursor position
void printWordOnConsole(HANDLE outputDeviceHandle, int x, int y, char* word);
// Deletes word with specified length from specified position on output device (console)
void deleteWordFromConsole(HANDLE outputDeviceHandle, int x, int y, int length);
// Thread function that prints "UP" and "DOWN"
DWORD WINAPI upAndDownPrintingLoop(LPVOID continuePrinting);
```

```
void main ()
{
cout << "Press [Enter] key to stop printing...";
bool keepPrinting = true;
HANDLE hThread = CreateThread(NULL, 0, upAndDownPrintingLoop, &keepPrinting, 0, NULL);
while (getch() != 13)
{ }
keepPrinting = false;
CloseHandle(hThread);
}

DWORD WINAPI upAndDownPrintingLoop(LPVOID continuePrinting)
{
bool* keepPrinting = (bool*)continuePrinting;
HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
CONSOLE_SCREEN_BUFFER_INFO csbi;
      GetConsoleScreenBufferInfo(consoleHandle, &csbi);

int consoleColumns = csbi.srWindow.Right - csbi.srWindow.Left + 1;
int consoleRows = csbi.srWindow.Bottom - csbi.srWindow.Top + 1;
int middleColumn = consoleColumns / 2;

while(*keepPrinting)
{
for (int i = 0, j = consoleRows - 1; j >= 0 && *keepPrinting; i++, j--)
{
printWordOnConsole(consoleHandle, middleColumn - CENTER_PADDING, j, "UP");
printWordOnConsole(consoleHandle, middleColumn + CENTER_PADDING, i, "DOWN");
Sleep(700);
deleteWordFromConsole(consoleHandle, middleColumn - CENTER_PADDING, j, 2);
deleteWordFromConsole(consoleHandle, middleColumn + CENTER_PADDING, i, 4);
}
}
return 0;
}

void printWordOnConsole(HANDLE outputDeviceHandle, int x, int y, char* word)
{
gotoxy(x, y, outputDeviceHandle);
cout << word;
}

void deleteWordFromConsole(HANDLE outputDeviceHandle, int x, int y, int length)
{
gotoxy(x, y, outputDeviceHandle);
//cout << string(length, '\b').c_str();
cout << string(length, ' ').c_str();
}

void gotoxy(int x, int y, HANDLE outputDeviceHandle)
{
COORD coord;
coord.X = x;
coord.Y = y;
SetConsoleCursorPosition(outputDeviceHandle, coord);
}
```