Lab 3 Alarm Clock

Brandon Wong and Duc Tran
9/27/2013

1.0 OBJECTIVES:

Our objective for the third lab was to design an alarm clock. The design of the alarm clock includes developing four interfaces that effectively communicate with each other. The four different modules include the switch interface, SysTick time management interface, OLED interface, and speaker interface. The graphics driver for the OLED will be in charge of displaying the time as well as drawing the lines for an analog clock. The switch interface will debounce the inputs using edge triggered interrupts. The timer and alarm will interrupt by SysTick periodic interrupts. We will be designing these modules as well as debug and test our software by using the techniques from the previous lab. Finally, our goal is to obtain a better understanding of interrupts on the LM3S1968.

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

The objectives of this project are to design, build and test an alarm clock. Educationally, students are learning how to design and test modular software and how to perform switch/keypad input in the background.

1.2. Process: How will the project be developed?

The project will be developed using the LM3S1968 board. There will be switches or a keypad. The system will be built on a solderless breadboard and run on the usual USB power. The system may use the on board switches and/or the on board sound. Alternatively, the system may include an external keypad and/or speaker. There will be at least four hardware/software modules: switch/keypad input, time management, OLED graphics, and sound output. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

Duc will be responsible for the OLED graphics, and switch/keypad input. Brandon will be responsible for the time management, and sound output.

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the LM3S1968 board, a solderless breadboard, and be powered using the USB cable.

1.5. Terminology: Define terms used in the document.

Power budget – one can estimate the operation time of a battery-powered embedded system by dividing the energy storage by the average current required to run the system.

device driver – a set of software functions that facilitate the use of an I/O port

critical section – a loss of input, corruption of data, or shifting of data after a subroutine due to shared global variables

latency – time between hardware trigger and software response

time jitter – undesired deviation from true periodicity

modular programming – a style of software development that divides the software problem into distinct and independent modules

1.6. Security: How will intellectual property be managed?

The system may include software from StellarisWare and from the book. No software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). It is the responsibility of the team to keep its EE445L lab solutions secure.

2. Function Description

2.1. Functionality: What will the system do precisely?

The clock must be able to perform five functions. 1) It will display hours and minutes in both graphical and numeric forms on the OLED. The graphical output will include the 12 numbers around a circle, the hour hand, and the minute hand. The numerical output will be easy to read. 2) It will allow the operator to set the current time using switches or a keypad. 3) It will allow the operator to set the alarm time including enabling/disabling alarms. 4) It will make a sound at the alarm time. 5) It will allow the operator to stop the sound. An LED heartbeat will show when the system is running.6) Allow user to snooze the alarm. 7) The interface will show emoticons to indicate time range (morning, lunch, afternoon, night). 8) The Alarm has welcome screen.

2.2. Scope: List the phases and what will be delivered in each phase.

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the LM3S1968 board and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the clock display should be beautiful and effective in telling time. Third, the operation of setting the time and alarm should be simple and intuitive. The system should not have critical sections. All shared global variables must be identified with documentation that a critical section does not exist. Backward jumps in the ISR should be avoided if possible. The interrupt service routine used to maintain time must complete in as short a time as possible. This means all OLED I/O occurs in the main program. The average current on the +5V power will be measured with and without the alarm sounding.

2.5. Usability: Describe the interfaces. Be quantitative if possible.

There will be four switch inputs. In the main menu, the switches can be used to activate 1) set time; 2) set alarm; 3) turn on/off alarm; and 4) display mode. In set time and alarm modes, two switches add and subtract hours and the other two add and subtract minutes. After 10 seconds of inactivity the system reverts to the main menu. The display mode switch toggles between graphical and numeric displays. The switches will be debounced, so only one action occurs when the operator touches a switch once.

The OLED display shows the time using graphical display typical of a standard on the wall clock. The 12 numbers, the minute hand, and the hour hand are large and easy to see. The clock can also display the time in numeric mode using numbers.

The alarm sound can be a simple square wave. The sound amplitude will be just loud enough for the TA to hear when within 3 feet.

2.6. Safety: Explain any safety requirements and how they will be measured.

The alarm sound will be VERY quiet in order to respect other people in the room during testing. Connecting or disconnecting wires on the protoboard while power is applied may damage the board.

3. Deliverables

3.1. Reports: How will the system be described?

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

3.2. Audits: How will the clients evaluate progress?

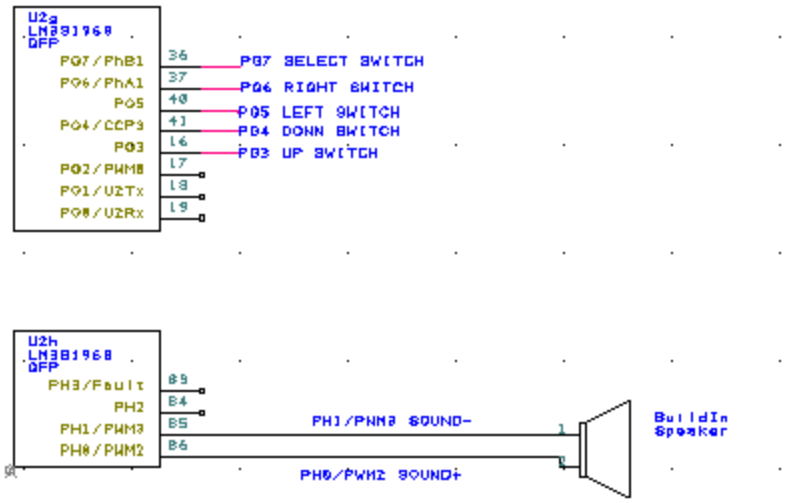
The preparation is due at the beginning of the lab period on the date listed in the syllabus.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

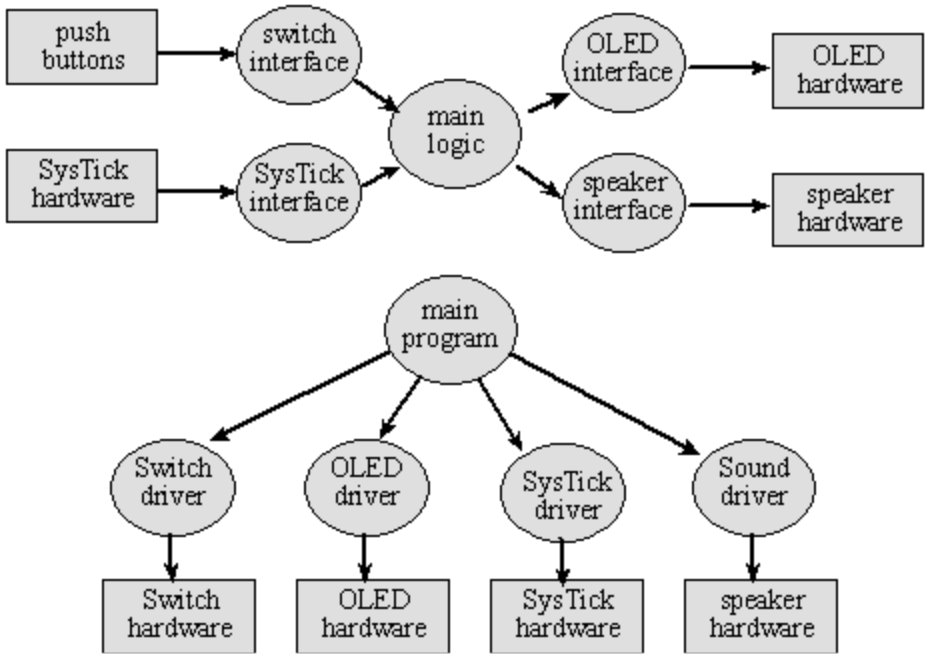
There are three deliverables: preparation, demonstration, and report.

2.0 HARDWARE DESIGN:

Used internal hardware for the four buttons and speaker.



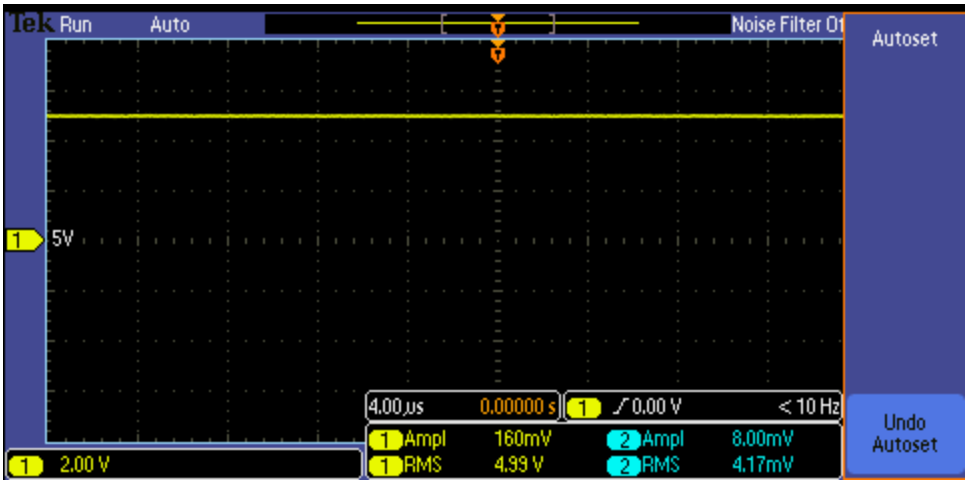
3.0 SOFTWARE DESIGN:



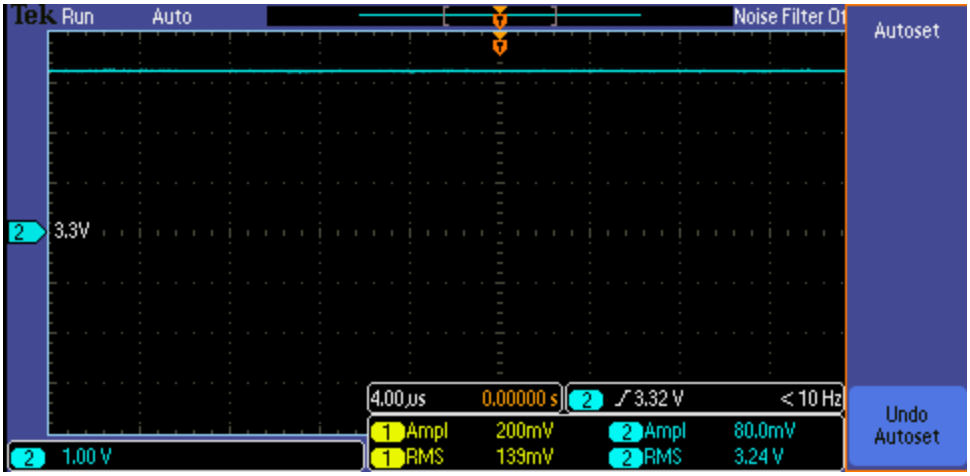
4.0 MEASUREMENT DATA:

4.1 Plot the +5 and +3.3 supply voltages versus time and record the rms magnitudes

+5 Volts: RMS = 4.99V



+3.3 Volts: RMS = 3.24V

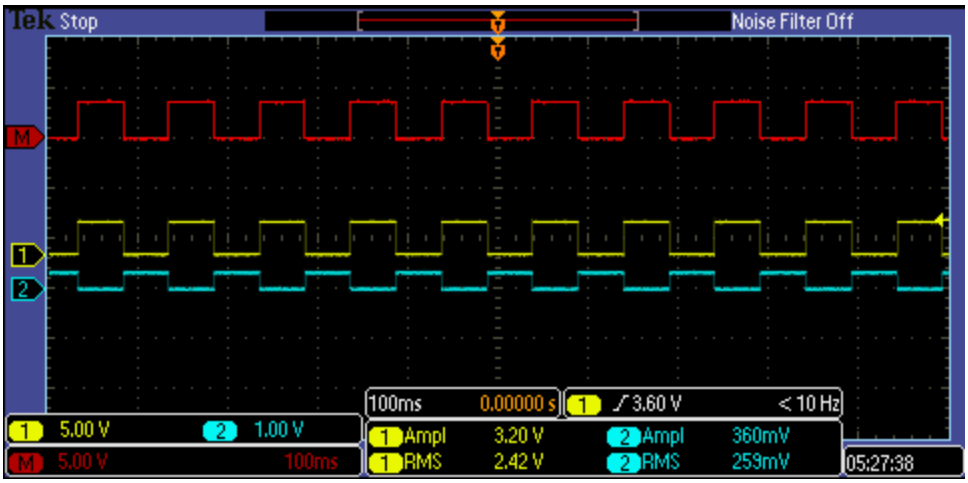


4.2 Plot the speaker voltage (or output voltage) versus time during an alarm sound

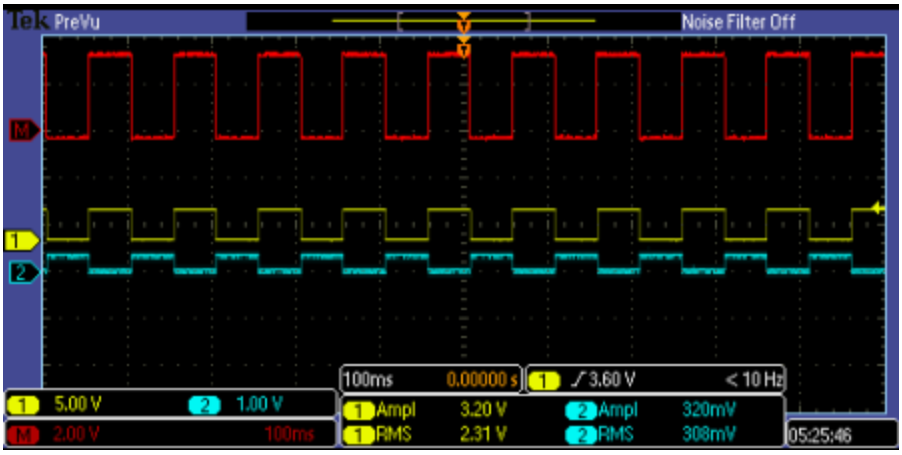
Channel 1 (Yellow) : PH1

Channel 2 (Blue) : PH0

+5 Volts:



+3.3 Volts:



4.3 Measurements of current required to run the alarm clock, with and without the alarm

Supply Voltage	**Current with Alarm**	**Current without Alarm**
5V	.300 Amp	.305 Amp
3.3V	.306 Amp	.311 Amp

5.0 ANALYSIS AND DISCUSSION:

1) Give two ways to remove a critical section.

You can remove this by avoiding the use of global variables for ISRs and also by first disabling an interrupt when entering an ISR and disabling before returning from an ISR.

2) How long does it take to update the OLED with a new time?

The time that it takes for the OLED to update depends on where the PC is at during the while loop. So in theory, the longest time it would take to display the time onto the OLED is however long it would take to run the infinite while loop just once.

3) What would be the disadvantage of updating the OLED in the background ISR?

By updating it during the background ISR, it would lead into critical sections if you are using a lot of clock cycles to display time to OLED. If you try to disabling/enabling interrupts in the ISR, then you are not properly managing the time.

4) Did you use a OLED clear function? If so, how could you have redesigned the OLED update to run much faster?

We used an OLED clear function that we designed ourselves by instead of clearing the entire

display, it just cleared the line functions for displaying the analog clock. This saved time because the only thing you needed to clear on the analog display were the lines.

5) Assuming the system were battery powered, list three ways you could have saved power.

Include a power saving mode for the OLED display (put the display to sleep when not needed). Write software that can shutdown output pins or reduce current to pins when not needed. Include a volume/mute button to turn off the speaker, since we observed that in part 4.3 the current increased with alarm sound.

6.0 CODE

```
//Filename: LogoTestMain.c
//Author: Duc Tran, Brandon Wong
//Initial Creation Date: September 13, 2013
//Description: Main logic for digital alarm clock
//Lab Number: M W 2-3:30
//TA : Omar & Mahesh
//Date of last revision :September 27, 2013
//Hardware Configuration : NONE

// LogoTestMain.c
// Runs on LM3S1968 or LM3S8962
// Jonathan Valvano
// November 12, 2012

/* This example accompanies the books
   "Embedded Systems: Real Time Interfacing to Arm Cortex M Microcontrollers",
   ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2012

   "Embedded Systems: Introduction to Arm Cortex M Microcontrollers",
   ISBN: 978-1469998749, Jonathan Valvano, copyright (c) 2012
 Copyright 2012 by Jonathan W. Valvano, valvano@mail.utexas.edu
        You may use, edit, run or distribute this file
        as long as the above copyright notice remains
 THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS,
IMPLIED
 OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
```

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.
VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
For more information about my classes, my research, and my books, see
http://users.ece.utexas.edu/~valvano/
*/

```
#include "hw_types.h"
#include "sysctl.h"
#include "lm3s1968.h"
#include "rit128x96x4.h"
#include "logo.h"
#include "PLL.h"
#include "SysTickInts.h"
#include <stdio.h>
#include "SwitchModule.h"
#include "SoundModule.h"
#include <math.h>
#define SYSCTL_RCGC2_R          (*((volatile unsigned long *)0x400FE108))
#define SYSCTL_RCGC2_GPIOG   0x00000040  // port D Clock Gating Control
#define GPIO_PORTG_DIR_R        (*((volatile unsigned long *)0x40026400))
#define GPIO_PORTG_DEN_R        (*((volatile unsigned long *)0x4002651C))
#define PI 3.14159265

void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
long StartCritical (void);        // previous I bit, disable interrupts
void EndCritical(long sr);       // restore I bit to previous value
void WaitForInterrupt(void);  // low power mode
char* Convert(unsigned int msec);
void D2A(unsigned int msec);
char* DigitalRead(unsigned int msec);
void D2AMin(unsigned int msec);
void D2ASec(unsigned int msec);
void D2AHr(unsigned int msec);
#define PG1    (*((volatile unsigned long *)0x40026008))
```

```
// which delays 3*ulCount cycles
#ifdef __TI_COMPILER_VERSION__
   //Code Composer Studio Code
   void Delay(unsigned long ulCount){
   __asm (    "  subs     r0, #1\n"
                    "        bne      Delay\n"
                    "        bx       lr\n");
}

#else
   //Keil uVision Code
   __asm void
   Delay(unsigned long ulCount)
   {
         subs    r0, #1
         bne     Delay
         bx      lr
   }

#endif
const unsigned char Horse[] ={
 0x42, 0x4D, 0x76, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00,
0x80,
 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80,
0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00,
0xFF,
 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x07,
 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00,
0x00, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x70,
0x00,
 0x0F, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xF8,
0x00, 0x00, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8F,
0x00,
```

0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F,
0x70, 0x00, 0x7F, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
0xFF,
0x00, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0x8F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x08, 0xFF,
0xFF, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F,
0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xF8, 0xFF,
0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xF0, 0x00, 0x00, 0x00, 0x07, 0x87,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x07, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0x80, 0x77, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x70, 0x00, 0x08, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x8F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x77, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00, 0x00, 0x00, 0xF0, 0x7F, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xF7, 0x08, 0xFF, 0xF7, 0x00, 0x7F, 0x07, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
0xFF, 0x00, 0x07, 0xF8, 0x87, 0x00, 0x7F, 0x77, 0xFF, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xF8, 0x70, 0x00, 0x08, 0xFF, 0xF8, 0x00, 0x0F, 0x80, 0x07, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF,
0xF7, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0xF0, 0x00, 0x0F, 0x80, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0x00, 0x08, 0xF0, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7,
0x00, 0x00,
0x00, 0x00, 0x7F, 0xFF, 0xF8, 0x00, 0x00, 0x8F, 0xF8, 0x08, 0xFF, 0xFF, 0xFF, 0x8F, 0x80,
0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0xFF, 0xFF, 0xFF, 0x8F,
0xF8, 0x70,
0x00, 0x00, 0x07, 0x78, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0xF8, 0xF7,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xF8,
0xF8, 0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x8F, 0xFF, 0xF8, 0xFF,
0x87, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xF8, 0xF8, 0xFF, 0xF8,
0xF8, 0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x8F, 0xFF,
0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0x8F,
0xF0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xFF, 0xF8, 0xFF, 0xF8,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8F, 0x88, 0xFF, 0x00,
0x00,

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x77, 0x70, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,

};
const unsigned char ECE[] ={
 0x42, 0x4D, 0xF6, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x22, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x80, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x80,
 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80,
0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00,
0xFF,
 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xC0, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0x9E,
0x9E,
 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE,
0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E,
```

0xDE, 0x9E,
0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x00, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xE0, 0x00, 0xCB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB,
0xFB, 0xFF, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xEE, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00,
0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFB, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xF9, 0xEE, 0xEE, 0xE0, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xE0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE9, 0xE0, 0x00, 0x00, 0x00, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x0C, 0xFF, 0xFF,
0xFF, 0xFF, 0xDC, 0x00, 0x00, 0x9E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xCE, 0xEE, 0xEC, 0xDF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x0E, 0xDF, 0xFE,
0xDF, 0xDE, 0xDF, 0xF0, 0x00, 0x00, 0x0E, 0xFF, 0xDF, 0x9E, 0xD0, 0x00, 0x00, 0x9F, 0xFF, 0xDE, 0xDF, 0xDF, 0xF0, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0xF9, 0x00, 0xEF, 0xFF, 0xF0, 0xEF, 0xFF,
0xFF, 0xFF, 0xFF, 0xE0, 0x09, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xCE, 0xEE, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xA0, 0x00, 0xE0, 0x00, 0xAF, 0xF0,
0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xFF, 0xF0, 0x00, 0x00, 0xEB, 0x00, 0x00, 0x0F, 0xFB, 0x00, 0x00, 0x00, 0xFB, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0x00, 0xFF,
0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0x0E, 0xEE, 0xC0, 0x9E, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0x90, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x00, 0x9E, 0x00, 0x0E, 0xFE, 0x00, 0x00, 0x00, 0x00, 0xD0, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0xE0, 0xCF, 0xFF, 0xFF, 0xFF,

0xF0, 0xFF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF9, 0xAE, 0xEE, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0xCF, 0xF0,
0x00, 0x00, 0x00, 0x0B, 0xC0, 0xEF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xA0, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x0E, 0xEE, 0xEC, 0x9F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x9F, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0xDF, 0xD0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0xA0, 0x00, 0x00, 0x00, 0xA0, 0xEE, 0xEE, 0xEA, 0xA0, 0x00, 0x00, 0x00, 0xA0, 0x00, 0x00, 0x00, 0xE0, 0x00, 0xAF, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0x9F, 0xFF, 0xFF, 0xFF, 0xF0, 0xDF,
0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x9E, 0x90, 0x9A, 0x90, 0x9E, 0x9C, 0x00, 0xEE, 0xEE, 0xEE, 0x00, 0x9E, 0x9E, 0x90, 0x9E, 0x90, 0x9E, 0x9E, 0x90, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x0E, 0x00, 0x0E, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xF0, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xEF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFF, 0xC0, 0xEE, 0xEE, 0xEE, 0xE0, 0xF9, 0xFB, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0xCF, 0xF0,
0xC0, 0xA0, 0xF9, 0x00, 0x09, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF9, 0x00, 0xC0, 0xEF, 0xC0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xF0, 0x9F,
0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x00, 0x00, 0x0E, 0xEE, 0xEE, 0xEE, 0xD0, 0x0C, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x9F, 0xFE,
0xDE, 0x9E, 0xF0, 0x00, 0x00, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xFF, 0x9E, 0xDE, 0xFE, 0x90, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xEF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xEE, 0xEA, 0xEE, 0xEE, 0xEE, 0xEE, 0xEA, 0xEE, 0xE0, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0x00,

0xAF, 0xF0,
0x00, 0x00, 0xE0, 0x00, 0x00, 0xFF, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xF0,
0x00, 0x00, 0x0F, 0xA0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF,
0xF0, 0xDF,
0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x0E,
0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00,
0x0F, 0xF0,
0x00, 0x00, 0x0A, 0x00, 0x00, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xF0, 0xEF,
0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0xE9, 0xE0, 0xE9, 0x00, 0x00,
0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xE0, 0x00, 0xE9, 0xE0, 0xE9, 0xE0, 0xE0, 0x00,
0xCF, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0xEF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF,
0xF0, 0x9F,
0xFE, 0xFF, 0xFF, 0xF0, 0x0F, 0xFF, 0xFF, 0x9F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x0C, 0xEE,
0xCE, 0xDE, 0xEE, 0xEE, 0xEE, 0xE0, 0x0E, 0xEE, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0x90, 0x00,
0x9F, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x0E, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xF0, 0xFF,
0xF9, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xEF, 0xFF, 0xFF, 0xFF, 0xFB, 0x00, 0xEE, 0xE0,
0x00, 0xA0, 0x00, 0xA0, 0x00, 0x00, 0x00, 0x0A, 0xEA, 0x00, 0xFF, 0xFF, 0xFF, 0xE0, 0x00,
0xAF, 0xF0,
0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xFB, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x0B, 0xF0,
0x00, 0x00, 0x00, 0xE0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0x9F, 0xFF, 0xFF, 0xFF,
0xF0, 0xDF,
0xFE, 0x0F, 0xFF, 0xD0, 0x0E, 0xFF, 0xF0, 0x9F, 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xCA, 0x0E,
0xFF, 0xFE, 0x9E, 0x9E, 0x9E, 0xFF, 0xFF, 0x90, 0x0E, 0xE0, 0x0A, 0xFF, 0xFF, 0x90, 0x00,
0x9F, 0xF0,
0x00, 0x00, 0x0E, 0xD0, 0x00, 0x00, 0x9F, 0xD0, 0x00, 0x00, 0x9F, 0xF0, 0x00, 0x0F, 0xFE,
0x00, 0x00, 0x00, 0xF0, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xF0, 0x00, 0x09, 0xFF, 0xFF, 0xFF,
0xC0, 0x00,
0xF9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xEF, 0xFF, 0xE0, 0xE0, 0x0A, 0xE0, 0x00, 0xE9,
0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0x00, 0xEA, 0xE0, 0x00, 0xE9, 0xC0, 0x09,
0xFB, 0xFF,
0xFB, 0xFF, 0xFB, 0xE0, 0x00, 0x00, 0x00, 0xEF, 0xFB, 0xEF, 0xF0, 0xF0, 0x09, 0xEF, 0xFB,
0xFB, 0xFB, 0xFF, 0xF0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xDE, 0xDE, 0xDF, 0xFF, 0xFF, 0xFE,

0xDE, 0xDE,
0xDF, 0xDE, 0xDF, 0xDE, 0xDE, 0xDE, 0xDF, 0xFF, 0xF0, 0x00, 0x00, 0xEE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xCA, 0xE0, 0xC0, 0xEF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0xA0, 0xC0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x90, 0x9E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xDE, 0xDE, 0x90, 0x90, 0x9A, 0x90,
0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9F, 0x00, 0x00, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0,
0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xF9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xF9, 0xEB, 0xF9, 0xEB,
0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x90, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00,
0x00, 0x00,
0x00, 0x00,
0x00, 0x00,
0x00, 0x00,

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,

};

const unsigned char clock1[] ={
 0x42, 0x4D, 0x76, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x20, 0x2E, 0x00, 0x00, 0x20, 0x2E, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00,
0x80,
 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80,
0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00,
0xFF,
 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xB7, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B,
0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x78, 0x8B, 0xBB, 0xBB, 0xBB, 0xBB, 0xB8, 0x87, 0x00, 0x00, 0x00, 0x00, 0x0B,
0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB,
0xB0, 0x00,
 0x07, 0x8B, 0xBB, 0xB7, 0x00, 0x0B, 0xB0, 0x00, 0x7B, 0xBB, 0xB8, 0x80, 0x00, 0x00, 0xBB,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB,
0xBB, 0x07,
 0x8B, 0xB7, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x0B, 0xBF, 0x80, 0x0B, 0xB0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB,
```

0x8B,
0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBF, 0x8B, 0xB0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8B,
0xB0,
0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBB, 0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xBB,
0xBB,
0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBB, 0xF8,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFB, 0x00,
0x7B,
0x0F, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB, 0x00, 0xB8,
0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xB0,
0x00, 0x00,
0x00, 0xF0, 0x00, 0x00, 0x0F, 0xF0, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x0B, 0x00, 0x0B,
0xBF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xB7, 0x00,
0x00, 0x00,
0x00, 0xF0, 0x00, 0x00, 0x0F, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0xF0, 0x0F, 0xF0, 0x00, 0x00,
0x0B, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0x70, 0x00,
0x00, 0x00,
0x00, 0x0F, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00,
0x00, 0xBF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xB7, 0x00, 0x00,
0x00, 0x00,
0x00, 0x0F, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00,
0x00, 0x0B, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xB0, 0x00, 0x00,
0x00, 0x00,
0xF0, 0x00, 0xF0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xF0, 0x00, 0x00,
0x00, 0x0B, 0x88, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0xBB, 0xB0, 0x8B, 0x00, 0x00, 0x00,
0x00, 0x00,
0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0xBF, 0x70, 0xBB, 0xB0, 0x00, 0x00, 0x00, 0xBB, 0xBB, 0xB0, 0x00, 0x00, 0x00,
0x00, 0x00,
0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0B, 0xBB, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xBB, 0x00, 0xFF, 0xF0,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00,
0x0F, 0xF0, 0x0B, 0xBB, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8B, 0x8B, 0x0F, 0x00, 0xFF,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0xF0, 0x0B, 0xBF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xB7, 0x07, 0x0F, 0x00, 0xFF,

0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F,
0xFF, 0xFF, 0xFB, 0x0B, 0x80, 0x00, 0x00, 0x00, 0x00, 0x07, 0xB0, 0x00, 0x00, 0xF0, 0xFF,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F,
0x0F, 0xF0, 0x00, 0x0B, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x70, 0x00, 0x00, 0xFF, 0xFF,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xF0, 0x00, 0x00, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x7B, 0x00, 0x00, 0x0F, 0xFF, 0xF0,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xF0, 0x00, 0x00, 0xBF, 0x00, 0x00, 0x00, 0x00, 0xB8, 0x00, 0x00, 0x0F, 0xF0, 0x0F,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0xF0, 0x00, 0x00, 0x0B, 0x70, 0x00, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x00, 0xFF, 0xF0,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0xF0, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x07, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0B, 0xF0, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xB0, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0xFF, 0xF0, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x0F, 0xF0, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0x00, 0xFF, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x0B, 0x70, 0x00, 0x0F, 0xFF, 0xF0, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xFF, 0x00, 0x00, 0xB7, 0x00, 0xBB, 0xBB, 0xBB, 0xB0, 0x0F, 0x0F, 0xF0, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x08, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xFF, 0x0B, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0xB0, 0xFF, 0x0F, 0xF0, 0x00, 0x00,
0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x08, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xF0, 0x0B, 0xBB, 0xBB, 0xBB, 0x00, 0x0B, 0x70, 0x00, 0xFF, 0x0F, 0xF0, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x0F, 0x00, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x0F, 0xFF, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0x0F, 0xF0, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xF0, 0x00, 0x00, 0xB7, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xB0, 0x00, 0x00, 0x07, 0xB0, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0xF0,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0xB0, 0x00, 0x0F, 0xF0, 0x0F, 0xF0, 0xF0,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0xF0, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0xB8, 0x00, 0x0F, 0xF0, 0x0F, 0xF0,
0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F,
0x00, 0xF0, 0x00, 0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0xB8, 0x00, 0x0F, 0xF0, 0x0F, 0xF0,
0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xF0, 0x00, 0x00, 0x00, 0x0B, 0x70, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0F, 0xF0, 0x0F, 0xF0,
0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0x00, 0x00, 0x00, 0xBF, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x70, 0x0F, 0xF0, 0x0F, 0xF0,
0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0F, 0xF0, 0x00, 0x00, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x07, 0xB0, 0xFF, 0xF0, 0x00, 0xF0,
0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0,
0x0F, 0xF0, 0xBB, 0x0B, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xB0, 0x0F, 0xF0, 0x00, 0xFF,
0xF0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0,
0x0F, 0xF0, 0xBB, 0xBB, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xB7, 0x00, 0x00, 0x00,
0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F,
0xFF, 0x00, 0x07, 0xBB, 0xBB, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBB, 0xB0, 0x00, 0x00, 0x00,
0xFF, 0xFF,
0x0F, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0B, 0x88, 0xBB, 0xB0, 0x00, 0x00, 0x00, 0xBB, 0xB7, 0xB0, 0x00, 0x00, 0x00,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xF0, 0x00, 0x00,
0x00, 0x00, 0x0B, 0xF0, 0x78, 0xBB, 0x00, 0x00, 0x00, 0xBB, 0x00, 0x8B, 0x00, 0x00, 0x00,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0xBF, 0x70, 0x00, 0xBB, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xB0, 0x00, 0x00,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x0F, 0xFF, 0xF0, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x0B, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xB7, 0x00, 0x00,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x0F, 0x00, 0xF0, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x0B, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0x70, 0x00,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x00, 0xBF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xB7, 0x00,
0xFF, 0xF0,
0x0F, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x0B, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xB0,
0x0F, 0xF0,
0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x0B,
0xB8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFB,
0x00, 0xBB,
0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xF0, 0x0F, 0xF0, 0x00, 0x00, 0xFF, 0x0B, 0x00, 0xBF,
0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xBB,
0xB0,
0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00, 0xF0, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x0B, 0xBB, 0xF8,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB,
0xB0,
0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBB, 0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB,
0x8B,
0xB7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xBF, 0x8B, 0xB0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB7,
0x00,

```
 0x8B, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7B, 0xBF, 0x80, 0x0B, 0xBB,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB, 0x70,
0x00,
 0x00, 0x7B, 0xBB, 0xB7, 0x77, 0x0B, 0xB0, 0x07, 0x7B, 0xBB, 0xB8, 0x70, 0x00, 0x00, 0xBB,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBB, 0x00,
0x00,
 0x00, 0x00, 0x07, 0x8B, 0xBB, 0xBB, 0xBB, 0xBB, 0xB8, 0x87, 0x00, 0x00, 0x00, 0x00, 0xBB,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xB0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,

};



const unsigned char control[] ={
 0x42, 0x4D, 0x76, 0x14, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x14, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x80,
 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80,
0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00,
0xFF,
 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,

};


//x2,y2 positions for sec and min
static const unsigned int linetable[60][2] =
{
    {32, 20},
    {35, 20},
    {38, 21},
    {41, 21},
    {43, 22},
  {46, 24},
  {48, 25},
  {51, 27},
  {53, 29},
  {55, 32},
  {56, 34},
    {58, 37},
```

{59, 39},
{59, 42},
{60, 45},
{60, 48},
{60, 51},
{59, 54},
{59, 57},
{58, 59},
{56, 62},
{55, 64},
{53, 67},
{51, 69},
{48, 71},
{46, 72},
{43, 74},
{41, 75},
{38, 75},
{35, 76},
{32, 76},
{29, 76},
{26, 75},
{23, 75},
{21, 74},
{18, 72},
{16, 71},
{13, 69},
{11, 67},
{9, 64},
{8, 62},
{6, 59},
{5, 56},
{5, 54},
{4, 51},
{4, 48},
{4, 45},
{5, 42},
{5, 39},
{6, 37},
{8, 34},

```
    {9, 32},
    {11, 29},
    {13, 27},
    {16, 25},
    {18, 24},
    {21, 22},
    {23, 21},
    {26, 21},
    {29, 20}
};

//x2 and y2 position for hour
static const unsigned int hourtable[48][2] =
{
    {32, 35}, //0
    {34, 35}, //1
    {35, 35}, //2
    {37, 36}, //3
    {39, 37}, //4
    {40, 38}, //5
    {41, 39}, //6
    {42, 40}, //7
    {43, 42}, //8
    {44, 43}, //9
    {45, 45}, //10
    {45, 46}, //11
    {45, 48}, //12
    {45, 50}, //13
    {45, 51}, //14
    {44, 53}, //15
    {43, 55}, //16
    {42, 56}, //17
    {41, 57}, //18
    {40, 58}, //19
    {39, 59}, //20
    {37, 60}, //21
    {35, 61}, //22
    {34, 61}, //23
    {32, 61}, //24
```

```
    {30, 61}, //25
    {29, 61}, //26
    {27, 60}, //27
    {26, 59}, //28
    {24, 58}, //29
    {23, 57}, //30
    {22, 56}, //31
    {21, 55}, //32
    {20, 53}, //33
    {19, 51}, //34
    {19, 50}, //35
    {19, 48}, //36
    {19, 46}, //37
    {19, 45}, //38
    {20, 43}, //39
    {21, 42}, //40
    {22, 40}, //41
    {23, 39}, //42
    {24, 38}, //43
    {25, 37}, //44
    {27, 36}, //45
    {29, 35}, //46
    {30, 35} //47
};

//badass
const unsigned char db[] ={
 0x42, 0x4D, 0xF6, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x6A, 0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
 0x00, 0x00, 0x80, 0x11, 0x00, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00,
0x80,
 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80,
0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00,
0xFF,
 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x04, 0x01, 0x37, 0x77, 0x77,
0x74, 0x44,
```

0x00, 0x04, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x11, 0x00, 0x17, 0x77, 0x88, 0x88, 0x88,
0x88, 0x88, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x77, 0x77, 0x74, 0x40, 0x00, 0x00, 0x01, 0x45, 0x77, 0x77, 0x77, 0x77, 0x77,
0x77, 0x77, 0x00, 0x07, 0x77, 0x88, 0x88, 0x88, 0x88, 0x88, 0x70, 0x00, 0x00, 0x00, 0x05, 0x00,
0x88, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x57, 0x77, 0x77, 0x44, 0x00, 0x00,
0x00, 0x54, 0x04, 0x00, 0x04, 0x77, 0x77, 0x77, 0x77, 0x77, 0x70, 0x07, 0x77, 0x78, 0x88, 0x88,
0x88, 0x87, 0x70, 0x00, 0x00, 0x00, 0x07, 0x00, 0xF8, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x55, 0x13, 0x77, 0x44, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x74, 0x47, 0x77, 0x77,
0x77, 0x77, 0x77, 0x07, 0x77, 0x77, 0x77, 0x78, 0x88, 0x88, 0x70, 0x00, 0x00, 0x00, 0x07, 0x00,
0x8F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x77, 0x74, 0x00, 0x00, 0x01,
0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x54, 0x77, 0x77, 0x77, 0x77, 0x77, 0x50, 0x77, 0x77, 0x78, 0x77,
0x77, 0x77, 0x70, 0x00, 0x00, 0x00, 0x77, 0x00, 0x4F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x01, 0x10, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x11, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x37, 0x33, 0x11, 0x10, 0x00, 0x55, 0x77, 0x77,
0x77, 0x77, 0x77, 0x70, 0x77, 0x77, 0x77, 0x77, 0x87, 0x77, 0x70, 0x00, 0x00, 0x00, 0x70, 0x00,
0x07, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x17, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77,
0x77, 0x77, 0x77, 0x77, 0x50, 0x01, 0x47, 0x77, 0x77, 0x78, 0x77, 0x77, 0x47, 0x77, 0x77, 0x77,
0x77, 0x77, 0x77, 0x70, 0x00, 0x00, 0x70, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00,
0x10,
0x01, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x50, 0x77,
0x77, 0x77, 0x77, 0x77, 0x67, 0x06, 0x77, 0x77, 0x77, 0x77, 0x77, 0x00, 0x00, 0x07, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x11, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x40, 0x03, 0x77, 0x77, 0x77, 0x77, 0x77, 0x74, 0x37,
0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x74, 0x07, 0x77, 0x77, 0x77, 0x77, 0x74, 0x77, 0x04, 0x77,
0x77, 0x77, 0x77, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00,
0x17, 0x77, 0x77, 0x77, 0x77, 0x77, 0x74, 0x15, 0x77, 0x40, 0x47, 0x77, 0x77, 0x77, 0x77, 0x40,
0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x87, 0x07, 0x77, 0x77, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x01, 0x37, 0x77, 0x77, 0x76, 0x44, 0x00, 0x00, 0x77,
0x74, 0x00, 0x00, 0x04, 0x67, 0x77, 0x77, 0x70, 0x77, 0x77, 0x77, 0x77, 0x76, 0x77, 0x68, 0x87,

0x78, 0x77, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11,
0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x01,
0x77, 0x77, 0x77, 0x74, 0x00, 0x00, 0x00, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00, 0x47, 0x77, 0x77,
0x07, 0x77, 0x77, 0x77, 0x75, 0x77, 0x77, 0x87, 0x77, 0x77, 0x76, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x01, 0x01, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x11, 0x77, 0x77, 0x77, 0x77, 0x40, 0x00, 0x00, 0x00, 0x04,
0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x77, 0x77, 0x04, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77, 0x77,
0x78, 0x77, 0x74, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x10, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x35, 0x17,
0x77, 0x77, 0x74, 0x74, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x71, 0x00, 0x74, 0x07, 0x77,
0x74, 0x48, 0x77, 0x77, 0x77, 0x77, 0x78, 0x77, 0x87, 0x77, 0x74, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x01, 0x11, 0x11, 0x01, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x13, 0x77, 0x77, 0x76, 0x00, 0x44, 0x00, 0x00, 0x00, 0x00,
0x01, 0x11, 0x33, 0x11, 0x14, 0x07, 0x40, 0x77, 0x77, 0x07, 0x87, 0x77, 0x77, 0x77, 0x77, 0x77,
0x87, 0x77, 0x70, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x10, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77,
0x77, 0x74, 0x47, 0x74, 0x00, 0x00, 0x00, 0x00, 0x11, 0x33, 0x33, 0x33, 0x11, 0x75, 0x74, 0x77,
0x77, 0x70, 0x77, 0x77, 0x77, 0x77, 0x47, 0x87, 0x78, 0x77, 0x70, 0x00, 0x0F, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x11, 0x11, 0x11, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x01, 0x77, 0x77, 0x50, 0x77, 0x77, 0x70, 0x00, 0x00, 0x00,
0x13, 0x33, 0x33, 0x33, 0x31, 0x54, 0x07, 0x04, 0x77, 0x77, 0x47, 0x77, 0x77, 0x77, 0x87, 0x78,
0x77, 0x77, 0x70, 0x00, 0xFF, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x10, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77,
0x77, 0x77, 0x57, 0x04, 0x00, 0x00, 0x00, 0x11, 0x31, 0x33, 0x33, 0x33, 0x11, 0x00, 0x70,
0x77, 0x77, 0x47, 0x77, 0x77, 0x87, 0x78, 0x77, 0x77, 0x77, 0x40, 0x3F, 0xFF, 0x10, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11,
0x11,
0x00, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x47, 0x37, 0x55, 0x00, 0x00, 0x01,
0x11, 0x11, 0x13, 0x33, 0x33, 0x31, 0x00, 0x76, 0x07, 0x77, 0x74, 0x87, 0x78, 0x88, 0x78, 0x87,
0x87, 0x77, 0x07, 0xF8, 0x8F, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x11, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x13, 0x31, 0x11, 0x10, 0x00, 0x00, 0x00, 0x77,
0x77,
0x74, 0x57, 0x74, 0x04, 0x54, 0x00, 0x00, 0x01, 0x11, 0x11, 0x31, 0x33, 0x33, 0x33, 0x04, 0x07,
0x07, 0x77, 0x77, 0x77, 0x77, 0x78, 0x78, 0x88, 0x78, 0x77, 0xFF, 0xF8, 0x8F, 0x31, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x13,
0x39, 0x93, 0x33, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x74, 0x15, 0x40, 0x00, 0x11,
0x31, 0x11, 0x11, 0x11, 0x33, 0x33, 0x10, 0x07, 0x57, 0x77, 0x77, 0x47, 0x78, 0x88, 0x87, 0x88,
0x87, 0x78, 0xF8, 0x88, 0x8F, 0x71, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11,
0x11, 0x10, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x13, 0x33, 0x99, 0x00, 0x00, 0x00, 0x77, 0x77,

0x77, 0x77, 0x77, 0x77, 0x76, 0x40, 0x00, 0x13, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x10, 0x40,
0x07, 0x67, 0x77, 0x77, 0x78, 0x88, 0x77, 0x88, 0x88, 0x78, 0xF8, 0xFF, 0xFF, 0x71, 0x10,
0x00, 0x10, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x00, 0x11, 0x11, 0x00, 0x00, 0x00, 0x11, 0x11,
0x11,
0x11, 0x33, 0x33, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x71, 0x31, 0x00, 0x00, 0x11,
0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x07, 0x67, 0x77, 0x77, 0x78, 0x78, 0x88, 0x88,
0x87, 0x8F, 0xFF, 0xFF, 0xFF, 0x81, 0x10, 0x00, 0x10, 0x01, 0x00, 0x10, 0x11, 0x11, 0x10,
0x00, 0x01, 0x13, 0x10, 0x03, 0x10, 0x00, 0x01, 0x11, 0x11, 0x11, 0x33, 0x00, 0x00, 0x00, 0x77,
0x77,
0x77, 0x77, 0x77, 0x77, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00,
0x00, 0x74, 0x77, 0x87, 0x77, 0x77, 0x78, 0x78, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xF1, 0x11,
0x01,
0x00, 0x11, 0x11, 0x10, 0x11, 0x10, 0x00, 0x00, 0x00, 0x11, 0x31, 0x00, 0x00, 0x10, 0x00, 0x11,
0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x74, 0x00, 0x00, 0x00, 0x11,
0x11, 0x11, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x77, 0x78, 0x77, 0x77, 0x88, 0x78, 0x78,
0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x11, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x11, 0x13, 0x33, 0x10, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77,
0x77, 0x77, 0x77, 0x77, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00,
0x00, 0x77, 0x77, 0x88, 0x74, 0x47, 0x87, 0x87, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x11,
0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x77, 0x93, 0x31, 0x00,
0x01,
0x33, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x73, 0x10, 0x00, 0x00, 0x11,
0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x06, 0x77, 0x78, 0x87, 0x78, 0x77, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x71, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x11, 0x11, 0x37, 0x73, 0xB3, 0x30, 0x01, 0x39, 0x93, 0x11, 0x00, 0x00, 0x00, 0x77,
0x77,
0x77, 0x77, 0x77, 0x50, 0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x10, 0x00,
0x00, 0x04, 0x77, 0x78, 0x88, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x71,
0x11,
0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x11, 0x13, 0x78, 0x33, 0xB1,
0x00, 0x39, 0x99, 0x33, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x44, 0x00, 0x00, 0x01,
0x11,
0x11, 0x10, 0x00, 0x00, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x7F, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF3, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00,
0x00, 0x00,
0x00, 0x11, 0x11, 0x11, 0x77, 0x73, 0x33, 0x3B, 0x39, 0x99, 0x93, 0x00, 0x00, 0x00, 0x77,
0x77, 0x77, 0x77, 0x77, 0x40, 0x00, 0x00, 0x01, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x01, 0x11,
0x00,

0x00, 0x07, 0x77, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x37, 0x73, 0x33, 0xB3,
0x39, 0x99, 0xB9, 0x00, 0x00, 0x00, 0x77, 0x77, 0x77, 0x77, 0x77, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x10, 0x11, 0x11, 0x11, 0x11, 0x11, 0x10, 0x00, 0x04, 0x77, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x13, 0x37, 0x87, 0x33, 0x33, 0x39, 0x99, 0x33, 0x00, 0x00, 0x00, 0x88, 0x77,
0x77, 0x77, 0x77, 0x70, 0x00, 0x00, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x33, 0x33, 0x11, 0x10, 0x00, 0x07, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xF8, 0x73, 0x10, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x11, 0x11, 0x33, 0x37, 0x87, 0x13, 0x33, 0xB9, 0x99, 0x9B, 0x00, 0x00, 0x00, 0x88, 0x87, 0x77, 0x77, 0x77, 0x70, 0x00, 0x00, 0x01, 0x11,
0x11, 0x13, 0x33, 0x11, 0x11, 0x13, 0x31, 0x11, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x30, 0x00, 0x00, 0x10, 0x13, 0x33,
0x33, 0x11, 0x11, 0x33, 0x77, 0x87, 0x13, 0x33, 0xB9, 0x99, 0x33, 0x00, 0x00, 0x00, 0x88, 0x77, 0x77, 0x77, 0x77, 0x75, 0x00, 0x00, 0x01, 0x11, 0x11, 0x33, 0x11, 0x11, 0x11, 0x13, 0x31, 0x11,
0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x13, 0x33, 0x31, 0x11, 0x33, 0x78, 0x87, 0x11, 0x3B,
0x33, 0x99, 0x93, 0x00, 0x00, 0x00, 0x88, 0x88, 0x77, 0x77, 0x77, 0x74, 0x00, 0x00, 0x01, 0x31, 0x13, 0x33, 0x11, 0x11, 0x11, 0x11, 0x33, 0x11, 0x11, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x01, 0x11, 0x11, 0x13, 0x11, 0x11, 0x11, 0x33, 0x78, 0x83, 0x11, 0x1B, 0x31, 0x99, 0x33, 0x00, 0x00, 0x00, 0x77, 0x77,
0x77, 0x77, 0x77, 0x70, 0x00, 0x00, 0x01, 0x31, 0x13, 0x33, 0x31, 0x11, 0x11, 0x31, 0x13, 0x31, 0x11, 0x08, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x11, 0x33, 0x11, 0x11, 0x11, 0x37, 0x78, 0x81, 0x11, 0x3B, 0x33, 0x39, 0xBB, 0x00, 0x00, 0x00, 0x87, 0x78, 0x77, 0x77, 0x77, 0x75, 0x40, 0x00, 0x03, 0x31,
0x13, 0x33, 0x31, 0x13, 0x31, 0x11, 0x11, 0x11, 0x11, 0x11, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,

0x11, 0x13, 0x33, 0x37, 0x78, 0x70, 0x11, 0x3B, 0x33, 0x93, 0xBB, 0x00, 0x00, 0x00, 0x87, 0x78, 0x87, 0x77, 0x77, 0x75, 0x00, 0x00, 0x00, 0x71, 0x11, 0x33, 0x11, 0x01, 0x37, 0x70, 0x00, 0x00,
0x11, 0x11, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x01, 0x10, 0x00, 0x00, 0x01, 0x33, 0x33, 0x33, 0x88, 0x10, 0x13, 0x3B,
0xBB, 0x9B, 0xB3, 0x00, 0x00, 0x00, 0x80, 0x88, 0x87, 0x77, 0x77, 0x00, 0x70, 0x00, 0x00, 0x31, 0x11, 0x13, 0x10, 0x17, 0x77, 0x10, 0x01, 0x10, 0x11, 0x11, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x11, 0x00, 0x00, 0x00, 0x01, 0x77, 0x31, 0x01, 0x33, 0x33, 0x37, 0x83, 0x01, 0x13, 0xBB, 0x39, 0x3B, 0xBB, 0x00, 0x00, 0x00, 0x70, 0x78,
0x88, 0x87, 0x77, 0x8F, 0xFF, 0x88, 0x88, 0x13, 0x11, 0x10, 0x01, 0x00, 0x00, 0x11, 0x77, 0x70, 0x11, 0x11, 0x37, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x71, 0x11, 0x10, 0x00, 0x00, 0x00, 0x07, 0x73, 0x11, 0x13, 0x37, 0x77, 0x70, 0x01, 0x13, 0xBB, 0x33, 0x3B, 0x77, 0x00, 0x00, 0x00, 0x77, 0x88, 0x77, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xF7, 0x33,
0x31, 0x00, 0x31, 0x11, 0x11, 0x13, 0x33, 0x10, 0x11, 0x11, 0x31, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x11, 0x11, 0x11, 0x11, 0x11, 0x10, 0x01, 0x31,
0x11, 0x33, 0x77, 0x73, 0x00, 0x01, 0x3B, 0x88, 0x88, 0x77, 0x88, 0x00, 0x00, 0x00, 0x78, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x33, 0x31, 0x01, 0x73, 0x11, 0x10, 0x00, 0x11, 0x10,
0x11, 0x33, 0x33, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x01, 0x37, 0x77, 0x77, 0x11, 0x11, 0x33, 0x77, 0x77, 0x00, 0x13, 0xB8, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x77, 0x31, 0x00, 0x13, 0x11, 0x00, 0x00, 0x00, 0x11, 0x13, 0x77, 0x73, 0x10, 0x8F, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x11, 0x11, 0x30, 0x00, 0x77, 0x87, 0x71, 0x11, 0x33, 0x33, 0x77, 0x73, 0x01, 0x38, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x73, 0x33, 0x11, 0x11, 0x10, 0x01, 0x11, 0x37, 0x71, 0x37, 0x77, 0x77, 0x30, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80,
0x31, 0x11, 0x11, 0x00, 0x78, 0x73, 0x01, 0x17, 0x77, 0x77, 0x77, 0x77, 0x17, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x37,

0x31, 0x11, 0x11, 0x13, 0x37, 0x77, 0x88, 0x73, 0x77, 0x77, 0x77, 0x70, 0x08, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x01, 0x10, 0x00, 0x00, 0x00, 0x37, 0x10,
0x11, 0x10,
0x01, 0x77, 0x77, 0x77, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x77, 0x33, 0x11, 0x11, 0x37, 0x77, 0x88,
0x87, 0x77,
0x77, 0x77, 0x77, 0x71, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7,
0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x07, 0x37, 0x77, 0x70, 0x77, 0x77, 0x77, 0x8F, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x87,
0x77, 0x77, 0x77, 0x31, 0x13, 0x37, 0x88, 0x87, 0x77, 0x77, 0x37, 0x77, 0x31, 0x00, 0x8F, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11,
0x77, 0x77, 0x18, 0x88, 0x75, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x41, 0x77, 0x77, 0x77, 0x77, 0x77, 0x37, 0x78, 0x77,
0x71, 0x00, 0x11, 0x13, 0x31, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x83, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x17, 0x77, 0x17, 0x88, 0x71, 0x8F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x00, 0x77,
0x77, 0x77, 0x77, 0x73, 0x33, 0x77, 0x73, 0x10, 0x11, 0x71, 0x01, 0x11, 0x00, 0x08, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x03,
0x77, 0x17, 0x88, 0x11, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x77, 0x77, 0x77, 0x37, 0x31, 0x33, 0x77,
0x73, 0x10,
0x00, 0x00, 0x11, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x17, 0x03, 0x88, 0x00, 0x78, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00,
0x37, 0x73, 0x71, 0x11, 0x11, 0x11, 0x37, 0x31, 0x11, 0x01, 0x33, 0x10, 0x00, 0x00, 0x00, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x01, 0x11, 0x11, 0x10, 0x00, 0x00,
0x11, 0x11, 0x00, 0x77, 0x03, 0x78, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x37, 0x37, 0x33, 0x31, 0x00, 0x01, 0x11, 0x11,
0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70,
0x00, 0x00,

0x00, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x04, 0x71, 0x07, 0x18, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x17,
0x33, 0x11, 0x00, 0x00, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x01, 0x11, 0x33, 0x33, 0x31, 0x00, 0x01,
0x77, 0x77, 0x70, 0x01, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x03, 0x33, 0x31, 0x11, 0x11, 0x11, 0x11, 0x11, 0x10,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x01, 0x11, 0x13, 0x33, 0x37, 0x77, 0x77, 0x77, 0x77, 0x00, 0x07, 0x8F, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF5, 0x00, 0x01, 0x33, 0x37, 0x73, 0x10, 0x00, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x33, 0x37, 0x77, 0x77, 0x77, 0x73, 0x00, 0x04, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF5, 0x00, 0x00, 0x13, 0x31, 0x10, 0x00, 0x00, 0x11, 0x11, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00,
0x00, 0x00, 0x11, 0x11, 0x13, 0x37, 0x77, 0x37, 0x73, 0x10, 0x00, 0x04, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF4, 0x00, 0x00,
0x03, 0x00, 0x00, 0x00, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x13, 0x33, 0x33, 0x33,
0x10, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF4, 0x00, 0x00, 0x01, 0x71, 0x01, 0x33, 0x31, 0x10, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x13, 0x37, 0x73, 0x31, 0x00, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00, 0x00, 0x00, 0x00, 0x13, 0x33, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x37, 0x33, 0x11, 0x00, 0x00, 0x00, 0x04, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00, 0x00, 0x00, 0x01, 0x31, 0x31, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x01, 0x33, 0x10, 0x00, 0x00, 0x00, 0x00, 0x04, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00, 0x00,
0x00, 0x01, 0x11, 0x10, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x48, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x87, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x17, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x87, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x13, 0x77, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x88, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x17, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x8F,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x40, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x77,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x15, 0x78, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x07, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x88, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x57, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x01, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x87, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x77, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x8F, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x74, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x77, 0x00, 0x00, 0x00, 0x00, 0x07, 0x78, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x88,
0x88, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x78, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xF8, 0x70, 0x07, 0x77, 0x70, 0x77, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88, 0xF8, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8,
0x88, 0x87, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xFF, 0x87, 0x75, 0x88, 0x87, 0x78, 0x8F,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88, 0x88, 0xF8, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x88, 0x88, 0x70, 0x00, 0x00, 0x00, 0x00,
0x78, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x87, 0x88, 0xF8, 0x88, 0x88, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0xF8, 0x8F,
0xFF, 0xFF, 0xFF, 0xFF, 0x88, 0x76, 0x00, 0x44, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0xF8, 0x8F, 0x8F, 0xF8, 0xFF, 0xFF, 0xFF, 0xF8, 0x88, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0x8F, 0xFF,
0x88, 0xF8, 0xFF, 0x88, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0xFF, 0x88, 0x88, 0x00, 0x00, 0x00, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x8F, 0x88, 0x88, 0x88, 0xFF, 0xF8,
0xF8,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xF8,
0x88, 0x88, 0x88, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0xF8, 0xFF, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x8F, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x00, 0x00, 0x00,
0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x8F, 0xF8, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0x88, 0x88, 0x8F, 0xF8, 0x88, 0x88, 0x88, 0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x88, 0x00, 0x00, 0x00, 0xFF,

};

extern unsigned int msec;

```
extern unsigned int deltasec, alarmsec;
extern unsigned char UP,DOWN,LEFT,RIGHT,SELECT;
//unsigned int sec, min, hr;
char *pcStr;
//int count;
//int sec = 04, min = 40, hr= 11;
//char pointer[10]={"11:40:04\0"};
char* pointer1;
char* alarm;

int x2, y2;
int main(void){ int i;
   int state = 0;
   int count = 0;
   int digitalEnable=0;
   int analogEnable=1;
   unsigned int time;
   time =msec;
  SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN |
SYSCTL_XTAL_8MHZ);
   PLL_Init();
   SysTick_Init(50000);
   Switch_Init();
  RIT128x96x4Init(1000000);
   Sound_Init();
//  RIT128x96x4_Logo(60);   // place in middle
   //RIT128x96x4StringDraw(pointer, 60, 60, 15);
  //RIT128x96x4_BMP(0, 60, ECE);
 // Delay(50000000);              // delay 3 sec at 50 MHz
//  RIT128x96x4Clear();
//   for(i=90; i>5; i=i-2){
//         RIT128x96x4_BMP(i, 80, Horse);
//         Delay(1000000);        // delay 0.3 sec at 50 MHz
//   }
   //RIT128x96x4StringDraw(pointer, 60, 60, 15);
   //pointer={"11:40:04"};

  //Delay(1000000000);          // delay 60 sec at 50 MHz
  //RIT128x96x4DisplayOff(); // screen saver
```

```
  //RIT128x96x4Clear(); //added
  RIT128x96x4_BMP(10, 80, db); //0,80
  Delay(50000000);
  RIT128x96x4Clear();

  //RIT128x96x4_Line(32, 48, 60, 48, 15);
  //RIT128x96x4_Line(32, 48, 32, 20, 15);
  //RIT128x96x4_ShowImage();
  EnableInterrupts();
  //RIT128x96x4StringDraw(const char *pcStr, unsigned long ulX,
                //         unsigned long ulY, unsigned char ucLevel)

  RIT128x96x4_BMP(0, 80, clock1); //0,80
 while(1){
        //int min, sec;
        //D2A(5000);
        //RIT128x96x4_ClearImage();

        //Conversion();

                RIT128x96x4_BMP(0, 80, clock1);
                //RIT128x96x4_BMP(0, 80, control);
                D2AMin(msec);
                D2ASec(msec);
                D2AHr(msec);

                pointer1 = DigitalRead(msec);
                RIT128x96x4StringDraw(pointer1, 70, 80, 15);

                alarm = DigitalRead(alarmsec);
                pcStr = "Alarm: ";
                RIT128x96x4StringDraw(pcStr, 70, 0, 15);
                RIT128x96x4StringDraw(alarm, 70, 10, 15);
           RIT128x96x4_ClearImage();

//Comment out the min = ... and UNcommnent switch to make sound work*/
        if(deltasec){
        state = 3;
        }
```

```
switch(state){
        case(0):
                if(SELECT){state = 3;}
                if(DOWN){
                state = 1;
                DOWN =0;
                }
                if(UP){
                state = 2;
                UP = 0;
                }
                SELECT = 0;
                RIGHT  = 0;
                LEFT   = 0;
                DOWN   = 0;
                UP     = 0;
                if(msec >= 0 && msec <= 28800000)
                {
                        pcStr = " (-.-)zz";       //Sleeping from 00:00 - 08:00 hours
                }
                else if(msec>28800000 && msec <=43200000)
                {
                        pcStr = "(^ . ^) ";       // Awake from 08:01 - 12:00 hours
                }
                else
                {
                        pcStr = "(^ v ^) ";     // Happy until bedtime
                }

                RIT128x96x4StringDraw(pcStr, 70, 50, 15);

                break;
        case(1)://Set time
                if(SELECT){state = 0;} else{ state =1;}
                if(UP){
                        SetTime(msec+60000);          //Add one minute
                        UP =0;
                }
                if(DOWN){
```

```
                SetTime(msec-60000);          //Subtracts one minute
                DOWN =0;
        }
        if(LEFT){
                SetTime(msec+3600000);    //Adds one hour
                LEFT = 0;
        }
        if(RIGHT){
                SetTime(msec-3600000);       //Soubtracts one hour
                RIGHT = 0;
        }
        SELECT = 0;
        RIGHT  = 0;
        LEFT   = 0;
        DOWN   = 0;
        UP     = 0;
  pcStr = "Set Time ";
        RIT128x96x4StringDraw(pcStr, 70, 50, 15);
        break;
case(2):
        if(SELECT){state = 0;} else{ state =2;}
        if(UP){
                SetAlarm(alarmsec+60000);  //Adds one minute
                UP =0;
        }
        if(DOWN){
                SetAlarm(alarmsec-60000);   //Subtracts one minute
                DOWN =0;
        }
        if(LEFT){
                SetAlarm(alarmsec+3600000);         //Adds one hour
                LEFT = 0;
        }
        if(RIGHT){
                SetAlarm(alarmsec-3600000);         //Subtracts one hour
                RIGHT = 0;
        }

        SELECT = 0;
```

```
                RIGHT  = 0;
                LEFT   = 0;
                DOWN   = 0;
                UP     = 0;

            pcStr = "Set Alarm ";
                RIT128x96x4StringDraw(pcStr, 70, 50, 15);
                break;
            case(3)://ALARM
                AlarmOn();
              if(SELECT){state = 0;} else{ state =3;}
                if(UP){
                    SetAlarm(alarmsec+300000);//snooze
                    state = 0;
                }
                SELECT = 0;
                RIGHT  = 0;
                LEFT   = 0;
                DOWN   = 0;
                UP     = 0;
            pcStr = "Wake up!";
                RIT128x96x4StringDraw(pcStr, 70, 50, 15);
                break;
            default:
                state = state;
                pcStr = "Default";
                RIT128x96x4StringDraw(pcStr, 70, 50, 15);
                break;
        }


};
}
```

```
/***************DigitalRead**************
Takes reads digital time
```

```
 Input: msec
 Output: char* pointer to string of
                    digital reading
          */
char* DigitalRead(unsigned int msec)
{
   char* pointer1;
   unsigned int hr, min, sec;
   hr = (msec/(60*60*1000))%24;
   min = (msec/(60*1000))%60;
   sec = (msec/1000)%60;
   sprintf(pointer1, "%.2u:%.2u:%.2u", hr, min, sec);
   //RIT128x96x4StringDraw(pointer1, 70, 60, 15);
   return pointer1;
}


/****************D2AMin***************
 Takes msec and draws the minute hand
   by look up table
 Input: msec
 Output: none
          */
void D2AMin(unsigned int msec)
{   //unsigned int hr;
   unsigned int min;
   //unsigned int sec;
//    hr = (msec/(60*60*1000))%24;
   min = (msec/(60*1000))%60;
   //sec = (msec/1000)%60;
   RIT128x96x4_Line(32, 48, linetable[min][0], linetable[min][1], 15);
   RIT128x96x4_ShowImage();
   return;
}



 /****************D2ASec***************
 Takes msec and draws the second hand
```

```
   by look up table
 Input: msec
 Output: none
          */
void D2ASec(unsigned int msec)
{    //unsigned int hr;
   //unsigned int min;
   unsigned int sec;
   //hr = (msec/(60*60*1000))%24;
   //min = (msec/(60*1000))%60;
   sec = (msec/1000)%60;
   RIT128x96x4_Line(32, 48, linetable[sec][0], linetable[sec][1], 15);
   RIT128x96x4_ShowImage();
   return;
}

/***************D2AHr***************
 Takes msec and draws the hour hand
   by look up table
 Input: msec
 Output: none
          */
void D2AHr(unsigned int msec)
{    //unsigned int hr;
   //unsigned int min;
   //unsigned int sec;
   unsigned int qhr;
   //hr = (msec/(60*60*1000))%24;
   //min = (msec/(60*1000))%60;
   //sec = (msec/1000)%60;
   qhr = (msec/(60*1000*15))%48;
   RIT128x96x4_Line(32, 48, hourtable[qhr][0], hourtable[qhr][1], 15);
   RIT128x96x4_ShowImage();
   return;
}

//Filename: logo.c
//Author: Duc Tran, Brandon Wong
//Initial Creation Date: September 10, 2013
```

//Description: Functions for drawing lines on analog clock
//Lab Number: MW 2-3:30
//TA : Omar & Mahesh
//Date of last revision :September 27, 2013
//Hardware Configuration : NONE

// Logo.c
// Runs on LM3S1968 or LM3S8962
// Jonathan Valvano
// November 12, 2012

#include "rit128x96x4.h"

//
****
//
// The size of the header on the bitmap image. For a typical 4bpp Windows

```
// bitmap this is 0x76 bytes comprising the total number of bytes in the
// bitmap file header, the bitmap info header and the palette.
//
//
//*****************************************************************************
#define BITMAP_HEADER_SIZE   0x76

//
//*****************************************************************************
//
// The byte offsets into the image at which we can find the height and width.
// Each of these values are encoded in 4 bytes.
//
//
//*****************************************************************************
#define BITMAP_WIDTH_OFFSET  0x12
#define BITMAP_HEIGHT_OFFSET 0x16
//
//*****************************************************************************
//
// Simple 4-bit gray scale image for test.  This image was dumped directly from
// a 16 color Windows-format bitmap (BMP) file. The size must be
// (BITMAP_HEADER_SIZE + ((width /2) * height)) bytes.
//
//
//*****************************************************************************

//
//*****************************************************************************
// //
// // A bitmap for the ECE logo.  This logo is 128x34.
// //
//
//*****************************************************************************
// const unsigned char LogoBuffer[] ={
//  0x42, 0x4D, 0xF6, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x00, 0x00, 0x00, 0x28,
0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x22, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x00,
0x00,
//  0x00, 0x00, 0x80, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x80,
// 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0xC0, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF,
// 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF, 0xFF, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
// 0x00,
// 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xC0,
// 0x00, 0xE0, 0x00, 0xE0, 0xE0, 0x00, 0xA0, 0xE0, 0x09, 0x00, 0x00, 0x09, 0xE0, 0x00, 0x00, 0xE0, 0x00, 0xE0, 0xE0, 0x09, 0xE0, 0x09, 0xE0, 0x00, 0x00, 0x00, 0xE0, 0x00, 0x00, 0x09, 0xE0, 0x00,
// 0x09, 0xC0, 0x00, 0xE0, 0x00, 0x09, 0x00, 0xE0, 0x00, 0xE0, 0xA0, 0x00, 0xE0, 0x00, 0xEB, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xA0, 0x00, 0x00, 0x90, 0x00, 0x90, 0x00, 0x90, 0x00, 0x90, 0x00, 0x90,
// 0x00, 0x90, 0x00, 0x90, 0x00, 0x00, 0x0A, 0x00, 0x0A, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x90, 0x00, 0x90, 0x00, 0x90, 0x90, 0x00, 0x90, 0x9A, 0x00, 0x00, 0x90, 0x00, 0x00, 0x9E, 0x90, 0x00,
// 0x0E, 0x90, 0x0E, 0x00, 0x0A, 0x90, 0x00, 0x90, 0x90, 0x00, 0x0A, 0x00, 0x9A, 0x00, 0x90, 0x00, 0x90, 0x0E, 0x00, 0x0A, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
// 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E,
// 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E,

0xDE, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0x9E, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E,
// 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x9E, 0xDE, 0x00, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xE0, 0x00, 0xCB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB, 0xFB,
// 0xFB, 0xFF, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xEE, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00,
// 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFB, 0xFF, 0xFF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xF9, 0xEE, 0xEE, 0xE0, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xFB, 0xFF, 0xE0, 0x00, 0x00, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE9, 0xE0, 0x00, 0x00, 0x00, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x0C, 0xFF, 0xFF,
// 0xFF, 0xFF, 0xDC, 0x00, 0x00, 0x9E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xCE, 0xEE, 0xEC, 0xDF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x0E, 0xDF, 0xFE,
// 0xDF, 0xDE, 0xDF, 0xF0, 0x00, 0x00, 0x0E, 0xFF, 0xDF, 0x9E, 0xD0, 0x00, 0x00, 0x9F, 0xFF, 0xDE, 0xDF, 0xDF, 0xF0, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0xF9, 0x00, 0xEF, 0xFF, 0xF0, 0xEF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xE0, 0x09, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xCE, 0xEE, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xA0, 0x00, 0xE0, 0x00, 0xAF, 0xF0,
// 0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xFF, 0xF0, 0x00, 0x00, 0xEB, 0x00, 0x00, 0x0F, 0xFB, 0x00, 0x00, 0x00, 0xFB, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0x00, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0x0E, 0xEE, 0xC0, 0x9E, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0xDE, 0x90, 0x00, 0x0F, 0xF0,
// 0x00, 0x00, 0x00, 0x9E, 0x00, 0x0E, 0xFE, 0x00, 0x00, 0x00, 0x00, 0xD0, 0x00, 0x0F, 0xF0,

0x00, 0x00, 0x00, 0x0E, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0xE0, 0xCF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF9, 0xAE, 0xEE, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0xCF, 0xF0,
// 0x00, 0x00, 0x00, 0x0B, 0xC0, 0xEF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xA0, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0x0E, 0xEE, 0xEC, 0x9F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x9F, 0xF0,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0xDF, 0xD0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0xA0, 0x00, 0x00, 0x00, 0xA0, 0xEE, 0xEE, 0xEA, 0xA0, 0x00, 0x00, 0x00, 0xA0, 0x00, 0x00, 0x00, 0xE0, 0x00, 0xAF, 0xF0,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0x9F, 0xFF, 0xFF, 0xFF, 0xF0, 0xDF,
// 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x9E, 0x90, 0x9A, 0x90, 0x9E, 0x9C, 0x00, 0xEE, 0xEE, 0xEE, 0x00, 0x9E, 0x9E, 0x90, 0x9E, 0x90, 0x9E, 0x9E, 0x90, 0x00, 0x0F, 0xF0,
// 0x00, 0x00, 0x0E, 0x00, 0x0E, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xF0, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xEF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xFF, 0xC0, 0xEE, 0xEE, 0xEE, 0xE0, 0xF9, 0xFB, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0xCF, 0xF0,
// 0xC0, 0xA0, 0xF9, 0x00, 0x09, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF9, 0x00, 0xC0, 0xEF, 0xC0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xF0, 0x9F,
// 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x00, 0x00, 0x0E, 0xEE, 0xEE, 0xEE, 0xD0, 0x0C, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x9F, 0xFE,
// 0xDE, 0x9E, 0xF0, 0x00, 0x00, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xFF, 0x9E, 0xDE, 0xFE, 0x90, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xEF, 0xFF, 0xFF, 0xFF, 0xF0, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0,

0xEE, 0xEA, 0xEE, 0xEE, 0xEE, 0xEE, 0xEA, 0xEE, 0xE0, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0,
0x00, 0xAF, 0xF0,
// 0x00, 0x00, 0xE0, 0x00, 0x00, 0xFF, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0xF0,
0x00, 0x00, 0x0F, 0xA0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF,
0xF0, 0xDF,
// 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x0E,
0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00,
0x0F, 0xF0,
// 0x00, 0x00, 0x0A, 0x00, 0x00, 0xFF, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xF0, 0xEF,
// 0xFF, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0xE9, 0xE0, 0xE9, 0x00, 0x00,
0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xE0, 0x00, 0xE9, 0xE0, 0xE9, 0xE0, 0xE0, 0x00,
0xCF, 0xF0,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0xEF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x00, 0x0F, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0xDF, 0xFF, 0xFF, 0xFF,
0xF0, 0x9F,
// 0xFE, 0xFF, 0xFF, 0xF0, 0x0F, 0xFF, 0xFF, 0x9F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x0C,
0xEE, 0xCE, 0xDE, 0xEE, 0xEE, 0xEE, 0xE0, 0x0E, 0xEE, 0x00, 0xDF, 0xFF, 0xFF, 0xFF, 0x90,
0x00, 0x9F, 0xF0,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x0E, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x0B, 0xFF, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF,
0xF0, 0xFF,
// 0xF9, 0xFF, 0xFF, 0xF0, 0x0B, 0xFF, 0xFF, 0xEF, 0xFF, 0xFF, 0xFF, 0xFB, 0x00, 0xEE, 0xE0,
0x00, 0xA0, 0x00, 0xA0, 0x00, 0x00, 0x00, 0x0A, 0xEA, 0x00, 0xFF, 0xFF, 0xFF, 0xE0, 0x00,
0xAF, 0xF0,
// 0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xFB, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x00, 0x0B, 0xF0,
0x00, 0x00, 0x00, 0xE0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0x00, 0x9F, 0xFF, 0xFF, 0xFF,
0xF0, 0xDF,
// 0xFE, 0x0F, 0xFF, 0xD0, 0x0E, 0xFF, 0xF0, 0x9F, 0xFF, 0xFF, 0xFF, 0xD0, 0x0E, 0xCA,
0x0E, 0xFF, 0xFE, 0x9E, 0x9E, 0x9E, 0xFF, 0xFF, 0x90, 0x0E, 0xE0, 0x0A, 0xFF, 0xFF, 0x90,
0x00, 0x9F, 0xF0,
// 0x00, 0x00, 0x0E, 0xD0, 0x00, 0x00, 0x9F, 0xD0, 0x00, 0x00, 0x9F, 0xF0, 0x00, 0x0F, 0xFE,
0x00, 0x00, 0x00, 0xF0, 0x00, 0x0E, 0x00, 0x09, 0xFF, 0xF0, 0x00, 0x09, 0xFF, 0xFF, 0xFF,
0xC0, 0x00,
// 0xF9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xEF, 0xFF, 0xE0, 0xE0, 0x0A, 0xE0, 0x00, 0xE9,
0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0x00, 0xEA, 0xE0, 0x00, 0xE9, 0xC0, 0x09,
0xFB, 0xFF,
// 0xFB, 0xFF, 0xFB, 0xE0, 0x00, 0x00, 0x00, 0xEF, 0xFB, 0xEF, 0xF0, 0xF0, 0x09, 0xEF, 0xFB,

0xFB, 0xFB, 0xFF, 0xF0, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xDE, 0xDE, 0xDF, 0xFF, 0xFF, 0xFE,
0xDE, 0xDE,
// 0xDF, 0xDE, 0xDF, 0xDE, 0xDE, 0xDE, 0xDF, 0xFF, 0xF0, 0x00, 0x00, 0xEE, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0xC0, 0x00, 0x00, 0x00,
0x00, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x0B, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xCA, 0xE0, 0xC0, 0xEF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0xA0, 0xC0, 0x00,
0x00, 0x00, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF,
// 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x90, 0x9E, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xDE, 0xDE, 0x90, 0x90,
0x9A, 0x90,
// 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9A,
0x90, 0x9A, 0x90, 0x9A, 0x90, 0x9F, 0x00, 0x00, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0,
0xE9, 0xE0,
// 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xF9, 0xE0, 0xE9, 0xE0, 0xE9,
0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xE9, 0xE0, 0xF9, 0xEB,
0xF9, 0xEB,
// 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB,
0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0xEB, 0xF9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x90,
0x00, 0x00, 0x00,
// 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
// 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x00, 0x00,
0x00, 0x90, 0x00, 0x90, 0x00, 0x00, 0x00, 0x00, 0x09, 0x00, 0xE0, 0xE0, 0xC9, 0x00, 0x00,
0x00, 0x00,
// 0x00, 0xC9, 0xA0, 0xE0, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE9, 0x00, 0xE0, 0xA9,
0x00, 0x00, 0x00, 0x00, 0x09, 0xA0, 0x09, 0xE0, 0x00, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xC9,
0x00, 0xE0,
// 0xE0, 0x00, 0xA0, 0x00, 0xA0, 0xE0, 0x09, 0xC0, 0x00, 0x00, 0xA9, 0x00, 0xA0, 0x00, 0x0B,
0xC0, 0xA0, 0x00, 0xE0, 0xC0, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x90, 0x90, 0x9E, 0x00, 0x00,
0x00, 0x00,
// 0x00, 0x90, 0x00, 0x90, 0x0A, 0x90, 0x0A, 0x90, 0x0A, 0x90, 0x00, 0x90, 0x0A, 0x90, 0x0E,

```
0x00, 0x0E, 0x00, 0x00, 0x9C, 0x0E, 0x00, 0x90, 0x00, 0x00, 0x90, 0x00, 0x90, 0x00, 0x90,
0x00, 0x00,
//  0x00, 0x00, 0x00, 0x00, 0x00, 0x90, 0x0E, 0x90, 0x00, 0x00, 0x90, 0x00, 0x00, 0x90, 0x90,
0x00, 0x00, 0x00, 0x90, 0x0E, 0x00, 0x00, 0x00, 0xE0, 0xC0, 0xE0, 0xC0, 0x00, 0xC0, 0x00,
0xC9, 0x00,
//  0x00, 0x00, 0xC0, 0xE0, 0xC0, 0x00, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xA0, 0xC0,
0x00, 0xE0, 0xA0, 0x00, 0x00, 0xC0, 0x00, 0xC0, 0x00, 0x00, 0xE0, 0x00, 0xE0, 0x00, 0xE0,
0xC0, 0x00,
//  0x00, 0xA0, 0xC0, 0x00, 0x00, 0x00, 0xC0, 0xE0, 0x00, 0x00, 0xE0, 0x00, 0xC0, 0x00, 0xC0,
0x00, 0xE9, 0x00, 0xE0, 0xA0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
//  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
//  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,

// };


// //*************RIT128x96x4_Logo*****************************************
// //  Displays the 4-bit color BMP image stored in LogoBuffer
// //  Inputs: ypos (94 is bottom, 80 is near the bottom)
// //  Outputs: none
// //  Must be less than or equal to 128 pixels wide by 80 rows high
// //  The width must be an even number
// void RIT128x96x4_Logo(unsigned long ypos){
//         unsigned long ulRow, ulWidth, ulHeight;
//         unsigned char *pucRow;

//         // Extract the width and height from the bitmap data.  These are encoded
//         // in 4 byte fields but this application can't support images wider than
//         // 128 pixels or taller than 80 rows so we merely read the least
//         // significant byte.
//         //
//         ulHeight = (unsigned long)LogoBuffer[BITMAP_HEIGHT_OFFSET];
//         ulWidth = (unsigned long)LogoBuffer[BITMAP_WIDTH_OFFSET];

//         // Display the BMP Image.
```

```
//          // The image is an ulWidth by ulHeight 4-bit gray scale image in BMP
//          // format.  The divides by two are to account for the fact that there
//          // are 2 pixels in each byte of image data.

//          // Get a pointer to the first row of pixels in the image (which maps to the
//          // bottom row on the display since bitmaps are encoded upside down).
//          //************ ypos is 94 (bottom) 80 (starter file) ***************
//          pucRow = (unsigned char *)&LogoBuffer[BITMAP_HEADER_SIZE];
//          for(ulRow = 0; ulRow < ulHeight; ulRow++){
//          // Display in reverse row order.  We center the image horizontally on
//          // the display.
//          RIT128x96x4ImageDraw(pucRow, ((128 - ulWidth) / 2), (ypos - ulRow), ulWidth, 1);
//
//          pucRow += (ulWidth / 2);   // Move to the next row in the source image.
//          }
// }
//*************RIT128x96x4_BMP*********************************************
//  Displays a 16 color BMP image
//  (xpos,ypos) is the screen location of the lower left corner of BMP image
//  Inputs: xpos horizontal position to display this image, columns from the left edge
//          must be less than 128 and even
//          0 is on the left 126 on the right
//          ypos vertical position to display this image, rows from the top edge
//          2 is on the top, 95 is bottom, 80 is near the bottom
//          pointer to a 16 color BMP image
//  Outputs: none
//  Must be less than or equal to 128 pixels wide by 80 rows high
//  The BMP image width must be an even number
void RIT128x96x4_BMP(unsigned long xpos, unsigned long ypos, const unsigned char *Buffer){
          unsigned long ulRow, ulWidth, ulHeight;
          unsigned char *pucRow;

          // Extract the width and height from the bitmap data.  These are encoded
          // in 4 byte fields but this application can't support images wider than
          // 128 pixels or taller than 80 rows so we merely read the least
          // significant byte.
          //
          ulHeight = (unsigned long)Buffer[BITMAP_HEIGHT_OFFSET];
          ulWidth = (unsigned long)Buffer[BITMAP_WIDTH_OFFSET];
```

```
        // Display the BMP Image.
        // The image is an ulWidth by ulHeight 4-bit gray scale image in BMP
        // format.  The divides by two are to account for the fact that there
        // are 2 pixels in each byte of image data.

        // Get a pointer to the first row of pixels in the image (which maps to the
        // bottom row on the display since bitmaps are encoded upside down).
        //************ ypos is 94 (bottom) 80 (starter file) ***************
        pucRow = (unsigned char *)&Buffer[BITMAP_HEADER_SIZE];
        for(ulRow = 0; ulRow < ulHeight; ulRow++){
        // Display in reverse row order.
        RIT128x96x4ImageDraw(pucRow, xpos, (ypos - ulRow), ulWidth, 1);

//      pucRow += (ulWidth / 2);   // Move to the next row in the source image.
        pucRow += (((ulWidth + 7) / 8) * 4);   // Move to the next row in the source image.
        }
}


#define XPOS_OFFSET 0x0
#define YPOS_OFFSET 0x1
#define HEIGHT_OFFSET  0x2
#define WIDTH_OFFSET  0x3
#define COLOR_OFFSET  0X4

static int xpos,ypos,height,width;
static char color;
static unsigned char PLOT[96][128];
//const unsigned char Line_Buffer[] =
//{
//};


//************RIT128x96x4_Line*********************************************
// Draws one line in the RAM version of the image
// Inputs: (x1,y1) is the start point
// (x2,y2) is the end point
// color is 0 (off) to 15 (white)
```

```
// coordinates range from 0 to MAX, 0 is bottom or left, MAX is top or right
// Outputs: none

void RIT128x96x4_Line(int x1, int y1, int x2, int y2, unsigned char color)
{

  int sx,sy, dx, dy, err, e2;
  int i =1;
  xpos = x1;
  ypos = y1;
  height = y2-y1;
  width = x2-x1;
  color = color;
  ////

  dx = abs(x2-x1);
  dy = abs(y2-y1);
  if(x1<x2){
        sx = 1;
  }
  else
  {
        sx = -1;
  }
  if(y1<y2)
  {
        sy = 1;
  }
  else
  {
        sy = -1;
  }
  err = dx - dy;
  while(i==1){
        PLOT[y1][x1] = color;
        if((x1 == x2) && (y1 == y2))
        {
                        break;
        }
```

```
        e2 = 2*err;
        if(e2 > -dy)
        {
                err = err - dy;
                x1 = x1 + sx;
        }
        if((x1 == x2) && (y1 == y2))
        {
                PLOT[y1][x1]= color;
                break;
        }
        if(e2<dx)
        {
                err = err + dx;
                y1 = y1 + sy;
        }
   }

}
//RIT128x96x4ImageDraw(const unsigned char *pucImage, unsigned long ulX,
//                     unsigned long ulY, unsigned long ulWidth,
//                     unsigned long ulHeight)
//*************RIT128x96x4_ShowImage************************************
// Copies the RAM version of the image to the OLED
// Inputs: none
// Outputs: none
void RIT128x96x4_ShowImage(void)
{
   int i, j;
   unsigned char *pucImage;


   for(j = 0; j < 128; j++)
   {
        for(i = 0; i < 95; i++)
        {
                if(PLOT[i][j] == 0)
                {
                        continue;
```

```
                }
                RIT128x96x4ImageDraw(pucImage,j,i,2, 2);
            }
    }




//    unsigned char *pucImage;
//    int ulRow,ulColumn,ulWidth,ulHeight,x1,y1,slopex,slopey;
//
//    ulHeight = HEIGHT;
//    ulWidth = WIDTH;//(WIDTH%2)+
//    *pucImage = COLOR;
//    x1 = XPOS;//+(XPOS%2);
//    y1 = YPOS;
//    slopex = ulWidth/Greatest_Common(ulHeight,ulWidth);
//    slopex = (slopex+slopex%2);
//    slopey = ulHeight/Greatest_Common(ulHeight,ulWidth);
//    slopey = slopey + slopey%2;
//    for (ulRow = slopey;ulRow <ulHeight;ulRow=ulRow+slopey){
//          x1 = x1+slopex;
//      RIT128x96x4ImageDraw(pucImage,x1,y1+ulRow,slopex,slopey);
//    }


}

    //*************RIT128x96x4_ClearImage*************************************
// Clears the RAM version of the image
// Inputs: none
// Outputs: none
void RIT128x96x4_ClearImage(void)
{
    int i,j;
    xpos = 0;
    ypos = 0;
    height = 0;
```

```
    width = 0;
    color = 0;
    //RIT128x96x4Clear();
    for(i =0; i < 96;i++){
            for(j =0;j <128;j++){
                    PLOT[i][j] =0x00;
            }
    }
    //RIT128x96x4_ShowImage();
}



//int Greatest_Common(int x,int y)
//{
//    int mod;
//    mod = x%y;
//    if(mod == 0){
//            return y;
//    }
//    else{
//            return(y,mod);
//    }
//}

//Filename: PeriodicSysTickInts.c
//Author: Duc Tran, Brandon Wong
//Initial Creation Date: September 18, 2013
//Description: SysTick interrupt handler for Time Manager Module and Alarm
//Lab Number: MW 2-3:30
//TA : Omar & Mahesh
//Date of last revision :September 27, 2013
//Hardware Configuration : NONE

// SysTickInts.c
// Runs on LM3S1968
// Use the SysTick timer to request interrupts at a particular period.
// Daniel Valvano
// June 27, 2011
```

```
/* This example accompanies the book
   "Embedded Systems: Real Time Interfacing to the Arm Cortex M3",
   ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2011

   Program 5.12, section 5.7

 Copyright 2011 by Jonathan W. Valvano, valvano@mail.utexas.edu
        You may use, edit, run or distribute this file
        as long as the above copyright notice remains
 THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS, IMPLIED
 OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.
 VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
 OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 For more information about my classes, my research, and my books, see
 http://users.ece.utexas.edu/~valvano/
 */

// oscilloscope or LED connected to PD0 for period measurement

#include "SoundModule.h"

#define NVIC_SYS_PRI3_R         (*((volatile unsigned long *)0xE000ED20))  // Sys. Handlers 12 to 15 Priority
#define NVIC_ST_CTRL_R          (*((volatile unsigned long *)0xE000E010))
#define NVIC_ST_RELOAD_R        (*((volatile unsigned long *)0xE000E014))
#define NVIC_ST_CURRENT_R       (*((volatile unsigned long *)0xE000E018))
#define NVIC_ST_CTRL_CLK_SRC    0x00000004  // Clock Source
#define NVIC_ST_CTRL_INTEN      0x00000002  // Interrupt enable
#define NVIC_ST_CTRL_ENABLE     0x00000001  // Counter mode
#define GPIO_PORTG_DIR_R        (*((volatile unsigned long *)0x40026400))
#define GPIO_PORTG_DEN_R        (*((volatile unsigned long *)0x4002651C))
#define SYSCTL_RCGC2_R          (*((volatile unsigned long *)0x400FE108))
#define SYSCTL_RCGC2_GPIOG      0x00000040  // port D Clock Gating Control
```

```
void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
long StartCritical (void);        // previous I bit, disable interrupts
void EndCritical(long sr);       // restore I bit to previous value
void WaitForInterrupt(void);  // low power mode
volatile unsigned long Counts = 0;
#define PG2             (*((volatile unsigned long *)0x40026010))

int msec=0;
unsigned int countflag = 0;
unsigned int deltasec = 0;
unsigned int alarmsec = 0;


// **************SysTick_Init*********************
// Initialize Systick periodic interrupts
// Input: interrupt period
//         Units of period are 20ns
//         Maximum is 2^24-1
//         Minimum is determined by length of ISR
// Output: none
void SysTick_Init(unsigned long period){
  SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOG; // activate port D
  Counts = 0;
  GPIO_PORTG_DIR_R |= 0x04;   // make PD0 out
  GPIO_PORTG_DEN_R |= 0x04;   // enable digital I/O on PD0
  NVIC_ST_CTRL_R = 0;     // disable SysTick during setup
  NVIC_ST_RELOAD_R = period-1;// reload value
  NVIC_ST_CURRENT_R = 0;        // any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000; // priority 2
                          // enable SysTick with core clock and interrupts
  NVIC_ST_CTRL_R =
NVIC_ST_CTRL_ENABLE+NVIC_ST_CTRL_CLK_SRC+NVIC_ST_CTRL_INTEN;
}


 /***************SysTick_Handler**************
 Interrupt Handler. Counts milliseconds.
 Toggles heartbeat PG2.
```

```
 Input: none
 Output: none
          */
// Interrupt service routine
// Executed every 20ns*(period)
//alarmsec
void SysTick_Handler(void){
   //DisableInterrupts();
  PG2 ^= 0x04;           // toggle PD0
   //status LED
   msec = (msec + 1) % 86400000;
   if(alarmsec == msec){
          deltasec = 1;
   }
//    if(countflag == 0)
//    {
//          Counts = Counts + 1;
//    }
}


/****************SetTime***************
 Takes msec and updates time
 Input: msec
 Output: none
          */
void SetTime(int ms)
{
   NVIC_ST_CTRL_R ^= NVIC_ST_CTRL_INTEN;
   msec = ms;
   NVIC_ST_CTRL_R ^= NVIC_ST_CTRL_INTEN;
}

/****************SetAlarm***************
 Takes msec and Sets Alarm
 Input: msec
 Output: none
          */
void SetAlarm(unsigned int ms)
```

```
{
   deltasec =0;
   alarmsec = ms%86400000;
}

/****************AlarmOn***************
 Toggles PH0 and PH1 for alarm ringing
 Input: None
 Output: none
          */
void AlarmOn(void)
{
   NVIC_ST_CTRL_R ^= NVIC_ST_CTRL_INTEN;
   PH0 ^=0x01;
   PH1 ^=0x02; //Delay1(400);
   NVIC_ST_CTRL_R ^= NVIC_ST_CTRL_INTEN; // PH0 =0x00; //PH1 =0x01;
}

//Filename: Switch.c
//Author: Duc Tran, Brandon Wong
//Initial Creation Date: September 13, 2013
//Description: Switch Module
//Lab Number: MW 2-3:30
//TA : Omar & Mahesh
//Date of last revision :September 27, 2013
//Hardware Configuration :  Switches on board.
//                                                Can be found in Lab3_artist.sch
//                                                UP_PG3 =  0x08;//0x08;
//                                                DOWN_PG4 = 0x10;//0x10;
//                                                LEFT_PG5 = 0x20;//0x20;
//                                                RIGHT_PG6 = 0x40;
//                                                SELECT_PG7 = 0x80;


#include "SwitchModule.h"


#define GPIO_PORTG_DATA_R     (*((volatile unsigned long *)0x400263FC))
#define GPIO_PORTG_DIR_R      (*((volatile unsigned long *)0x40026400))
```

```
#define GPIO_PORTG_AFSEL_R   (*((volatile unsigned long *)0x40026420))
#define GPIO_PORTG_DEN_R     (*((volatile unsigned long *)0x4002651C))
#define SYSCTL_RCGC2_R       (*((volatile unsigned long *)0x400FE108))
#define SYSCTL_RCGC2_GPIOG   0x00000040  // port G Clock Gating Control
#define GPIO_PORTG_RIS_R     (*((volatile unsigned long *)0x40026414))
#define GPIO_PORTG_PUR_R                    (*((volatile unsigned long *)0x40026510))
#define GPIO_PORTG_IS_R      (*((volatile unsigned long *)0x40026404))
#define GPIO_PORTG_IBE_R     (*((volatile unsigned long *)0x40026408))
#define GPIO_PORTG_IEV_R     (*((volatile unsigned long *)0x4002640C))
#define GPIO_PORTG_ICR_R     (*((volatile unsigned long *)0x4002641C))
#define GPIO_PORTG_IM_R      (*((volatile unsigned long *)0x40026410))
#define NVIC_PRI7_R        (*((volatile unsigned long *)0xE000E41C))
#define NVIC_PRI0_R        (*((volatile unsigned long *)0xE000E400))
#define NVIC_EN0_R         (*((volatile unsigned long *)0xE000E100))
static unsigned long UP_PG3 =  0x08;//0x08;
static unsigned long DOWN_PG4 = 0x10;//0x10;
static unsigned long LEFT_PG5 = 0x20;//0x20;
static unsigned long RIGHT_PG6 = 0x40;
static unsigned long SELECT_PG7 = 0x80;
void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
long StartCritical (void);    // previous I bit, disable interrupts
void EndCritical(long sr);    // restore I bit to previous value
void WaitForInterrupt(void);  // low power mode
unsigned long GPIO;
unsigned char UP,DOWN,LEFT,RIGHT,SELECT;
volatile unsigned long FallingEdges = 0;


/****************Switch_Init***************
 Description: initilizes switches
 Input: none
 Output: none  */
void Switch_Init(void){
   SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOG;
   UP =0;
   DOWN =0;
   LEFT = 0;
   RIGHT =0;
```

```
    SELECT = 0;
    FallingEdges =  0;
    GPIO_PORTG_DIR_R &= ~0xF8;
    GPIO_PORTG_DEN_R |= 0xF8;
    GPIO_PORTG_PUR_R  |= 0xF8;
    GPIO_PORTG_IS_R &= ~0xF8;
    GPIO_PORTG_IBE_R &= ~0xF8;
    GPIO_PORTG_IEV_R |= 0xF8;
    GPIO_PORTG_ICR_R = 0xF8;
    GPIO_PORTG_IM_R    |= 0xF8; // enable interrupt
    NVIC_PRI7_R = (NVIC_PRI7_R&0x00FFFFFF)|0x40000000; // (g) priority 5
 // NVIC_EN0_R |= 4;           // (h) enable interrupt 2 in NVIC
    //NVIC_EN0_R |= 0x00000040;
    //NVIC_PRI7_R = (NVIC_PRI0_R&0x00FFFFFF)|0x40000000; // (g) priority 2
    NVIC_EN0_R |= 0x80000000;
    //EnableInterrupts();
}


void GPIOPortG_Handler(void){
    FallingEdges+=1;
    if(GPIO_PORTG_RIS_R& UP_PG3){
            GPIO_PORTG_ICR_R = UP_PG3;
            UP = 1;
    }
    if(GPIO_PORTG_RIS_R& DOWN_PG4){
            GPIO_PORTG_ICR_R = DOWN_PG4;
            DOWN = 1;
    }
    if(GPIO_PORTG_RIS_R& LEFT_PG5){
            GPIO_PORTG_ICR_R = LEFT_PG5;
            LEFT = 1;
    }

    if(GPIO_PORTG_RIS_R& SELECT_PG7){
            GPIO_PORTG_ICR_R = SELECT_PG7;
            SELECT = 1;
    }
    if(GPIO_PORTG_RIS_R& RIGHT_PG6){
```

```
        GPIO_PORTG_ICR_R = RIGHT_PG6;
        RIGHT = 1;
    }
}
```

```
//Filename: SoundModule.c
//Author: Duc Tran, Brandon Wong
//Initial Creation Date: September 13, 2013
//Description: Switch Module
//Lab Number: MW 2-3:30
//TA : Omar & Mahesh
//Date of last revision :September 27, 2013
//Hardware Configuration :  NONE

#define SYSCTL_RCGC2_R          (*((volatile unsigned long *)0x400FE108))
#define GPIO_PORTH_DIR_R        (*((volatile unsigned long *)0x40027400))
#define GPIO_PORTH_DEN_R        (*((volatile unsigned long *)0x4002751C))
#define PH0                                                 (*((volatile unsigned long
*)0x40027004))
#define PH1                                                 (*((volatile unsigned long
*)0x40027008))
#define GPIO_PORTH_PUR_R        (*((volatile unsigned long *)0x40027510))


 /***************Sound_Init**************
 Initializes sound port H
 Input: none
 Output: none
          */
void Sound_Init(void){volatile unsigned long delay;
    SYSCTL_RCGC2_R |=0x00000080;
```

```
    delay =SYSCTL_RCGC2_R;
    GPIO_PORTH_DIR_R |= 0x03;
   GPIO_PORTH_DEN_R |= 0x03;
    GPIO_PORTH_PUR_R |= 0x03;

    PH1 &= 0x02;
    PH0 |= 0x01;
}
```