8086: The code that I wrote in the exam has no error so I did not change any thing.

ARM:

List of the correction in different files are commented as “Edited” in front of instructions.

Startup file:

insertionSort PROC

EXPORT insertionSort

**PUSH** {R4**-R11,**LR}

**MOV** I**,**#1

WHILE1 **CMP** I**,**LENGTH ;Edited:added while1 label

BHS END\_WHILE

LDRSB TEMP**,[**A**,**I**]** ;Edited:LDRB IS NOT RIGHT IT SHOULD BE LDRSB BECAUSE IT IS SIGN

**MOV** X**,**TEMP

**SUB** J**,**I**,**#1

WHILE2 **CMP** J**,**#0 ;Edited:adding a label

BLT END\_INNER\_WHILE ;Edited:IT SHOULD BE BLT

LDRSB TEMP**,[**A**,**J**]**

**CMP** TEMP**,**X

BLE END\_INNER\_WHILE ;Edited:LS was wrong because it is a sign number so LE is right

**ADD** TEMP2**,**J**,**#1

STRB TEMP**,[**A**,**TEMP2**]**

**SUB** J**,**J**,**#1 ;Edited:IT SHOULD BE 1 INSTED OF -1

B WHILE2 ;Edited:REPEAT INNER WHILE

END\_INNER\_WHILE

**ADD** TEMP2**,**J**,**#1

STRB X**,[**A**,**TEMP2**]**

**ADD** I**,**I**,**#1

B WHILE1 ;REPEAT LOOP

END\_WHILE **POP** {R4**-R11,**PC}

ENDP

Main.c:

#include <stdio.h>

#include "LPC17xx.h"

#include "button.h"

#include "led.h"

#include "timer.h"

int main**(){**

LED\_init**();**

BUTTON\_init**();**

init\_timer**(**1**,**0XFF**);** //Edited:initial timer one with value 0xFF

enable\_timer**(**1**);** //Edited:enabling timer

**while(**1**);**

**}**

IRQ\_button.c:

#include "button.h"

#include "LPC17xx.h"

#include <stdbool.h> //Edited:Adding stdbool for using bool values in our code

extern insertionSort**(**int**,**int**);** //Edited:prototype of our ASM function

char array\_1**[**20**];** //Edited:Define some variable that we need in our code

char counter**=**0**;**

bool flag**=true;**

void EINT0\_IRQHandler **(**void**)**

**{**

**if(**counter**<**20**)**array\_1**[**counter**]=(**char**)**LPC\_TIM1 **->**TC**;** //Edited:whenever we press button we have an interrupt and if counter was below 20 we add the value of timer counter reg to array

**if(**counter**>=**20**){** //Edited:if it was above 20 then we turn of led 6 and turn off led 7 and with a flag we switch this situation

**if(**flag**){**

LED\_On**(**6**);**

LED\_Off**(**7**);**

**}**

**if(!**flag**){**

LED\_On**(**7**);**

LED\_Off**(**6**);**

**}**

**}**

counter**++;**

flag**=!**flag**;**

LPC\_SC**->**EXTINT **|=** **(**1 **<<** 0**);** /\* clear pending interrupt \*/

**}**

void EINT1\_IRQHandler **(**void**)**

**{**

LED\_Off**(**11**);** //Edited: if we press KEY1 then we turn off all leds

LED\_Off**(**6**);**

LED\_Off**(**7**);**

insertionSort**(&**array\_1**,**20**);** //Edited:we recall our ASM function and we sort array

LED\_On**(**11**);** //Edited:when we back from subroutine we turn on led 11

LPC\_SC**->**EXTINT **|=** **(**1 **<<** 1**);** /\* clear pending interrupt \*/

**}**

void EINT2\_IRQHandler **(**void**)**

**{**

LPC\_SC**->**EXTINT **|=** **(**1 **<<** 2**);** /\* clear pending interrupt \*/

**}**

Led.h:

/\* lib\_led \*/

void LED\_init**(**void**);**

void LED\_deinit**(**void**);**

void LED\_On**(**unsigned int num**);** //Edited: adding prototype of function

void LED\_Off**(**unsigned int num**);** //Edited: adding prototype of function

Lib\_led.c:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*--------------File Info---------------------------------------------------------------------------------

\*\* File name: lib\_led.c

\*\* Descriptions: Atomic led init functions

\*\*--------------------------------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "LPC17xx.h"

#include "led.h"

/\*----------------------------------------------------------------------------

Function that initializes LEDs

\*----------------------------------------------------------------------------\*/

void LED\_init**(**void**)**

**{**

LPC\_PINCON**->**PINSEL4 **&=** 0xFFFF0000**;** // PIN mode GPIO: P2.0-P2.7 are set to zero

LPC\_GPIO2**->**FIODIR **|=** 0x000000FF**;** // P2.0-P2.7 on PORT2 defined as Output

**}**

void LED\_deinit**(**void**)**

**{**

LPC\_GPIO2**->**FIODIR **&=** 0xFFFFFF00**;**

**}**

void LED\_On**(**unsigned int num**){**

LPC\_GPIO2 **->** FIOSET **|=** **(**1**<<(**11**-**num**));** //Set 11-num pin

**}**

void LED\_Off**(**unsigned int num**){**

LPC\_GPIO2 **->** FIOCLR **|=** **(**1UL **<<(**11**-**num**));** //Reset 11-num pin

**}**

Lib\_timer.c: because of lot of comments I copy the edited part of this file

LPC\_TIM0**->**MCR **=** 2**;** //Edited:we dont need INT so we change value from 3 to 2

// </h>

//\*\*\* <<< end of configuration section >>> \*\*\*

// NVIC\_EnableIRQ(TIMER0\_IRQn);

**return** **(**1**);**

**}**

**else** **if** **(** timer\_num **==** 1 **)**

**{**

LPC\_TIM1**->**MR0 **=** TimerInterval**;**

LPC\_TIM1**->**MCR **=** 2**;** //Edited:we dont need INT so we change value from 3 to 2 /\* Interrupt and Reset on MR1 \*/

// NVIC\_EnableIRQ(TIMER1\_IRQn);

**return** **(**1**);**

**}**

**return** **(**0**);**

**}**