

DIT 632

Written re-exam

2021-08-27

Examiner:

Mirosław Staron

mirosław.staron@gu.se

031 772 1081

General rules

This is an online exam, which means that the room opens at the exam time, you receive the questions, and you need to submit your answers before the exam ends. Answers cannot be submitted after the exam ends.

You are allowed to use the resources in the course and even from the internet, according to the following rules:

1. It is NOT allowed to collaborate on a solution with anyone → your answers must be individual.
2. Using materials from external source (internet, course web-page) requires reference to the source and description how you used the source (what you added/removed/modified) → your answers should be original, not copied from the internet. Failure to do so will be considered as plagiarism.
3. Copying more than 5 lines of code from an external source without modification and comments (see point #2 above), will be considered as plagiarism. If you copy part of the solution from your exercise, please provide the name of the group and the reference to which file for EVERY copied line.
4. Posting the question and/or receiving an answer at/from an internet forum or social media/collaboration platforms will be considered as unsolicited help → violate point #1 above; will be considered as cheating on the exam.

You should submit your answer by pasting your solution to this word document. This includes pictures of the solution (e.g. Arduino board), screenshots (e.g. TinkerCad, PowerPoint) or source code. If you want to attach a source code file, please .zip it with the exam and submit.

Using external links will not be accepted. For example, if you draw a diagram in TinkerCad, I expect either the picture of the board + source code and a downloaded project from TinkerCad; if you attach the link to TinkerCad project, I will not check it. This rule exist as there is no way to prevent modifications of the external links after the exam closes.

You are allowed to use the following tools when preparing the solution:

- C compiler, linker, programming environment → you submit .h/.c file(s), not the entire project
- TinkerCad → you submit the code and the screenshot
- Arduino board → you submit the code and the picture of your board (as a .jpg or .png, or pasted into the document)
- PowerPoint, dia, Visio → you submit the code and the .png/.jpg figure of the board

Please remember that it's your responsibility to make sure that your solution is readable and understandable!

Please remember that I will use the same grading as in the exercises – in particular, the comments in the code (all lines must be commented).

Good luck!

Total number of points: 35, for G (Pass) -> min. 17 points, for VG -> min. 30 points.

Question 1: Measuring temperature – 9 points

Create a system based on Arduino Uno or Arduino Yun, which measures the temperature and lids the LEDs. You should use 5 LEDs for the temperature measurement. The LEDs should be turned on depending on the temperature – e.g. for 0-5 degrees Celsius, 1 LED is turned on; for 5-10, 2 LEDs are turned on, and so on.

You should define for which temperature range the LEDs should be turned on, these should be provided as variables.

You should check for the temperature when a button is pressed, using interrupts. The button should be added to the board and when the user presses the button, the measurement is taken and the appropriate number of LEDs is turned on.

Your task is to:

- Create a board
 - with the right number of LEDs and buttons (2 points)
 - with the right wiring to prevent damage to components (1 points)
 - use different colors for the LEDs for the different temperature intervals (1 point)
- Write the code
 - Using interrupts (1 points)
 - Using the definitions (2 points)
 - Using comments (2 points)

Your solution should include:

- Screenshot/picture of the board
- Source code (in C)

Question 2: Bit packing and unpacking for the elevator controller – 7 points

The elevator controller uses one byte (8 bits) to store the status of the elevator. The values/meaning of the bits of the byte are:

Name	Bits	Info
engine_on no 7 (MSB)	1	Is engine on or off (the elevator moves or not). This is bit
floor_pos	4	Which floor number the elevator should go to (0-15)
door_pos	2	If the door is open or closed
brake1	1	Are the normal brakes on?

We should store them in a byte like this:

[engine_on]	[floor_pos]	[door_pos]	[brake1]
1 bit	4 bits	2 bits	1 bit

(8 bits in total)

Write a program **code.c** which takes 5 arguments (different number of arguments should result in an error message). The arguments should correspond to the values/variables above.

Example for a start of the program from command line:

```
code 1 7 1 1
```

The above should be treated as:

Name	Value

engine_on	1
floor_pos	7
door_pos	1
brake1	1

Your task:

- Write the function to pack these values together in a byte (unsigned char), and (2 points)
- Write the main function to take the arguments from the console, pack them into a byte and print it out to the console in hexadecimal form. (2 points)
- Make the program fail-safe, i.e. if it finds anything wrong (too many/few arguments, faulty input values) your program should print out an error message and exit (1 point)
- The code should be commented (2 points)

Question 3: Multithreaded property checker (Square of Sum = Sum of Cubes) – 7 points

A series of numbers with the property that their square of sum equals sum of cubes is such a list where:

$$(1 + 2 + 3 + \dots + n)^2 = 1^3 + 2^3 + 3^3 + \dots + n^3$$

Write a multithreaded program which calculates both sides of the equation for any number lower than n. For example, for 3, the program should calculate both sides of the equation for three series:

- 1
- 1, 2
- 1, 2, 3

There should be three threads in the program. The main thread should coordinate the work of two worker threads. The program should be executed using the following argument as example:

squares_calculator.exe 100

The output should be the list of the numbers with the result, e.g.:

```
1 - 1
2 - 9
3 - 36
...
```

After each thread finishes, the program should print the status, e.g.

2 out of 6 numbers checked

Your task is to:

- Write the functions for making the calculations of both sides of the equation (2 points)
- Write the multithreading logic (thread synchronization, printing status, printing result) (2 points)
- Comment the code (2 points)
- Find a way to reuse the results of previous calculations (for n) in the next loop (n+1) (1 point)

Question 4: Binary trees and memory allocation – 8 points

Below you have a program which handles the creation of a simple binary tree.

```

/*
 * This program demonstrates the data structure of a binary tree
 * Your task is specified in the exam
 */
#include <stdio.h>
#include <stdlib.h>

// This is one node of the tree
typedef struct {
    int data;           // data which is stored in the node
    void* left;        // pointer to the left sub-tree
    void* right;       // pointer to the right sub-tree
} Node;

/*
 * This function creates an empty nde
 */
Node * CreateNode(int val)
{
    Node *p;           // pointer to the new node

    // create the node
    p = (Node *) malloc(sizeof(Node));

    // store the value
    p->data = val;

    // Left and right child for node
    // will be initialized to null
    p->left = NULL;
    p->right = NULL;

    // return the new node
    return p;
}

/*
 * Main - create a simple tree
 */
int main()
{
    /*create root*/
    Node* root = CreateNode((int) 'X');

    // create two sub-trees
    root->left = CreateNode('ED');
    root->right = CreateNode(0xFFF);

    // and one more sub-tree
    ((Node *)root->left)->left = CreateNode(68);
    ((Node *)root->right)->left = CreateNode(1);

    return 0;
}

```

Your tasks:

A (2 points): Draw the tree for each step of its creation in the procedure main, starting with line below `/*create root*/`.

B (2 points): This program has a function to create a new node, but no function to delete a node. Please write a function that deletes a node and all its sub-trees only if the node has a specific value (e.g. 1).

C (2 points): This program contains errors in the main, please find these errors and correct them.

D (2 points): The tree is not deleted when the program ends, please write the code to delete the entire tree.

You should hand-in the solution to all tasks in a single .c file. Please use the comments to indicate which code you have added for each task.

Question 5: Multiprocessing – 4 points

Provide detailed examples, one for each of the following statements, to show that:

1. Multi-threaded applications are not a prerequisite to take advantage of a multi-core CPU
2. Multi-core CPUs are not a prerequisite to take advantage of multi-threaded applications.