

DIT 632

Written exam (example solutions)

2021-03-16

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 light and temperature – 8 points

Create a system based on Arduino Uno or Arduino Yun, which measures the light intensity and temperature at the same time. The systems should monitor the dependency between these two measurements and warn about the deviations.

Normal dependencies, all other variables are treated as deviating values:

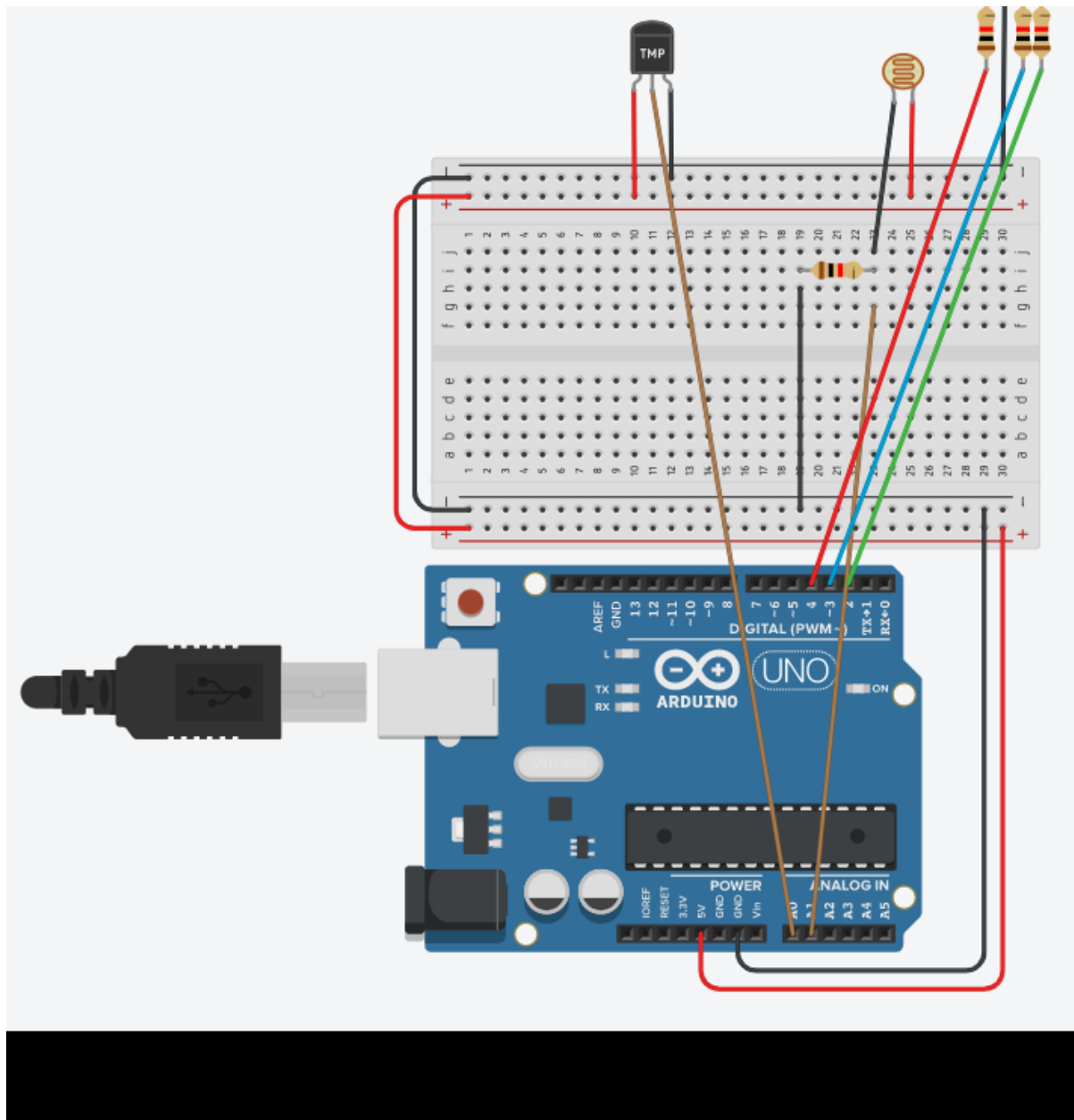
Temperature	Light intensity
< -12 °C	0 %
-12 °C - 0 °C	1% - 20%
0 °C - 20 °C	21% - 60%
>= 21 °C	61% - 100%

The system should use Interrupt Service Routine for periodically reading the temperature and light intensity (periodicity, in seconds, should be provided as a variable in the program).

The system should use two color LEDs to indicate the normal dependency (GREEN), deviation when the temperature is higher than it should be, given the Light intensity (RED), and the deviation when the temperature is lower than it should be (YELLOW or BLUE).

Your solution should include:

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



Question 2: Sending a list over sockets – 8 points

Create one client and one server programs that can send/receive a register of persons over the internet. The person register is a linked list of the records below.

Structure to use as a person:

```
#define MAX 10

typedef struct {
    char sName[MAX];           // Person's given name
    char sSurname[MAX];        // Person's surname
    char sPnr[MAX];            // Person's social security number
    char sAddress[MAX];         // Person's city of residence
} Person;
```

Your program should contain the code for both the client and the server. Both programs should use the same .h file with the definition of the structure and the function for creating a linked list. The code for the function for creating the list should be in a separate .c file (separate from the file which contains main()) that is compiled together with the code for the server and the client.

Server:

The server should create a linked list with different persons defined by you. The linked list should be created dynamically, i.e. each record should be a separate instance of the Person structure with the pointer to the next record. Using static arrays is not allowed.

The server program should print the list on the console once the list is created.

When the client connects to the server, it should send the number of persons in the register and then send each person separately.

You should test the program with 1 and 5 persons in the register. The server program should create these persons.

Client:

The client should connect to the server, receive the number of persons in the register from the server and then receive a linked list with each person. The printed list should contain all parameters in the structure and the age of the person. The age should be calculated by the client based on the current date and the person's social security number. Then the client should print the linked list on the console.

Question 3: Multithreaded elevator – 8 points

Create a multithreaded elevator program. The main thread should be responsible for the interaction between the user and the elevator. The child thread should control the elevator's movement.

The elevator should move between 0 and MAX floors (MAX is defined by you, should be between 10 and 20). The travel time between each floor is 1 second. When the elevator reaches the desired floor, a message is printed on the console. When the elevator moves between the floors, a message with the elevator's position should be printed to the console when passing a floor and once in-between every floor.

The user can choose the floor where he/she wants to travel only when the elevator is waiting at one of the floors. When the user chooses a new floor while the elevator is in motion (moving to the destination floor), an error message should be printed.

The user can always choose to print the status of the elevator. The output of the status should be: `waiting_for_user` or `moving`.

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) 'A');

    // create two sub-trees
    root->left = CreateNode('2');
    root->right = CreateNode(0x41);

    // and one more sub-tree
    ((Node *)root->left)->left = CreateNode(67);

    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): There is a memory leak in this program. Please find it, describe why the memory leak happens and suggest a solution.

C (2 points): This program is not fail-safe. For example, we can create a node that overwrites another node. Please identify at least 3 different problems in this program and add the fail-safety checks for these problems.

D (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.

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 (or describe the memory leaks for task B).

Question 5: Multiprocessing – 3 points

Which of the following statements are true and which of them are false? Please motivate your answer.

- After reducing the sequential part of her code to 20% and using 4 cores instead of 1 for the rest of the code, Maria's program ran 3 times faster compared to the original code.
- After reducing the sequential part of her code to 20% and using 8 cores instead of 1 for the rest of the code, Maria's program ran 3 times faster.
- After reducing the sequential part of her code to 40% and using 4 cores instead of 1 for the rest of the code, Maria's program ran 2 times faster.

Answer:

ANSWER HINT: check with Amdahl's law... E.g. $1/(0.2+0.8/4)=2.5 < 3 \rightarrow$ so first sentence is false

The second sentence is true

The third sentence is false