Instructions for the exam

DIT 633 - Development of Embedded and Real-time systems

This exam should be an individual work for you. You are not allowed to use any outside help.

If you are allowed to use a compiler, there is a link to an online one, which will open in a separate window. You can test the code in the online compiler, but **you must remember to copy-paste it back to the exam**, otherwise your code will disappear once you close the window.

The same is true for TinkerCad, please remember to copy-paste the code from TinkerCad to the exam.

You are NOT allowed to access code that was not written during the exam (e.g., from your previous assignments). This will be considered plagiarism.

You are not allowed to copy code from your colleagues or any other external source.

Remember: In programming questions, if the code does not compile, you get 0 points for the question!

Grading scale:
50% correct - 3
65% correct - 4
85% correct - 5

Good luck!

/Miroslaw
031 772 1081

1 Reading pointers

int (x)[]**

- x is pointer to pointer to array of function that returns an int
- x is pointer to pointer to array of int
- x is an array of pointer to pointer to int
- x is a function that returns a pointer to pointer to array of int

int (x[])()**

- x is a function returning array of pointer to pointer to int
- x is an array of function returning pointer to int
- x is an array of pointer to pointer to int
- x is an array of pointer to pointer to function returning int

int (*(*(*x)(void)))

- x is pointer to a pointer to a function (void) returning pointer to pointer to int
- x is a function (void) returning pointer to pointer to int
- x is a function (void) returning pointer to pointer to a pointer to int
- x is a pointer to function (void) returning pointer to pointer to int

int x(int *())

- x is a function that takes as an argument a function returning pointer to int, and returns int
- x is a function (function returning int) returning pointer to int
- x is a function (function returning pointer to int) returning pointer to int
- x is a function (function returning array of pointer to int) returning int

Totalpoäng: 0

2 Sustainability

Carbon emissions and carbon footprint are important aspects of sustainability, but we often forget to think about that when we design our software.

Given the lectures about sustainability in DIT633, please provide 4 suggestions of how to design sustainable software. You can focus both on the mechanisms or patterns in the software and the way in which software is developed, e.g., how we can measure carbon footprint.

Grading:
4 points - one point for every suggestion.

Skriv in ditt svar här

Teckenf...

Ord: 0

Totalpoäng: 4

3 Memory handling

Google was changing their algorithm for handling search. They've heard about the binary trees, which seemed to be the best to keep track of the data.

However, they did not manage to get the implementation complete. They asked you, the students of DIT633, to help them to finish up the program.

Here is the code that they use:

```
#include <stdio.h>
#include <stdlib.h>

// Define the structure for a tree node
typedef struct TreeNode {
    int data;
    struct TreeNode* left;
    struct TreeNode* right;
} TreeNode;

// Function to create a new tree node
TreeNode* createNode(int data) {
    // TODO: create the node here
}

// Function to insert a node into the binary tree
TreeNode* insertNode(TreeNode* root, int data) {
    // TODO: create the first node

    if (data < root->data) {
        root->left = insertNode(root->left, data);
    } else {
        root->right = insertNode(root->right, data);
    }
    return root;
}

// Function for in-order traversal of the binary tree
void inorderTraversal(TreeNode* root) {
    if (root == NULL) {
        return;
    }
    inorderTraversal(root->left);
    printf("%d ", root->data);
    inorderTraversal(root->right);
}
```

```
// Function to free the allocated memory for the tree
void freeTree(TreeNode* root) {
    // TODO: free the memory for the entire tree
}

int main() {
    TreeNode* root = NULL;
    int choice, value;

    while (1) {
        printf("\nBinary Tree Operations:\n");
        printf("1. Insert\n");
        printf("2. In-order Traversal\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                root = insertNode(root, value);
                break;
            case 2:
                printf("In-order Traversal: ");
                inorderTraversal(root);
                printf("\n");
                break;
            case 3:
                freeTree(root);
                printf("Exiting...\n");
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
    return 0;
}
```

Your task is to finish the program - the places are marked with TODO: comments. Being this DIT633, you must also comment the code (only the new code that you write + the entire body of the function **insertNode**).

You can use Online Compiler for this question.

Grading:
- correct implementation of createnode: 3 points
- correct implementation of freeTree: 3 points
- correct implementation of insertNode: 2 points

- comments: 2 points

Skriv in ditt svar här

Totalpoäng: 10

4 Finding strings

Please write a program that contains a function for checking if a string contains a substring.

The signature of the function should be like this:

int contains_substring(const char *stringone, const char*substring);

The function should return:
* position of the substring, or
* -1 if the substring is not found

You are NOT allowed to use any build-in or library functions for handling strings

You can use the online C compiler for this task: Online C compiler

Your program should include:
- write the function (2 points)
- main() procedure to test this function (2 points)
- comments (2 points)

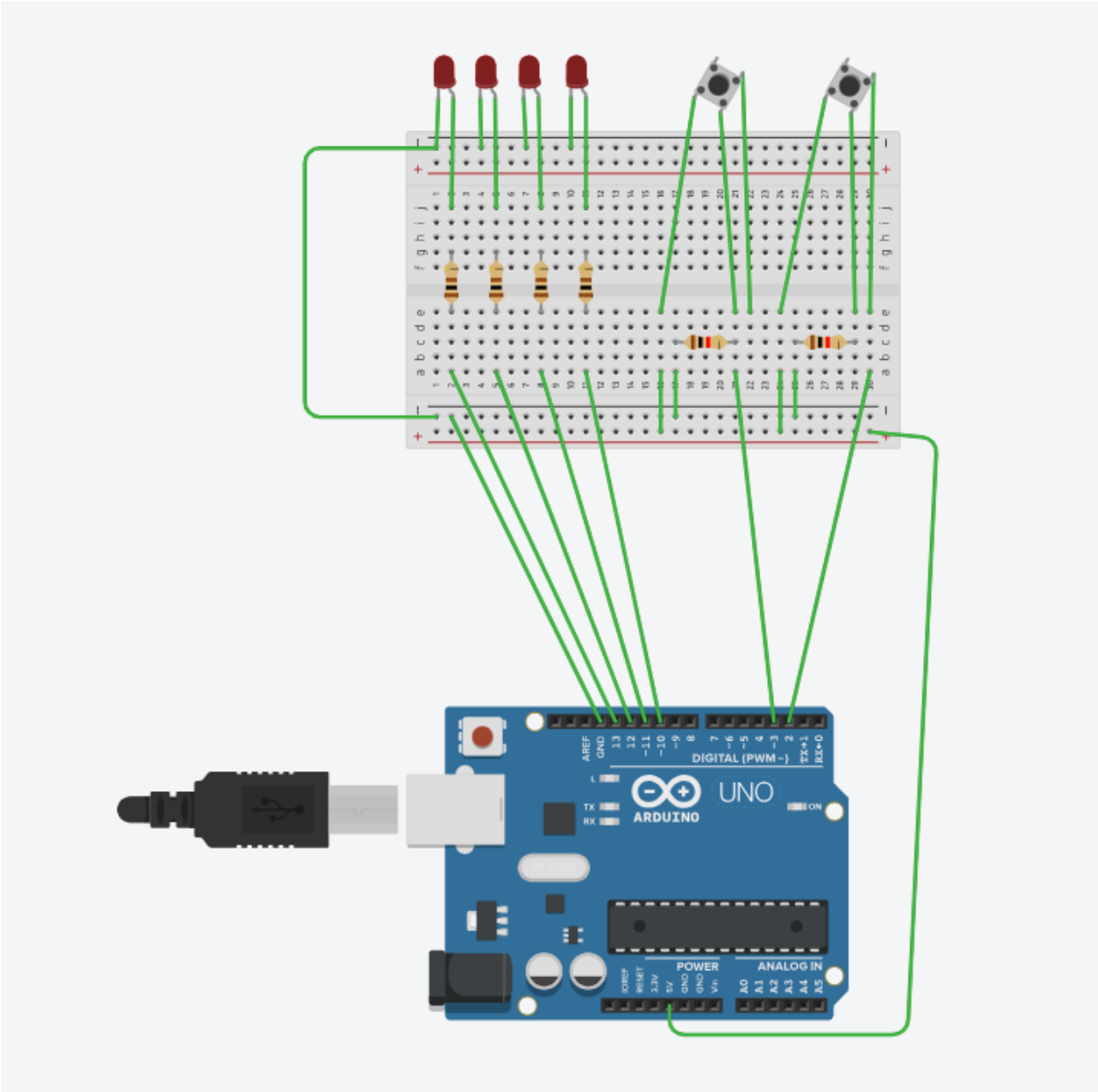
Skriv in ditt svar här

```
1
```

Totalpoäng: 6

5 Binary game in Arduino

Please use the following diagram in this question:



I would like you to use this circuit to implement a mini game.

The game is as follows:

1. Each player has his/her own button.
2. They need to press the button every other time - once per turn. If a wrong button is pressed, the player gets -1 point.
3. Once a button is pressed, the system should randomly generate a number between 1 and 4.
4. The LEDs should be turned on based on that generated number.
5. Once both players generate their numbers, the system should compare them. The player who generated a higher number gets +1 point.
6. The LEDs should be turned off after 2 seconds and the next player can press the button.
7. The system should write the current result after every turn to the serial port.
8. The game ends after a given number of turns (should be defined using the preprocessor directive).

You can use TinkerCad for this question here. Please remember to paste your code below!

Grading:
1. Implementing the correct random generation - 2 points
2. Implementing the game (sequence of pressing buttons, LEDs, timings) - 2 points
3. Using interrupts for the buttons - 2 points
4. Implementing the negative points - 1 point
5. Implementing printing to the serial port - 1 point
6. Commenting the code - 2 points

Skriv in ditt svar här

```
1
```

Totalpoäng: 10

6 Robot Race

A robot race

Before the 2024 Olympic Games, AI was asked about which new discipline we should add. AI, being AI, proposed robot games as a new game. The Olympic Committee agreed to that, at least to take it as the beta-test phase.

AI came up with the following game - AI Robot Race.
Each race has four players that play 10 rounds. Each player gets to randomly generate a 16-bit number in every round. The 16 bits are described as follows (from the MSB - Most Significat Bit):
- player ID: 2 bits
- speed multiplier: 3 bits
- forward or reverse gear: 1 bit
- number of steps: 9 bits

The game ends after 10 rounds and the robot with the longest travelled distance wins.

The program should print out the status of each player after each round: 1) the randomly drawn number, 2) the distance travelled, 3) all game parameters calculated from the random number (ID, speed multiplier, forward or reverse, number of steps).

You can use Online Compiler in this question.

You must use bit artihmetic/shifting for the question!!

Grading:
- correct functionality for a single Robot: 3 points
- correct funcitonality for the game: 3 points
- printing the status: 2 points
- commenting the code: 2 points

Skriv in ditt svar här

```
1
```

Totalpoäng: 10