

i Instructions for the exam

DIT 632 - 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.

If you access the code from your saved documents in TinkerCad, and use it in the exam, you **MUST** reference that code and describe clearly what you copied to the exam.

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 - G

85% correct - VG

Good luck!

/Miroslaw

031 772 1081

¹ Paging and addressing

Please explain the concepts of **paging** and **addressing** of memory.

Your answer should contain:

- 1) Explanation of these two concepts (2 points)
- 3) An example of how a physical address is calculated from a virtual address (2 points)

Your answer:

Maximum marks: 4

2 Reading pointers X

Please choose the right interpretation of "X" in each of the statements:

int *x;

- ☐ x is a pointer to function that returns a variable of type int
- ☐ x is a pointer to a variable of type int
- ☐ x is a variable of type int
- ☐ x is a reference to the variable of type int

int (*x[])();

- ☐ x is a pointer to a function that takes as an argument an array of integers
- ☐ x is a function which takes as an argument an array of pointers to variables of type int
- ☐ x is a function that takes an array as an argument and returns a pointer to int
- ☐ x is an array of pointers to functions that return int

char *(*(*x[][8]) ())[];

- ☐ x is array of array of 8 pointers to pointer to function returning pointer to array of pointer to char
- ☐ x is array of array of 8 pointers to functions returning pointer to array of pointer to char
- ☐ x is a function that takes as input an array of 8 pointers to pointer to an array and returns a pointer to array of pointer to char
- ☐ x is an array of 8 pointers to pointer to function returning pointer to array of pointer to char

int (*(*x) []) ();

- ☐ x is a function that takes as argument an array of pointers to functions and return a pointer to int
- ☐ x is a pointer to an array of pointers to functions returning an int
- ☐ x is a function that takes as an argument a pointer to an array of pointers and returns a pointer to int
- ☐ x is an array of pointers to functions that take no arguments and return pointers to int

char (* x ()) [20];

- ☐ x is a function returning a pointer to an array of 20 elements of type char
- ☐ x is an array of 20 pointers to functions returning char
- ☐ x is an array of pointers to functions returning pointers to functions returning pointers to char
- ☐ x is a pointer to a function returning a pointer to an array of 20 elements of type char

Maximum marks: 5

3 Enum calendar

Write a calendar program using enums.

Your task is to write a program that will use separate enums for days of the week (Mon, Tue, etc) and Weeks (week 1, week 2, ..., week 5). The program should increment the day every second. After Sunday, the week should increment as well and the counting should start from the beginning. Once it gets to week 5, it should start from the beginning.

The program should ask the user to provide the start day and week.

Every time the program increments the day, it should write the week and day on the console, e.g.:

Week 1, Monday

Week 1, Tuesday

...

You are NOT allowed to use strings to store the days and weeks.

In this question, you can use [the online compiler](#).

Points:

- using enums: 1p
- correct increment of weeks and days: 2p
- using interrupts on timer: 2p
- checking for correct input: 1p
- comments: 1p

Fill in your answer here

1

--	--

<http://onlinegdb.com>

Maximum marks: 7

4 Code quality

The following program has some problems. it is supposed to return 1 if a number is prime and 0 if it is not. However, it does not work.

Your task is to find all bugs, fix them and explain what the problems were and how you fixed them.

In this question you can use [the online compiler](#).

Your task is to:

- write the main function that will test the `is_prime()` function - 1p
- fix the mistakes - 3p
- comment the program - 1p

```
int is_prime(unsigned int n)
{
    unsigned int p;
    if (!(n & 1) || n <= 2 ) return n = 2;
    /* comparing p*p <= n can overflow */
    for (p = 3; p <= n/p; p += 3)
        if (!(n % p)) return 0;
    return 1;
}
```

Fill in your answer here

1

Maximum marks: 5

5 Bits and bytes

Write a program that counts the number of 0s and 1s in the binary representation of a decimal number.

Your program should ask the user to provide a decimal number. Then it should print the decimal number, it's binary representation and then count the number of 0s and 1s.

The interaction with the program should look like this:

Please provide a number: <user inputs a number here>

Number: 3

Binary: 0b11

Number of 0s: 0

Number of 1s: 2

In this task, you must use bit operations. You are not allowed to convert the number to strings or arrays of any sort. Leading 0s should not be counted (e.g., 0b0011 should return "Number of 0s: 0; Number of 1s: 2").

In this question you can use [the online compiler](#).

Points:

- conversion to binary: 1p
- counting 1s and 0s: 2 p
- comments: 2p
- check for correct input: 1p

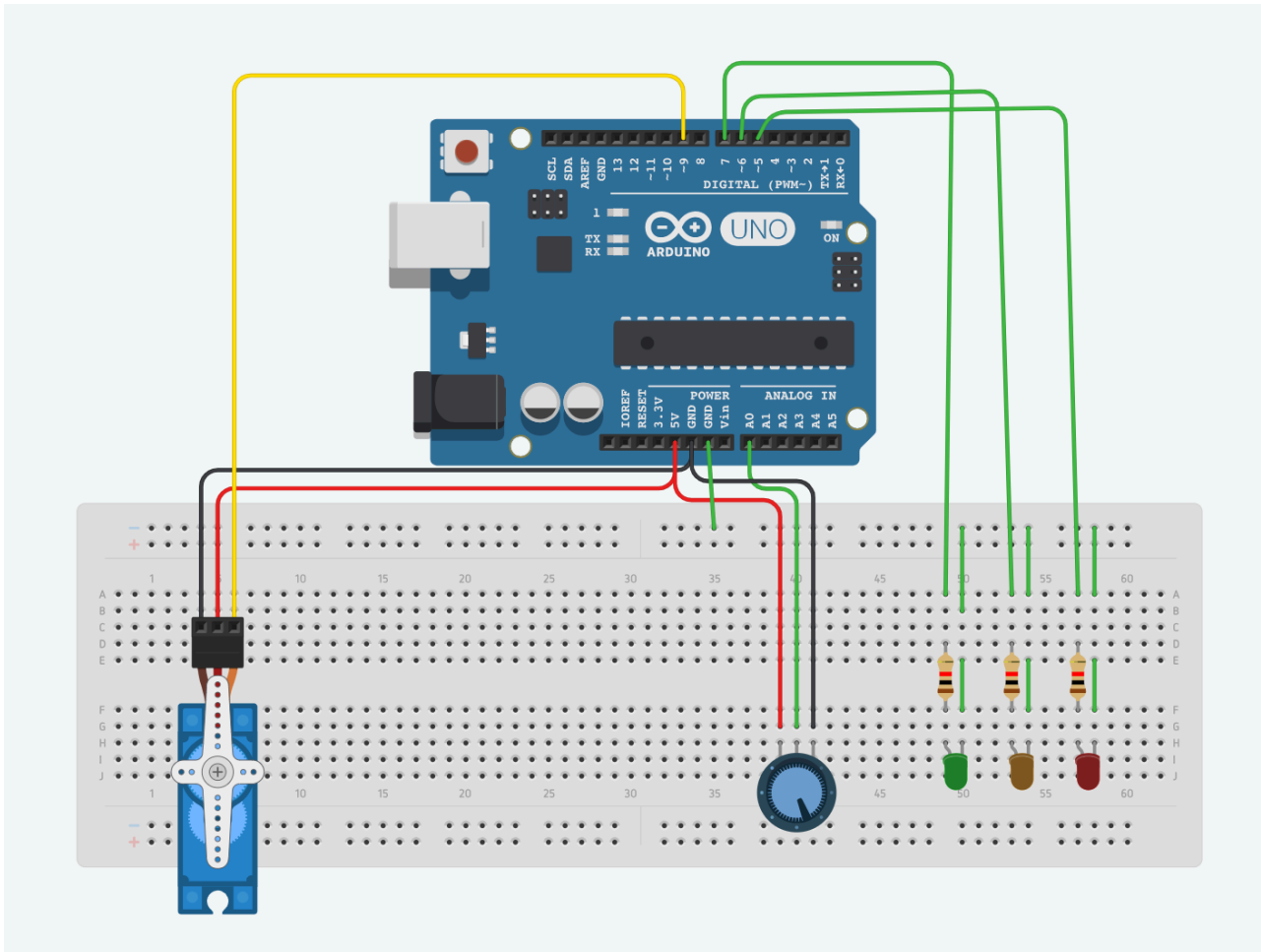
Fill in your answer here

1

--	--

Maximum marks: 6

6 Arduino board



The figure above contains a board with three LEDs, one potentiometer and one servo motorn.

Your task is to program the board so that the potentiometer steers the servo and the LEDs, in the following way.

1. The green LED should be on once the board starts. If the potentiometer is not active, then it should be lid constantly.
2. Once the potentiometer is active, i.e., servo is steered by the potentiometer, the green LED should blink (with 0.5 seconds interval).
3. Once the potentiometer/servo turns 90 degrees, the orange LED should be turned on.
4. Once the potentiometer/servo turns 160 degrees, the red LED should be turned on.

In this question you can use [TinkerCad](#).

You can get the following points:

- making the servo turn with the potentiometer - 2p
- correctly programming orange and red LEDs - 2p
- correctly programming the green LED - 2p
- commenting the code - 2p
- using interrupts - 2p

Fill in your answer here

--	--

Maximum marks: 10