STUDENT

# DIT633-0030-YTW

TENTAMEN

# DIT633 Development of embedded and Real-Time Systems, 2023-03-16 NOTE START TIME IS 14:00, EXAM OPENS AT 12:00 ONLY FOR STUDENTS WITH EXTRA TIME AIDS

| | |
|---|---|
| Kurskod | -- |
| Bedömningsform | -- |

| | |
|---|---|
| Starttid | 16.03.2023 12:00 |
| Sluttid | 16.03.2023 18:00 |
| Bedömningsfrist | -- |
| PDF skapad | 02.04.2024 12:22 |
| Skapad av | Maria Magnusson |

## 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.

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 - 3
65% correct - 4
80% correct - 5

Good luck!

/Miroslaw
031 772 1081

# 1 Reading pointers

**What is foo in the following expression: char (*foo)(int *, int*)**

☑ Foo is a pointer to a function that takes two parameters and returns a char. ✓

□ Foo is a pointer to a function that takes two parameters and returns a pointer to a char

□ Foo is a pointer to a pointer to a function that takes two parameters and returns a char

□ Foo is a function that has two parameters and returns a pointer to a char

**What is x in the following statement: int *a, b; char *y, x;**

□ a pointer of a variable of type char

☑ a variable of type char ✓

□ pointer to a variable that points to a variable of type char

□ variable that points to a pointer of type char

**foo in the expression: int *(*foo)(int *) is:**

☑ pointer to a function that returns a pointer to an int ✓

□ pointer to a pointer to a function that returns an int

□ function that returns a pointer to an int

□ function that returns a pointer to a pointer to an int

**Which of the following expressions is allowed (and correct) in C:**

□ char *p; int *x; p = x; ✓

□ char *p; char x[]; p = &x;

□ char *p; char x; p = x;

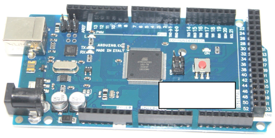
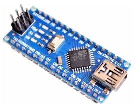
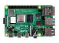
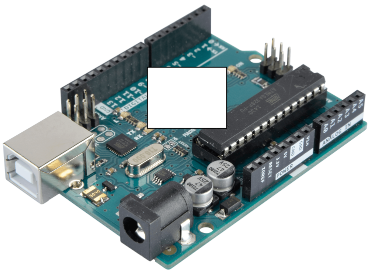
☑ char *p; char &x; p = x; ✗

Delvis rätt. 3 av 4 poäng.

## 2 Boards

Ersätt med din uppgiftstext...

**Which board is presented in the picture**

| |  |  |  |  |
|---|---|---|---|---|
| Arduino nano | ○ | ○ ✓ | ● ✗ | ○ |
| Arduino uno | ○ | ○ | ○ | ● ✓ |
| Raspberry Pi | ○ | ● ✗ | ○ ✓ | ○ |
| Arduino Mega | ● ✓ | ○ | ○ | ○ |

Delvis rätt. 2 av 4 poäng.

## 3 Sustainability

Please describe two ways in which a software (and hardware) system can impact environment. For each of the ways, please describe how to reduce the environmental impact of the software.

In this question, you can relate to either the embedded part of the system or the front-end part (as it was discussed in the lecture).

**Skriv in ditt svar här**

The software system impacts the environment in many ways.

To learn embedded systems you need a lot of spare parts and arduino materials to get started. In the future we can see that these parts gets thrown away as they no longer in need. This leads to pollution as they are not reused but rather just used for learning and then thrown away. And there are many students who are trying to learn embedded system which leads to more waste.

This can be solved using tinkercade instead which is an online tool to learn embedded system without the hardware parts.

Another way the software system can impact the environment is that most of the hardware part for embedded systems are not biodegradable( made out of plastic). This means it is really harmful for the planet if they are thrown away in the water ways which can affect marine lives severely. The marine lives are only in danger and many of the animals are getting extinct.

This can be solved by making more people aware of the impact they cause to the planet and reusing the hardware materials instead of throwing them away.

Ord: 192

Besvarad.

## 4 Arrays

Write a program in C that reads 10 strings from a console, stores them in an array and finds the shortest string in the array.

The program should contain a function that takes as parameter the array, find the shortest string, remove it and return the array that is shorter by one element.

The program should read the strings from the console.

The program should have a separate function for finding and removing the string.

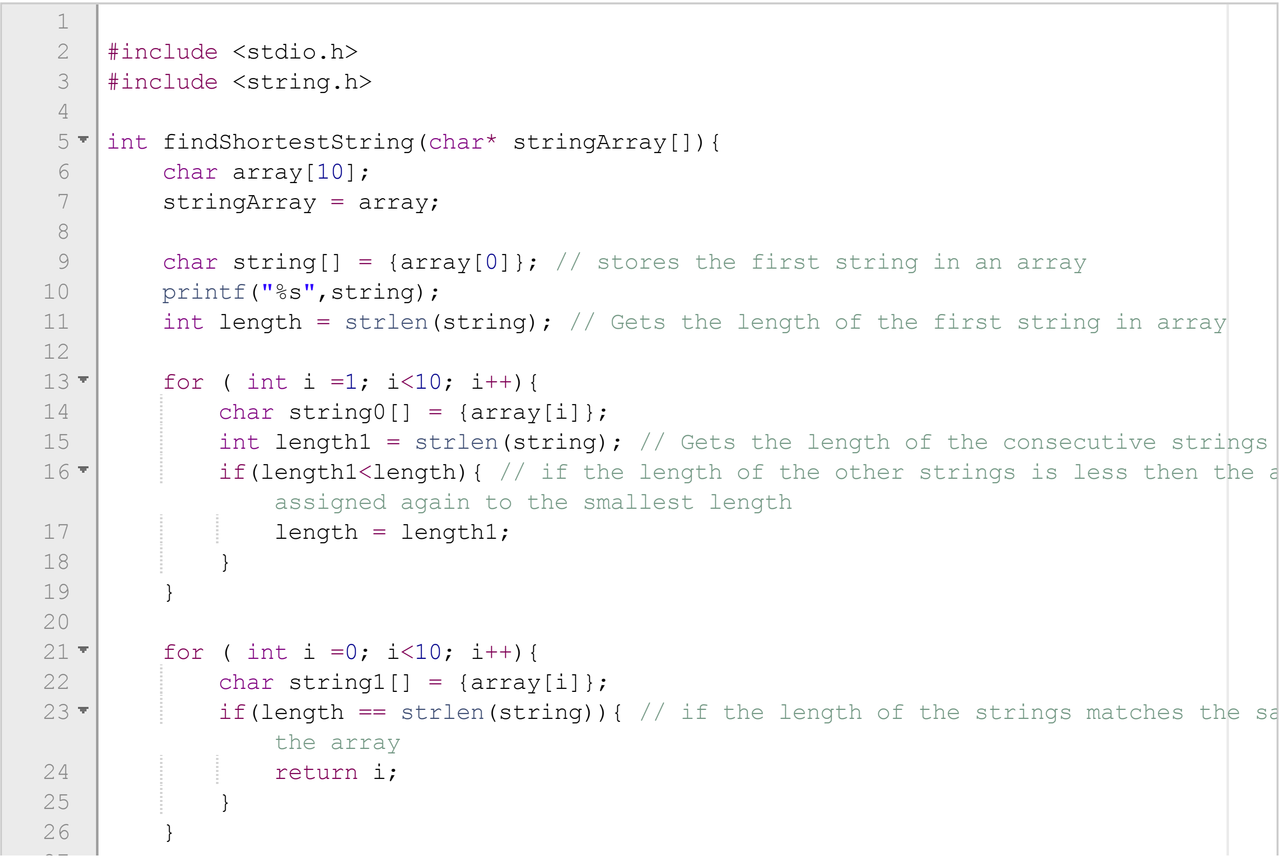
The program should write the string that has been removed to the console. The program should also write all elements of the array before and after removal of the shortest string.

You can use the online compiler for this question here: [www.onlinegdb.com](http://www.onlinegdb.com)

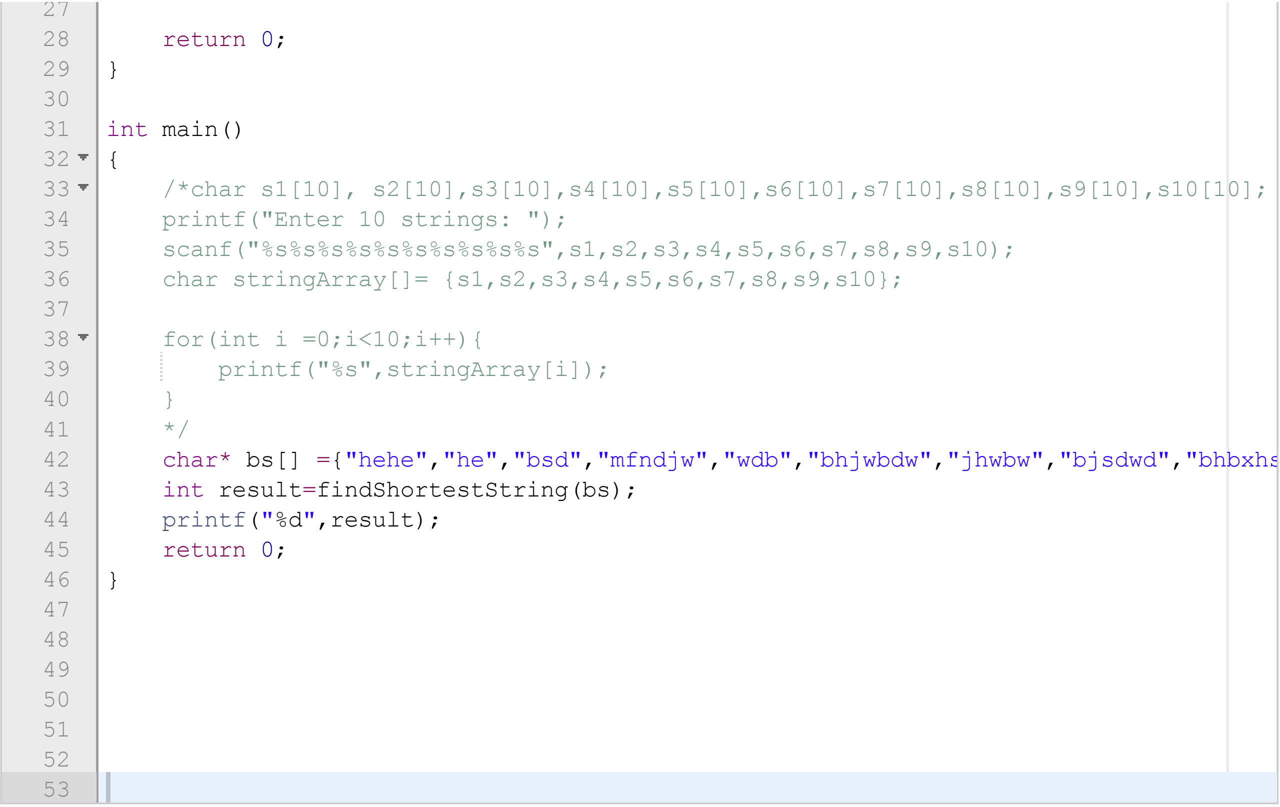
The program should contain the following:
* correct functionality (as specified above) - 3 points
* comments - 3 points
* function to find and remove the element - 2 points
* main to test the program - 2 points
* safety checks - 2 points

Please remember to paste the code from onlinegdb to the form below.

**Skriv in ditt svar här**



```

#include <stdio.h>
#include <string.h>

int findShortestString(char* stringArray[]){
    char array[10];
    stringArray = array;

    char string[] = {array[0]}; // stores the first string in an array
    printf("%s",string);
    int length = strlen(string); // Gets the length of the first string in array

    for ( int i =1; i<10; i++){
        char string0[] = {array[i]};
        int length1 = strlen(string); // Gets the length of the consecutive strings
        if(length1<length){ // if the length of the other strings is less then the a
            assigned again to the smallest length
            length = length1;
        }
    }

    for ( int i =0; i<10; i++){
        char string1[] = {array[i]};
        if(length == strlen(string)){ // if the length of the strings matches the sa
            the array
            return i;
        }
    }
```



```
    return 0;
}

int main()
{
    /*char s1[10], s2[10],s3[10],s4[10],s5[10],s6[10],s7[10],s8[10],s9[10],s10[10];
    printf("Enter 10 strings: ");
    scanf("%s%s%s%s%s%s%s%s%s%s",s1,s2,s3,s4,s5,s6,s7,s8,s9,s10);
    char stringArray[]= {s1,s2,s3,s4,s5,s6,s7,s8,s9,s10};

    for(int i =0;i<10;i++){
        printf("%s",stringArray[i]);
    }
    */
    char* bs[] ={"hehe","he","bsd","mfndjw","wdb","bhjwbdw","jhwbw","bjsdwd","bhbxhs
    int result=findShortestString(bs);
    printf("%d",result);
    return 0;
}
```

Besvarad.

## 5 Code quality

One of my colleagues wrote this program. However, I do not understand what it does.

Your task is to:
1) Comment the code explaining what the code does - in terms of the algorithm, not what each statement does (2 points)
2) Rewrite this program so that it uses meaningful names of the functions and variables (4 points)
3) Comment each statement - 2 points
4) Expand the main() function to test the rewritten function in a loop - 2 points

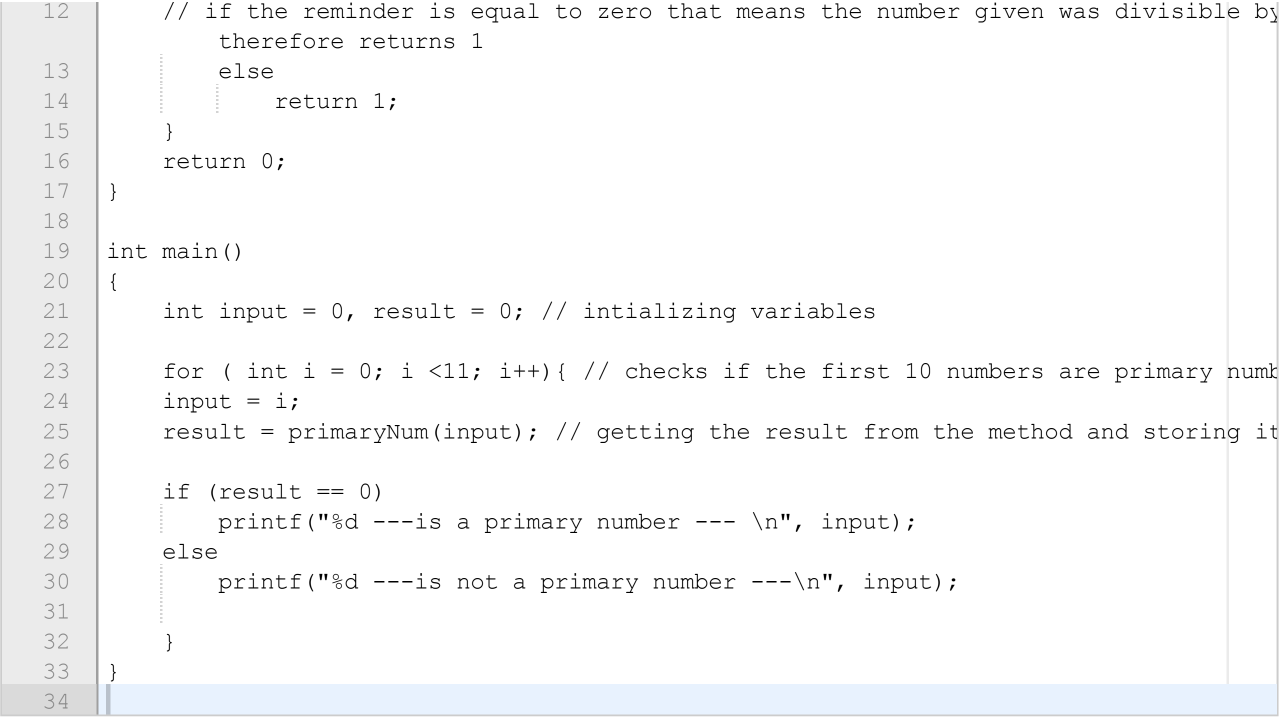
You can use the online compiler for this question: www.onlinegdb.com

Please remember to paste the code back to the form below!

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

int foo(int x)
{
    for (int i = 2; i <= x / 2; i++) {
        if (x % i != 0)
            continue;
        else
            return 1;
    }
    return 0;
}

int main()
{
    int a = 7, b = 0;

    b = foo(a);

    if (b == 0)
        printf("%d --- explain what it is --- ", a);
    else
        printf("%d --- explain what the opposite condition is ---", a);
}
---
```

**Skriv in ditt svar här**

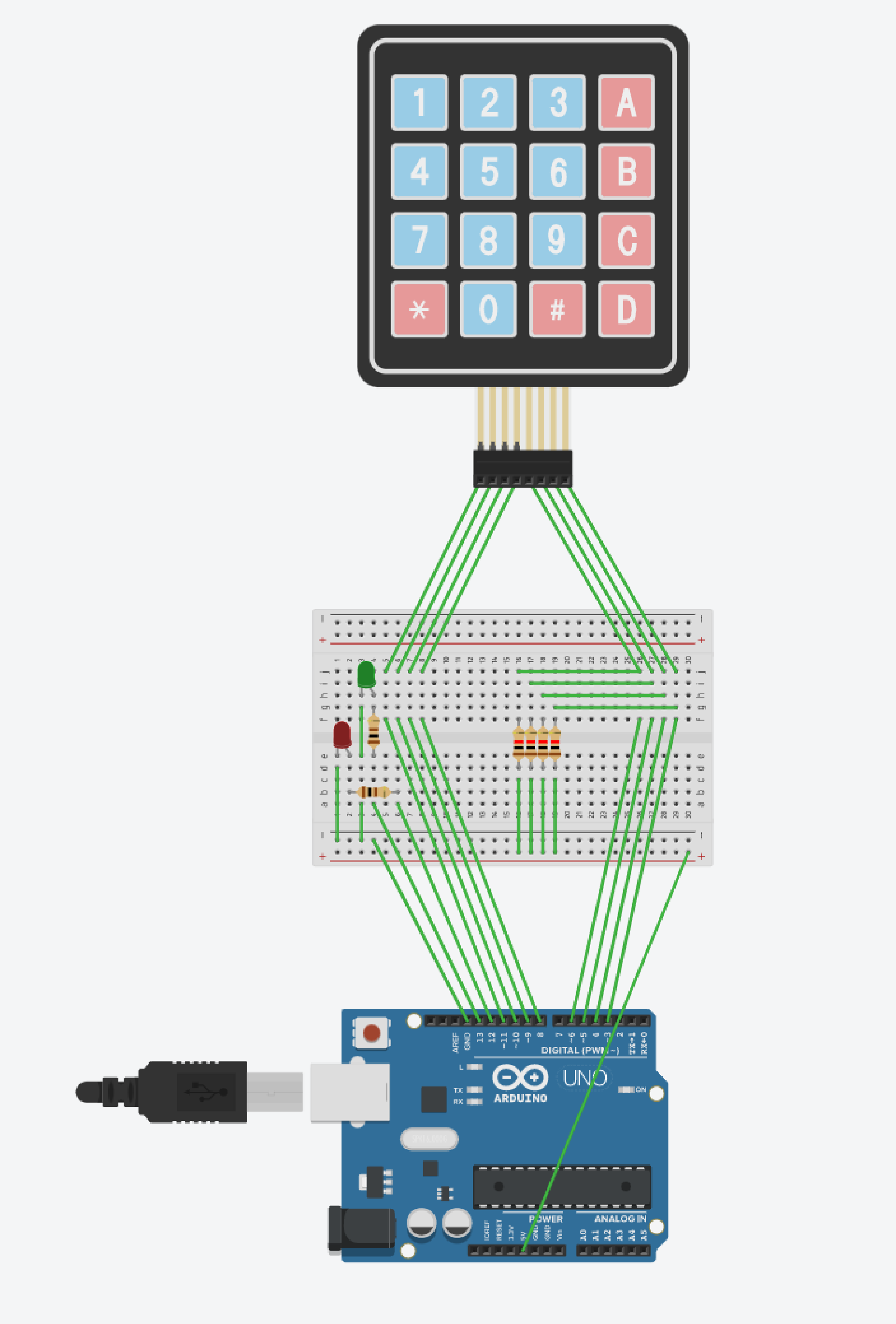
```

#include <stdio.h>

int primaryNum(int x)
{
    // checks if the given number is not divisible by
    // other numbers(starting with 2) than itself, if true return 0
    for (int i = 2; i <= x / 2; i++) {
    // if the reminder is not equal to zero the loop continues and keeps on checking
        if (x % i != 0)
            continue;
```



```
    // if the reminder is equal to zero that means the number given was divisible by
        therefore returns 1
        else
            return 1;
    }
    return 0;
}

int main()
{
    int input = 0, result = 0; // intializing variables

    for ( int i = 0; i <11; i++){ // checks if the first 10 numbers are primary numb
    input = i;
    result = primaryNum(input); // getting the result from the method and storing it

    if (result == 0)
        printf("%d ---is a primary number --- \n", input);
    else
        printf("%d ---is not a primary number ---\n", input);

    }
}

```

Besvarad.

## 6 Locker



The figure above contains the circuit of a locker system. You should write the software that will provide the functionality for this system.

The program should do the following. The user should be able to provide a 6 digit code. If the code is correct, then the system should unlock - this means that the green LED should turn on.

If the code is incorrect, then red LED should turn on.

If the user enters incorrect code three times, the system should lock for 30 seconds. During that time, the red LED should be turned on.

The user should be able to cancel entering of the code if the user makes a mistake. For example, if the user enters 3 digits and notices that he/she made a mistake). This should be done by pressing the button "C". Once the user presses "C", he/she should be able to start entering the code again.

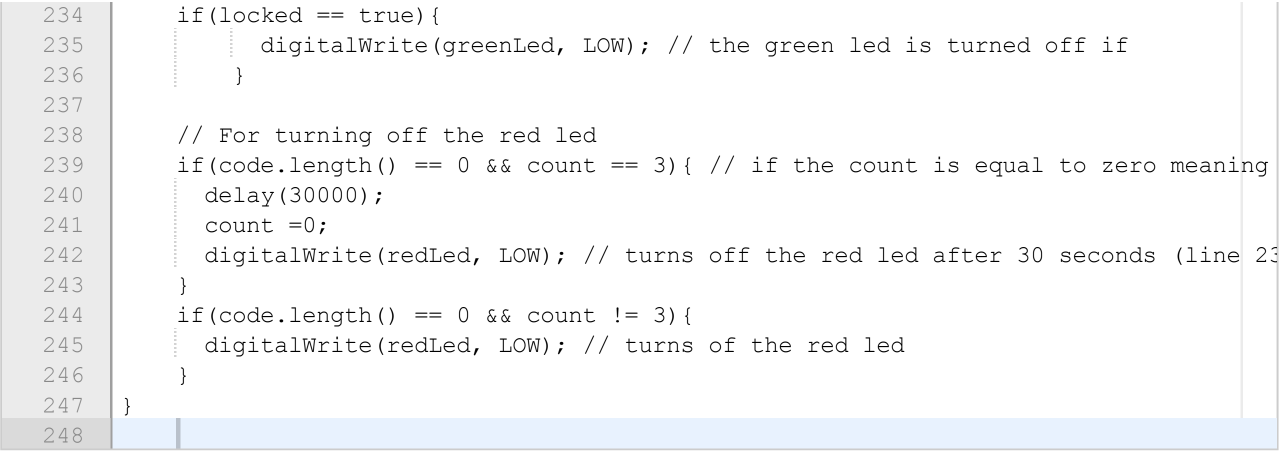
When the system is unlocked, the user should lock it again by pressing the button "D".

You can use tinkercad.com for this question. Please remember to copy the code from tinkercad.com into the form below.

Your program should:
1) unlock the system - 3 points
2) lock the system - 1 point
3) reset entering of the code - 2 points
4) handle 3x incorrect code - 2 points
5) be commented - 2 points
6) use #define or const for defining which code unlocks the system - 1 point
7) use separate functions whenever possible - 2 points

**Skriv in ditt svar här**

```
// The manual keypad was copied from my previouse work for one of the workpackages

const int row1 = 11; // digital pin for steering row 1
const int row2 = 10; // digital pin for steering row 2
const int row3 = 9;  // digital pin for steering row 3
const int row4 = 8;  // digital pin for steering row 4

const int col1 = 7; // digital pin for column 1
const int col2 = 6; // digital pin for column 2
const int col3 = 5; // digital pin for column 1
const int col4 = 4; // digital pin for column 1

const int redLed = 13; // digital pin for red led
const int greenLed = 12; // digital pin for green led

const String password = String("123456"); // this is the password to unlock

String code = String(""); // this will store the code when the buttons are pressed

int count = 0;  // This counts the number of mistakes

bool locked = true; // boolean to know if its locked or not

void setup()
{

    pinMode(row1, OUTPUT); // send signal through row1
    pinMode(row2, OUTPUT); // send signal through row2
    pinMode(row3, OUTPUT); // send signal through row3
    pinMode(row4, OUTPUT); // send signal through row4

    pinMode(col1, INPUT); // receive signal through column1
    pinMode(col2, INPUT); // receive signal through column2
    pinMode(col3, INPUT); // receive signal through column3
    pinMode(col4, INPUT); // receive signal through column4

    pinMode(redLed, OUTPUT);  // sets the red led to output
    pinMode(greenLed, OUTPUT); // sets the green led to output

    Serial.begin(9600);
}

void loop()
{
    digitalWrite(row1, LOW);      // send a signal through row 1
    int r1c1 = digitalRead(col1); // read column 1
    int r1c2 = digitalRead(col2); // read column 2
    int r1c3 = digitalRead(col3); // read column 3
    int r1c4 = digitalRead(col4); // read column 4
    digitalWrite(row1, HIGH);     // reset row 1

    digitalWrite(row2, LOW);      // send a signal through row 2
    int r2c1 = digitalRead(col1); // read column 1
    int r2c2 = digitalRead(col2); // read column 2
    int r2c3 = digitalRead(col3); // read column 3
    int r2c4 = digitalRead(col4); // read column 4
    digitalWrite(row2, HIGH);     // reset row 2

    digitalWrite(row3, LOW);      // send a signal through row 3
    int r3c1 = digitalRead(col1); // read column 1
    int r3c2 = digitalRead(col2); // read column 2
    int r3c3 = digitalRead(col3); // read column 3
    int r3c4 = digitalRead(col4); // read column 4
    digitalWrite(row3, HIGH);     // reset row 3

    digitalWrite(row4, LOW);      // send a signal through row 3
    int r4c1 = digitalRead(col1); // read column 1
    int r4c2 = digitalRead(col2); // read column 2
    int r4c3 = digitalRead(col3); // read column 3
    int r4c4 = digitalRead(col4); // read column 4
    digitalWrite(row4, HIGH);     // reset row 4

    // check if 1 was pressed
    if (r1c1 == LOW)
    {
        Serial.println('1'); // print 1 in the serial monitor
        code = String(code + 1); // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 2 was pressed
    if (r1c2 == LOW)
    {
        Serial.println('2'); // print 2 in the serial monitor
        code = String(code + 2); // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // we check if 3 was pressed
    if (r1c3 == 0)
    {
        Serial.println('3'); // print 3 in the serial monitor
        code = String(code + 3);  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 4 was pressed
    if (r2c1 == 0)
    {
        Serial.println('4'); // print 4 in the serial monitor
        code = String(code + 4); // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 5 was pressed
    if (r2c2 == 0)
    {
        Serial.println('5'); // print 5 in the serial monitor
        code = String(code + 5); // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 6 was pressed
    if (r2c3 == 0)
    {
        Serial.println('6'); // print 6 in the serial monitor
        code = String(code + 6);  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 7 was pressed
    if (r3c1 == 0)
    {
        Serial.println('7'); // print 7 in the serial monitor
        code = String(code + 7); // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 8 was pressed
    if (r3c2 == 0)
    {
        Serial.println('8'); // print 8 in the serial monitor
        code = String(code + 8);  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 9 was pressed
    if (r3c3 == 0)
    {
        Serial.println('9'); // print 9 in the serial monitor
        code = String(code + 9);  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if 0 was pressed
    if (r4c2 == 0)
    {
        Serial.println('0'); // print 0 in the serial monitor
        code = String(code + 0);  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if * was pressed
    if (r4c1 == 0)
    {
        Serial.println('*'); // print * in the serial monitor
        code = String(code + '*');  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if # was pressed
    if (r4c3 == 0)
    {
        Serial.println('#'); // print # in the serial monitor
        code = String(code + '#');  // adds to the code variable
        Serial.println(code);
        delay(1000);         // delay for one second
    }

    // check if A was pressed
    if (r1c4 == 0)
    {
        Serial.println('A'); // print A in the serial monitor
        code = String(code + 'A');  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if B was pressed
    if (r2c4 == 0)
    {
        Serial.println('B'); // print B in the serial monitor
        code = String(code + 'B');  // adds to the code variable
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if C was pressed
    if (r3c4 == 0)
    {
        Serial.println('C'); // print C in the serial monitor
        code = String(""); // resets the code variable to empty string
        Serial.println(code);
        delay(1000); // delay for one second
    }

    // check if D was pressed
    if (r4c4 == 0)
    {
        Serial.println('D'); // print D in the serial monitor
        code = String(""); // resets the code variable to empty string
        locked = true;  // changes the locked variable to true
        Serial.println(code);
        delay(1000); // delay for one second
    }


    //if the user has input 6 digits/letters
    if (code.length() == 6)
    {
      if(code == password){  // if the inputed code matches the password
        digitalWrite(greenLed, HIGH); // turns on the green led
        locked = false; // turns locked to false
      }else{
        digitalWrite(redLed,HIGH);
        delay(1000);
        count++;
        code = String("");
      }
    }

    // if the button D is pressed it turns the boolen to true
```



```
    if(locked == true){
          digitalWrite(greenLed, LOW); // the green led is turned off if
        }

    // For turning off the red led
    if(code.length() == 0 && count == 3){ // if the count is equal to zero meaning
      delay(30000);
      count =0;
      digitalWrite(redLed, LOW); // turns off the red led after 30 seconds (line 23
    }
    if(code.length() == 0 && count != 3){
      digitalWrite(redLed, LOW); // turns of the red led
    }
}

```

Besvarad.

## 7 Bits and bytes

Write a program that sets and unsets bits in a specific number.

The program should take three arguments:

- the number where the bits are to be set/unset
- the command to set or unset the bits
- the number of the bit to set/unset, where 1 means the least significant bit, 2 the second least significant, and so on.

The program should print the value of the number in both binary and hexadecimal form.

For example:
Calling **main.exe 0xFF00 set 1**
Should result in printing: **0xFF01 === 0b1111111100000001**

Your program should:
1) contain the function to set/unset the bit - 3 points
2) contain the function to print the binary number - 2 points
3) use the smallest possible dataset, e.g. for 0xFF - char, for 0xFFFF - int, etc. - 3 points
4) be commented - 3 points
5) be fail-safe - 4 points

You can use onlinegdb.com for this question. Please remember to paste the code back to the form below!

**Skriv in ditt svar här**

Obesvarad.