

Musterlösung Programmierübung

Julia Petz

24 April 2018

Aufgabe 1 - “Hello World”

Beginne mit dem bekannten „Hello World“-Programm. Gib dabei den String auf dem seriellen Monitor aus.

```
void setup(){
    Serial.begin(9600);
}

void loop(){
    Serial.println("Hello World!");
    delay(10); // Delay a little bit to improve simulation performance
}
```

Aufgabe 2 - Temperatur-Umrechner

Schreibe ein C-Programm, welches Celsius-Temperaturen (Zufallszahlen im Intervall [-15; 35]) in Grad Fahrenheit umrechnet. Verwende für die Umrechnung eine Funktion. Gib dafür sowohl die Celsius-Temperaturen als auch die Fahrenheit-Temperaturen aus.

```
//global variables
double celsius = 0;
double fahrenheit = 0;

//function for calculation
//no parameters due to the use of global variables
double celsius2fahrenheit(){
    return(((celsius*9)/5)+32);
}

void setup(){
    Serial.begin(9600);
}

void loop(){
    celsius = random(-15,36);
    fahrenheit = celsius2fahrenheit();
    //format the output
    Serial.print(celsius);
    Serial.print("\t");
    Serial.print(fahrenheit);
    Serial.print("\n");
    delay(10); // Delay a little bit to improve simulation performance
}
```

Aufgabe 3 - Minimum und Maximum

Dieses kurze Programm soll jeweils das Minimum und Maximum eines Arrays ausgeben. Sowohl Minimum als auch Maximum sollen nicht mit den bereits vorhandenen Funktionen $\min(x,y)$ oder $\max(x,y)$ berechnet werden, sondern mit einer selbst geschriebenen. Weiters soll ein Array mit zehn Zufallszahlen verwendet werden. Beachte dabei die Modularisierung von Programmcodes und entscheide selbst wie die Ausgabe im seriellen Monitor aussieht.

```
int myArray[10];

//function: finding minimum
int minimum(){
    int result = myArray[0];
    for(int i = 1; i < 10; i++){
        if(result > myArray[i]){
            result = myArray[i];
        }
    }
    return(result);
}

//function: finding maximum
int maximum(){
    int result = myArray[0];
    for(int i = 1; i < 10; i++){
        if(result < myArray[i]){
            result = myArray[i];
        }
    }
    return(result);
}

//print the array
void printArray(){
    for(int i = 0; i < 10; i++){
        Serial.print(myArray[i]);
        Serial.print("\t");
    }
    Serial.print("\n");
}

void setup(){
    Serial.begin(9600);
}

void loop(){
    //fill the array with random numbers
    for(int i = 0; i < 10; i++){
        myArray[i] = random(0, 100);
    }
    printArray();
    Serial.print("\n");
    Serial.print("Minimum:");
    Serial.print(minimum());
}
```

```

Serial.print("\n");
Serial.print("Maximum:");
Serial.print(maximum());
Serial.print("\n");
delay(1000); // Delay a little bit to improve simulation performance
}

```

Aufgabe 4 - Lotto-Tipp

Eine Funktion soll einen Lottotipp (6 aus 45) zusammenstellen. Dabei sind die 6 Lottozahlen in einem Array zu speichern. Bei „6 aus 45“ können Zahlen nicht mehrfach vorkommen. Überlege dir eine Kontrolle, dass jede Zahl nur einmal im Array vorkommt. Gib das Array aus.

```

int lottoTipp[6];
int checkArray[45];

//print the array
void printArray(){
    for(int i = 0; i < 6; i++){
        Serial.print(lottoTipp[i]);
        Serial.print("\t");
    }
    Serial.print("\n");
}

//checks if the number already exists in the lotto tip
bool tippCheck(int tipp){
    if(checkArray[tipp-1] == 1){
        return(false);
    }
    return(true);
}

//returns a random number which is not in the lotto tip
int randomNumber(){
    int rand = random(1,46);
    if(tippCheck(rand)){
        checkArray[rand-1]=1
        return(rand);
    }
    else{
        return(randomNumber());
    }
}

void setup(){
    Serial.begin(9600);
}

void loop(){
    //fill the array with unique random numbers
    for(int i = 0; i < 6; i++){
        lottoTipp[i] = randomNumber();
    }
}

```

```

}
printArray();
delay(1000); // Delay a little bit to improve simulation performance
}

```

Aufgabe 5 - Würfelergebnisse

Es sind n (maximal 1000) Zufallszahlen aus {1, 2, 3, 4, 5, 6} in ein Array zu speichern. Die Ausgabe sind die absoluten und relativen Häufigkeiten der einzelnen Würfelergebnisse.

```

int n = 10;
int diceResult[n];

//prints diceResult
void printArray(){
    for(int i = 0; i < n; i++){
        Serial.print(diceResult[i]);
        Serial.print("\t");
    }
    Serial.print("\n");
}

void setup(){
    Serial.begin(9600);
}

void loop(){
    //fill diceResult with random numbers
    for(int i = 0; i < n; i++){
        diceResult[i] = random(1, 7);
    }
    int resultAbs[6] = {0,0,0,0,0,0};
    double resultRel[6] = {0,0,0,0,0,0};
    //count the absolute values of each number
    for(int i = 0; i < n; i++){
        resultAbs[(diceResult[i])-1]=resultAbs[(diceResult[i])-1]+1;
    }
    printArray();
    //print the absolute results
    //calculate the relative results
    for(int i = 0; i < 6; i++){
        Serial.print(resultAbs[i]);
        Serial.print("\t");
        resultRel[i] = ((double)resultAbs[i])/n;
    }
    Serial.print("\n");
    //print the relative results
    for(int i = 0; i < 6; i++){
        Serial.print(resultRel[i]);
        Serial.print("\t");
    }
    Serial.print("\n");
}

```

```
    delay(1000); // Delay a little bit to improve simulation performance  
}
```