# Grundlagen Programmieren 2016 1. Nebenprüfungstermin - Gruppe A

**Erstellen Sie einen Workspace am USB-Stick oder in Laufwerk C:.  
Erzeugen Sie ein Projekt und darin ein Package mit Ihrem Namen.   
Speichern Sie alle Programme in dieses Package.   
Bennen Sie die Programme genauso, wie es in der Angabe steht.  
Exportieren Sie dann nur das Package („Create only selected directories“)   
in ein ZIP-File, das ebenfalls Ihren Namen trägt, auf den USB-Stick.**

## A1) SucheString (14 Punkte)

Erstellen Sie ein Unterprogramm „sucheString“, an das beliebig viele Strings übergeben werden. Geben Sie jenen String zurück, der aus den meisten Vokalen (A, E, I, O, U, a, e, i, o, u) besteht, aber keine Ziffern beinhaltet. Kommen mehrere Strings in Frage, können Sie einen beliebigen String zurückliefern. Die Groß- und Kleinschreibung soll dabei ignoriert werden.  
  
Beispiel:  
Übergabe: “Hallo123“, “Das ist ein Test“, “JUHUUUUU“, “88=Achtundachtzig“  
Rückgabe: “JUHUUUUU“  
Begründung: “Hallo123“ 🡪 enthält auch aus Ziffern  
 “Das ist ein Test“ 🡪 5 Vokale  
 “JUHUUUUU“ 🡪 6 Vokale  
 “88=Achtundachtzig“ 🡪 enthält auch auf Ziffern

## A2) EineAndereReihe (14 Punkte)

Berechnen Sie das 500. Glied folgender Reihe und geben Sie es aus:

1/2 + 2/3 - 3/4 - 4/5 - 5/6 + 6/7 + 7/8 - 8/9 - 9/10 - 10/11 + 11/12 + 12/13 - 13/14 - ….

## A3) BruteForce (14 Punkte)

Erstellen Sie ein zufälliges Wort, das aus 5 Zeichen besteht. Die ersten drei Zeichen sind Großbuchstaben gefolgt von 2 Ziffern. (Format: BBBZZ, Beispiel: „GNK15“ oder „ABC12“ oder „KUL01“).   
Versuchen Sie nun mittels der Brute-Force-Methode das zuvor erstellte Wort herauszufinden. Geben Sie die Anzahl der Versuche aus, die dazu benötigt werden.

## A4) Matrix (14 Punkte)

Erstellen Sie ein Unterprogramm „matrix“, an das 4 Zahlen übergeben werden.

Die erste Zahl ist die Anzahl der Zeilen, die zweite Zahl ist die Anzahl der Spalten einer Matrix.   
Die dritte Zahl stellt die Untergrenze und die vierte Zahl die Obergrenze für Zufallszahlen dar.   
Befüllen Sie nun die Matrix mit ganzzahligen Zufallszahlen, welche zwischen Unter- und Obergrenze liegen (die Grenzen dürfen nicht dabei sein!).  
Erstellen Sie nun ein neues Array, in dem die Summen der Werte der Zeilen und ein weiteres Array, in dem die Summen der Werte der Spalten abgespeichert werden.  
Nun suchen Sie jeweils die größte Zahl aus den beiden soeben erzeugten Arrays.   
Abschließend multiplizieren Sie die beiden Maxima (die Sie gerade aus den Arrays herausgesucht haben) und geben das Produkt als Ergebnis zurück.

Im Unterprogramm selbst müssen keine Ausgaben erfolgen.  
  
Beispiel:

Aufruf: matrix(3, 2, 7, 16)  
Ergebnis: 1023

Erklärung (muss nicht ausgegeben werden!):  
Befüllte Matrix: 8 8  
 14 15  
 9 10  
  
MatrixZ: 16 29 19 (8+8=16, 14+15=29, 9+10=19)

MatrixS: 31 33 (8+14+9=31, 8+15+10=33)

MaximumZ: 29 (Größte Zahl aus 16, 29, 19 ist 29)

MaximumS: 33 (Größte Zahl aus 31, 33 ist 33)

Rückgabe: 1023 (33 \* 31 = 1023)

## A5) ZehnKleineIrgendwas (14 Punkte)

Sie haben ein rundes Spielfeld, welches aus 10 Feldern besteht. (Nach dem 10. Feld kommt wieder das 1. Feld). Sie starten auf dem ersten Feld und würfeln mit einem normalen Würfel (Augenanzahl 1 bis 6). Sie ziehen die gewürfelte Augenzahl weiter (und stehen somit auf einem neuen Feld). Wenn Sie 1.000 Mal würfeln, welches Feld wurde am wenigsten besucht? Sollten mehrere Felder gleich wenig besucht werden, geben Sie nur eines davon aus.  
Anmerkung: Das Startfeld (Feld 1) zählt bereits mit.  
  
Die Ausgabe soll wie folgt aussehen:

Sie stehen auf Feld 1 und würfeln 3   
Sie stehen auf Feld 4 und würfeln 5   
Sie stehen auf Feld 9 und würfeln 2   
Sie stehen auf Feld 1 und würfeln 6   
Sie stehen auf Feld 7 und würfeln 1  
…  
Sie stehen auf Feld 4 und würfeln nicht mehr.

Feld 7 wurde nur 87 mal besucht