Aufgabe 1

Der gegebene Code-Abschnitt berechnet in einer while-Schleife die Fakultät einer positiven natürlichen Zahl.

```
<html><body><pre><script type="text/javascript">

var n = 6, i = 1, fak = 1;
while (i <= n) {
   fak *= i;
   i++;
}
document.write(n + "! = " + fak);

</script></pre></body></html>
```

a. Verändern Sie n.
b. Formulieren Sie das Programmstück unter Verwendung der for - Schleife um.
c. Verändern Sie das Programm so, dass die Fakultät nicht aufsteigend mittels 1·2·3·4·5·6 sondern absteigend per 6·5·4·3·2·1 berechnet wird.

Aufgabe 2

Erstellen Sie ein Programm nach folgendem Algorithmus:

1. Benutzereingabe eines Zeichens (window.prompt)
2. Fehlermeldung, wenn mehr als ein Zeichen eingegeben wurde (string.length , window.alert)
3. Erkennungsmeldung, wenn das Zeichen ein *Vokal*, also Element der Menge _[aeiouAEIOU]_ ist (string.charAt(0) , window.alert).

Verwenden Sie in Schritt 3. zuerst die if .. else Anweisung und anschliessend die switch Verzweigungsanweisung.

Aufgabe 3

Nachfolgend sehen wir ein einfaches Programm zur zeichenweise Stringanalyse:

```
<html><body><pre><script type="text/javascript">

var str = "Hallo Welt", out = "\n";
for (var i=0; i<str.length; i++) {
   out += str.charAt(i) + "\n";
}
document.write(out);
</script></pre></body></html>
```

a. Geben Sie das Programm ein und lassen Sie es laufen.
b. Ändern Sie den Programmcode so, dass der String **rückwärts** ausgegeben wird.
c. Ändern Sie weiterhin den Programmcode dahingehend, dass nur Konsonanten und Leerzeichen

(also keine Vokale) ausgegeben werden.

Aufgabe 4

Wir gehen nun das Zitat

> Das Schmieren von Politikern ist turnusmäßig durchzuführen, damit eine festgefahrene Politik immer mal wieder ins Rutschen kommt…
>
> Loriot

zeichenweise von vorne nach hinten durch und zählen die Leerzeichen. Hierzu lehnen wir uns an folgenden *Pseudocode* an:

```
define new variables n = 0, i=0, str = "Das Schmieren von ...";
while (i less than length of str) {
   if (character at position i is blank)
      increment n;
   increment i;
}
output of n;
```

1. Vervollständigen Sie die rudimentäre Zeichenkette im Pseudocode auf das gesamte Zitat. Beachten Sie dabei, dass innerhalb eines Strings keine neue Zeile angefangen werden kann! Wie behelfen wir uns?
2. Verwenden Sie an den entsprechenden Stellen im Programm die Eigenschaft length und die Methode charAt eines Strings.

Aufgabe 5

Wir wollen nun eine Multiplikationstafel für unseren Neffen in der 3. Klasse programmieren. Diese sieht für $n = 4$ etwa so aus:

-	**1**	**2**	**3**	**4**
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12
4	4	8	12	16

Der dazugehörige HTML-Code lautet:

```
<table style="border:1px solid black;">
  <tr><th>-</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
  <tr><th>1</th><td>1</td><td>2</td><td>3</td><td>4</td></tr>
  <tr><th>2</th><td>2</td><td>4</td><td>6</td><td>8</td></tr>
  <tr><th>3</th><td>3</td><td>6</td><td>9</td><td>12</td></tr>
  <tr><th>4</th><td>4</td><td>8</td><td>12</td><td>16</td></tr>
</table>
```

a. Lassen Sie die Dimension *n* der Tabelle vom Benutzer *(Neffen)* eingeben *(window.prompt)*.
b. Überlegen Sie sich eine geeignete Vorgehensweise *(Algorithmus)* für ein beliebiges ganzzahlig positives *n*.
c. Erzeugen Sie mittels Javascript den HTML-Code der zugehörigen Tabelle zunächst ohne die *<th>* - Elemente.
d. Fügen Sie nun die äusseren **fetten** <th> - Elemente als Zeilen- und Spaltenkopfzellen hinzu.
e. Fangen Sie mit einer geeigneten Fehlermeldung *(window.alert)* eine unsinnige Eingabe von *n* ab.

Hinweis: *Verwenden Sie zwei ineinandergeschachtelte for - Schleifen.*