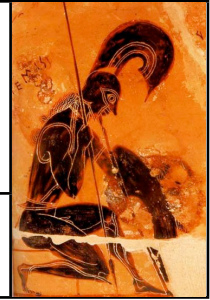


ZÄHLSCHLEIFEN IM REDCODE



Bearbeiten Sie die folgenden Aufgaben schriftlich. Absprachen mit dem Nachbarn sind gestattet und erwünscht.

1. Die *for*-Schleife

Die drei unterschiedlichen Segmente einer *for*-Schleife lauten _____,
_____, _____.

2. Dekrement und Inkrement

Sei *a* eine Variable vom Typ *Integer*. Tragen Sie in die Lücken jeweils den arithmetischen Operator, das Ergebnis und die Operation über *a* ein.

	prä-	post-
-inkrement	_____, ergibt _____, _____ <i>a</i> um 1.	_____, ergibt _____, _____ <i>a</i> um 1.
-dekrement	_____, ergibt _____, _____ <i>a</i> um 1.	a-- , ergibt a , verringert a um 1.

3. Die Adressierungsart *prädekrement-indirekt* im Redcode

a) Die Adressierungsart *prädekrement-indirekt* wird symbolisiert durch das Zeichen _____.

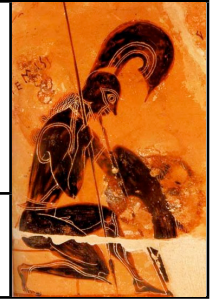
b) Was bewirken also folgende Codezeilen:

Speicheradresse	Anweisung
n	MOV #3 <1
n+1	DAT #0 #-2

c) Schreiben Sie ein *Redcode*-Programm, das hinter sich einen breiten (unendlichen) Bombenteppich streut, wobei die Adressierungsart *prädekrement-indirekt* Verwendung finden soll.



ZÄHLSCHLEIFEN IM REDCODE



Bearbeiten Sie die folgenden Aufgaben schriftlich. Absprachen mit dem Nachbarn sind gestattet und erwünscht.

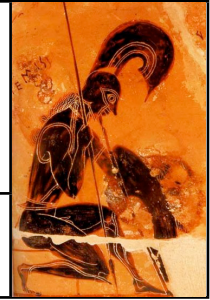
4. Die Vollendung der Schleife

- a) Welches Schleifensegment fehlt im Programm aus 3. c)? _____
- b) Die Abkürzung *DJN* steht für _____, _____.
- c) Erweitern Sie Ihr Programm in der Weise, dass das Programm direkt hinter sich einen Bombenteppich von fünf Bomben legt, und anschließend die Schleife verlässt.

5. Ein Ausblick

Sollte noch Zeit (und Lust) bestehen, versuchen Sie, das Programm in der Form zu erweitern, dass es nach dem Schleifendurchlauf hundert Speicherzellen vorwärts „hoppelt“ (sich kopiert) und wieder von vorn startet. Tipp: Mit dieser Modifikation nähern wir uns dem *Redcode*-Programm *Hase*. Versuchen Sie, die Kopieroutine (den dritten Teil) daraus zu verstehen und für sich nutzbar zu machen.

ZÄHLSCHLEIFEN IM REDCODE



Bearbeiten Sie die folgenden Aufgaben schriftlich. Absprachen mit dem Nachbarn sind gestattet und erwünscht.

Lösungen

1. Die drei unterschiedlichen Segmente einer *for*-Schleife lauten *Initialisierung*, *Schleifenbedingung* und *Fortschaltausdruck*.

2.

	prä-	post-
-inkrement	++a , ergibt a+1 , erhöht a um 1.	a++ , ergibt a , erhöht a um 1.
-dekrement	--a , ergibt a-1 , vermindert <i>a</i> um 1.	a-- , ergibt a , verringert a um 1.

3. a) Die Adressierungsart *prädekrement-indirekt* wird symbolisiert durch das Zeichen \leftarrow .

b) Der Wert #3 wird in die Speicherzelle *n-2* geschrieben. Danach zerstört sich das Programm selbst, da zu der *DAT*-Anweisung in der folgenden Speicherzelle gesprungen wird.

c)

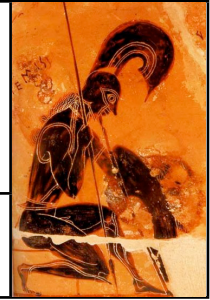
Speicheradresse	Anweisung
<i>n</i>	<i>MOV 2 <2</i>
<i>n+1</i>	<i>JMP -1</i>
<i>n+2</i>	<i>DAT #0 #-2</i>

4. a) Die **Schleifenbedingung**

b) *Decrement, Jump if Not zero*

c)

ZÄHLSCHLEIFEN IM REDCODE



Bearbeiten Sie die folgenden Aufgaben schriftlich. Absprachen mit dem Nachbarn sind gestattet und erwünscht.

Speicheradresse	Anweisung
n	MOV 3 <3
n+1	DJN -1 #5
n+2	JMP 2
n+3	DAT #0 #-3

5.

Anweisung	Kommentar
MOV 3 <3	<i>;Der Bombenlegerteil des Programms. Nach fünf Durchläufen ;wird die JMP-Anweisung ausgeführt und mit Teil 2 fortgefahren.</i>
DJN -1 #5	
JMP 2	
DAT #0 #-3	
MOV #-3 -1	<i>;Wiederherstellung ;der ursprünglichen ;Werte, ;um einen exakt ;identischen Programmablauf</i>
MOV #5 -4	<i>;nach Durchlauf der ersten Kopieroutine zu gewährleisten</i>
MOV #15 7	
MOV #5 3	
MOV #116 3	<i>;Das Programm muss 100 Zellen weiter kopiert werden, die ;letzte Anweisung also 115 Zellen weiter</i>
JMP 3	<i>;Überspringe die DAT-Anweisungen</i>
DAT #0 #5	<i>;Fange beim Kopieren mit dem Befehl JMP @-3 an, vier ;Zellen weiter</i>
DAT #0 #116	<i>;Kopiere diesen Befehl 115 Zellen nach vorne</i>
MOV <-2 <-1	<i>;Der Kopierbefehl, verbunden mit praedekrement</i>
DJN -1 #15	<i>;15 Schleifendurchläufe für dreizehn Anweisungen</i>
JMP @-3	