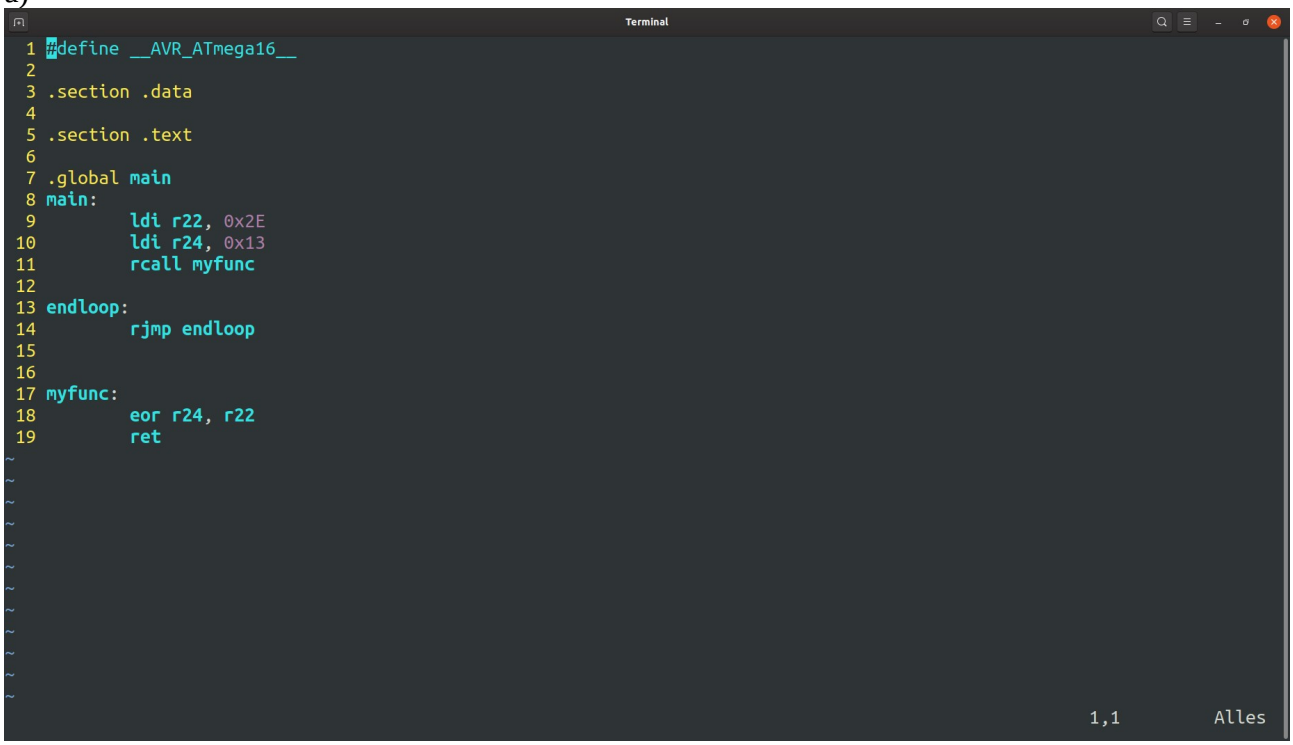
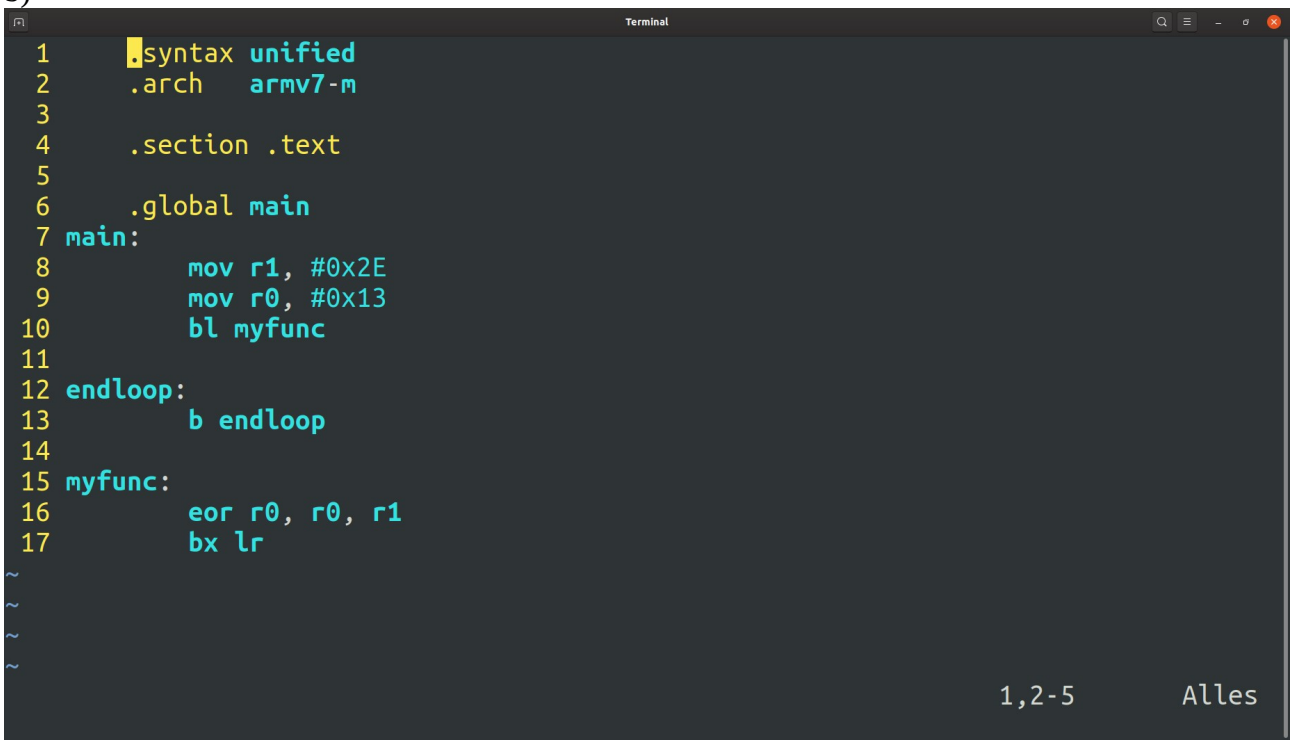
Georgios Markou MPT Test1

a)



b)



c)

LOAD-/STORE-Architekturen: d, e, a, c, b

Architekturen ohne LOAD-/STORE: d, e, c

d)
S-Flag: S = N XOR V. Wobei N=Negative und V=Overflow. Das S-Flag beinhaltet immer das richtige Vorzeichen auch nach Pufferüberläufen. Wenn es gesetzt ist, bedeutet es, dass das Ergebnis negativ ist, wenn es nicht gesetzt ist, ist das Ergebnis >= 0, dabei spielt es keine Rolle ob die Zahl als signed oder unsigned betrachtet wird.

N-Flag (Negative or less than): wird nur gesetzt, wenn das höchstwertige Bit (Bsp: 10001000) gesetzt ist. Der Benutzer muss am Ende entscheiden ob er die Zahl als signed (indem Fall, dass N=1, wäre die Zahl immer negativ) oder unsigned (positiv) interpretieren will.

e) Bei Verwendung des Befehls „b myfunc“:
Die Subroutine wird aufgerufen, doch sobald sie beendet ist und return („bx lr“) aufgerufen wird, wird das Programm beendet, weil lr=-1 (signed) und -1 ungültig ist.

Bei Verwendung des Befehls „b**l** myfunc“:
Die Subroutine wird aufgerufen und in lr wird ein gültiger Wert gespeichert. Somit funktioniert das Programm richtig.

f) apsr : 0x700001f3 ; 0b01110000000000000000000111110011
N=0 Z=1 C=1 V=1
r0 : 0x0
r1 : 0x80000000

Das N Flag ist nicht gesetzt, das bedeutet, dass das Ergebnis(r0) nicht negativ, sondern >=0, egal ob man es als signed oder unsigned interpretiert.
Das Z Flag ist gesetzt, d.h. dass r0 genau 0 ist.
Das C Flag ist ebenfalls gesetzt, das bedeutet dass der Zahlenbereich von r0 zu klein ist um das Ergebnis der Addition richtig/vollständig zu speichern.
Das V Flag ist ebenfalls gesetzt, das bedeutet, dass es einen Überlauf gab und das Ergebnis in r0 nicht richtig ist.