5. (a) Implémenter en assembleur x86-64 une fonction division(a, b) capable de calculer, en effectuant des soustractions, la division entière du nombre a par le nombre b. Les nombres a et b sont non signés et représentés sur 32 bits. On peut supposer que b est non nul.

EN PSEUDO-CODE

Division(a,b)

%% Ra : contient le nombre a sur 32 bits

%% Rb : contient le nombre b sur 32 bits

Division : Rinc <- 1

Rnb <- 0 %% contient le nombre de fois qu’on peut soustraire b à a

Rreste <- Ra %% contient le résultat des soustractions progressives de a et b

Loop : JMP end if Rreste < Rb

Rreste <- Rreste – Rb

Rnb <- Rnb + Rinc

JMP Loop

End:

EN ASSEMBLEUR

.intel\_syntax noprefix

.text

.global division

.type division, at function

Division: %% RDI : le nombre a sur 32 bits (EDI)

%% RSI : le nombre b sur 32 bits (ESI)

PUSH RBP

MOV RBP, RSP

MOV R8, 0

MOV R8D, EDI %% reste progressif des soustractions de a et b

MOV RAX, 0 %% nombre de fois qu’on peut soustraire b à a

Loop : CMP R8D, ESI

JL end

SUB R8D, ESI

INC RAX

JMP Loop

End: POP RBP %% RAX est l’argument de retour, il contient le

RET %% résultat de la division entière de a par b