4. Un palindrome est un mot qui se lit de façon identique de gauche à droite et de droite à gauche. Par exemple, le mot “RADAR” est un palindrome. Implémenter en assembleur x86-64 une fonction est\_palindrome(s) qui retourne 1 si la chaîne de caractères pointée par s est un palindrome, et 0 sinon. La chaîne s est encodée en Iso latin1, et son dernier caractère est suivi par un octet nul.

EN PSEUDO-CODE

Est\_palindrome

Palindrome : %% Rad : adresse de la chaîne de caractère

Rinc <- 1

Rsize <- 0 %% contiendra la taille (sans le zéro final)

Rid <- 0 %% indice qui parcourt la chaîne en partant du début

Rz <- 0

Rpal <- 1 %% 1 si c’est un palindrome, 0 sinon

Taille : Rtab <- Rad + Rsize

Rval <- ptr Rtab

JMP next if Rval == Rz

Rsize <- Rsize + Rinc

Next : Rfin <- Rsize – Rinc %% indice qui parcourt la chaîne en partant de la fin

Loop : Rtab <- Rad + Rid

Rval <- ptr Rtab

Rtab’ <- Rad + Rfin

Rval’ <- ptr Rtab’

JMP end if Rtab >= Rtab’

JMP is\_not\_pal if Rval != Rval’

Rid <- Rid + Rinc

Rfin <- Rfin – Rinc

JMP Loop

Is\_not\_pal: Rpal <- 0

End:

EN ASSEMBLEUR

.intel\_syntax noprefix

.text

.global palindrome

.type palindrome, at function

Palindrome: %% RDI: adresse de la chaîne de caractère

PUSH RBP

MOV RBP, RSP

MOV RAX, 1 %% 1 si c’est un palindrome, 0 sinon

MOV R8, 0 %% taille de la chaîne sans le zéro final

MOV R10, 0 %% indice pour parcourir la chaîne depuis le début

Taille : CMP byte ptr [RDI + R8 ], 0

JE next

INC R8

JMP Taille

Next : DEC R8

Loop : CMP R10, R8

JGE end

MOV R9B, byte ptr [RDI + R10]

CMP R9B, byte ptr [ RDI + R8]

JNE not\_a\_pal

INC R10

DEC R8

JMP Loop

Not\_a\_pal: MOV RAX, 0

End: POP RBP

RET