```
# EXERCICIO 4 - IMPLEMENTAÇÃO STRCAT

                 .data
                 .align 0
strcat_dig:      .asciiz "strcat(string dest, string source)\n>input =
string dest, string source\n> output = dest + source\n\n"
dest_dig:        .asciiz "Digite uma string para ser o dest do strcat: "
source_dig:      .asciiz "Digite uma string para ser o srce do strcat: "
result_txt:      .asciiz "O resultado foi: "
                 .align 2
dest_size:       .word 35
source_size:     .word 35
result_size:     .word 70
                 .text
                 .globl main
main: # 0. Imprimindo as boas vindas :) --------------------------------
      la   $a0,  strcat_dig       # load strcat_dig
      addi $v0,  $zero,     4     # print code
      syscall                     # system, do it!

      # 1. Lendo a str1 ------------------------------------------------
      la   $a0,  dest_dig         # load dest_dig
      addi $v0,  $zero,     4     # print code
      syscall                     # system, do it!
      lw   $a0,  dest_size        # sizeof memory to alloc
      addi $v0,  $zero,     9     # alloc memory code
      syscall                     # system, do it!
      move $s0,  $v0              # $s0 = &str1
      move $a0,  $s0              # $a0 = &str1
      lw   $a1,  dest_size        # $a1 = srt1size
      li   $v0,  8                # read str code
      syscall                     # system, do it!


     # 2. Lendo a str2 -------------------------------------------------
      la   $a0,  source_dig       # load source_dig
      addi $v0,  $zero,     4     # print code
      syscall                     # system, do it!
      lw   $a0,  source_size# sizeof memory to alloc
      addi $v0,  $zero,     9     # alloc memory code
      syscall                     # system, do it!
      move $s1,  $v0              # $s1 = &str2
      move $a0,  $s1              # $a0 = &str2
      lw   $a1,  source_size      # $a1 = srt2size
      li   $v0,  8                # read str code
      syscall                     # system, do it!

      # 3. Realizando a concatenacao das duas strings entradas ---------
      jal strcat                  # run strcat
      add  $s2,  $zero,     $v0   # save str_result
      # 4. Imprimindo o resultado na tela e encerrando o programa ------
      la   $a0,  result_txt       # load result_txt
      addi $v0,  $zero,     4     # print str code
      syscall                     # system, do it!

      move $a0,  $s2              # load  str_result
      addi $v0,  $zero,     4     # print str code
      syscall                     # system, do it!
      addi  $v0,  $zero,      10    # exit code
      syscall                       # system, do it!

#::::::::::::::::::: STRCAT ::::::::::::::::::::::
strcat:
      # 1. Montando a pilha
      addi  $sp,  $sp,  -12         # adjust stack
      sw    $s0,  8($sp)            # save &str1
      sw    $s1,  4($sp)            # save &str2
      sw    $ra,  0($sp)            # save return address

      # 2. Alocando a string resultado
      lw    $a0,  result_size       # sizeof memory to alloc
      addi  $v0,  $zero, 9          # alloc memory code
      syscall                       # system, do it!
      move  $a2,  $v0               # $a2 = &str_result


      # 2. Encontrando o final da str1
      move  $a0,  $s0               # $a0 = &str1
      move  $a1,  $a2               # $a1 = &str_result

find_end:
      lb    $t0,  0($a0)            # $t0 = str1[i]
      sb    $t0,  0($a1)            # str_result[i] = str1[i]
      addi  $a0,  $a0,  1           # incrementa endereco str1
      addi  $a1,  $a1,  1           # incrementa endereco str_result
      bne   $t0,  $zero, find_end   # loop while str1[i] != ‘\0’

      addi  $a1,  $a1,  -2          # ajusting end of the string

      # 3. Realizando a concatenação
      move  $a0,  $s1               # $a0 = &str2
loop:
      lb    $t0,  0($a0)            # $t0 = str2[i]
      sb    $t0,  0($a1)            # str_result[j] = str2[i]
      addi  $a0,  $a0,  1           # incrementa endereco str2
      addi  $a1,  $a1,  1           # incrementa endereco str_result
      bne   $t0,  $zero, loop       # loop while str2[i] != ‘\0’

      addi  $a1,  $a1,  -2          # ajusting end of the string


      # 4. Adicionando um ‘\0’ na string resultado
      addi $a0, $a0, -2             # ajusting end of the result string
      sb $zero, 0($a0)              # resultstring[final] = ‘\0’

      move  $v0, $a1                # $v0 = &resultstring

      # 5. Encerrando a funcao
      lw    $s0,  8($sp)            # load str1
      lw    $s1,  4($sp)            # load str2
      lw    $ra,  0($sp)            # load return address
      addi  $sp,  $sp,  12          # readjust stack

      jr $ra                        # return to function call
```