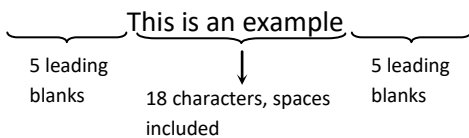


HOMEWORK #1 - SOLUTION

PROBLEM: Write a program that reads a character expression then inverts it and prints the inverted expression a number of times equal to the length of the expression. Leading and trailing blanks don't count and must not be included in the message to be displayed.

Example:

Assume that the following text is entered (input to program):



Then the output must be as follows:

Output showing 18 lines of the inverted string "elpmaxe na si sihT". The output is shown as 18 lines of text, with a bracket indicating that there are 18 lines without leading blanks.

Useful functions:

ADJUSTL (*string*): Adjusts a character string to the left by removing all leading blanks and inserting an equal number of trailing blanks.

ADJUSTR (*string*): Adjusts a character string to the right by removing all trailing blanks and inserting an equal number of leading blanks.

TRIM (*string*): Removes trailing blanks from a character string.

LEN (*string*): Returns the length of a string.

LEN_TRIM (*string*): Returns the length of a character string, without counting trailing blanks.

SOLUTION:

```
program homework1
character(len=100):: sentence
integer :: n1, n2, i
print *, "Enter a sentence"
read "(a100)", sentence
n2 = len_trim(sentence)
n1 = n2 - len_trim(adjustl(sentence)) + 1
do i = 1, n2-n1+1
print *, (sentence(i:i), i = n2, n1, -1)
end do
end program homework1
```