

CS 1014: NUMERICAL COMPUTATIONAL TECHNIQUES
PROJECT 5
Due: April 22, 1999
STRING PROCESSING

Fortran is a language largely used by scientists and engineers mainly for numerical data analysis as they require to deal with modeling, simulation, and design of physical systems. Undoubtedly, Fortran is one of the most powerful languages for such computation-intensive, math-oriented processing of numerical data. Fortran also supports processing of character data. In this project, you will have the opportunity to learn about and use character data processing features of the language.

All work for this project has to be done by you alone. In case of difficulty, you may seek help from the course instructor or from the course graduate teaching assistant.

Objectives:

In this project you will have the opportunity to learn how to:

1. Declare and initialize character type variables,
2. Read data for character type variables,
3. Process and output character type data, and
4. Declare, initialize, and use arrays (of course, it depends on your approach to the solution)

Problem Statement:

In this project, you are required to write a program to determine the number of lines, the number of words, the number of occurrences of each uppercase letter, the number of occurrences of each lowercase letter, the number of vowels, and the number of consonants in a text file. You may wonder what is the use of such program. Your program can be used to answer questions like:

Which letter is found most/least in English prose?

Which vowel/consonant is found most/least in English?

What is the percentage of occurrences of letter A in a typical English prose? And so on.

Also, a similar approach can be used to detect plagiarism in documents.

Input:

The input file (named as "text.dat") for this program contains only textual data and hence does not follow any structured organization usually observed and required in numerical data processing. In this case, each line of data in the input file contains words separated by one or more spaces, but does not exceed 80 characters in length. Also some lines in the input file may contain no character or word at all. Use the following text to create your input file for testing of your program:

Computers are today used to solve an almost unimaginable range of problems, and yet their basic structures has hardly changed in 40 years. They have become faster and more powerful, as well as smaller and cheaper, but the key to this change in the role that they play is due almost entirely to the developments in the programming languages which control their every action.

Fortran 90 is the latest version of the world's oldest high-level programming language, and is designed to provide better facilities for the solution of scientific and technological problems.

Type and create your input file with the PFE editor exactly as shown above. Do not move a word from one line to another

Output:

The program should direct all output to the file named as "report5.out". With the given sample input, the output file should look as follows:

Lines:	12	
a: 38	A: 0	
b: 7	B: 0	
c: 14	C: 1	
d: 16	D: 0	
e: 59	E: 0	
f: 8	F: 1	
g: 15	G: 0	
h: 24	H: 0	
i: 29	I: 0	
j: 0	J: 0	
k: 1	K: 0	
l: 28	L: 0	
m: 14	M: 0	
n: 28	N: 0	
o: 34	O: 0	
p: 10	P: 0	
q: 0	Q: 0	
r: 31	R: 0	
s: 30	S: 0	
t: 40	T: 1	
u: 11	U: 0	
v: 7	V: 0	
w: 4	W: 0	
x: 0	X: 0	
y: 10	Y: 0	
z: 0	Z: 0	
Vowels:	171	
Consonants:		290
Words:	92	

You must be very careful about the output format since the automated grader is very sensitive to order of the lines as well as the order of the words in a line of the output file. Also you must be careful about spelling. Note that a colon (:) **immediately follows a letter or a word in the output.**

Processing:

The most difficult part of this project is to compute the number of words in the input file. A word is to be recognized as a sequence of non-blank characters. Notice that a word in the input text file is separated from another one at least by one space. The first word in a line may or may not begin with a leading space or spaces. Similarly, the last word may or many not end with trailing a space(s). The grader server will be set in a way that **if you cannot compute the number of words correctly, still you will be able to get at most 80% of the points on the project.**

Documentation:

Do not forget to document your program. Some of your programs will be randomly selected from the archive (maintained on the grader server) later on for manual grading. Points will be deducted if your program is not well documented, well structured, and readable. You can use the program handed out for the first project as a model for documentation. Do not forget to include your name(s) in the program documentation.

Submission:

You should submit your source code electronically to the automated grader. Before you submit your program (the file with extension .f90), make sure your program runs perfectly and generates right results with some test input data. Your program will not receive any input from the keyboard or will not display any results on the screen. Do not submit the input or the output file. You will be allowed a maximum of four submits for this project. Remember the late penalty is 20% for each day late.