RÉPUBLIQUE FRANÇAISE
*Liberté Égalité Fraternité*

DOUANES & DROITS INDIRECTS

AI3O22V1

**CONCOURS INTERNE ET EXTERNE**

**POUR LE RECRUTEMENT D'INSPECTEURS DES DOUANES ET DROITS INDIRECTS**

**DANS LA SPÉCIALITÉ TRAITEMENT AUTOMATISE DE L'INFORMATION – PROGRAMMEUR DE SYSTÈME D'EXPLOITATION**

**SESSION 2022**

# ÉPREUVE ÉCRITE D'ADMISSION N°3 (OBLIGATOIRE)

(DURÉE : 2 HEURES – COEFFICIENT 1)

## TRADUCTION ET RÉPONSE A DES QUESTIONS PORTANT SUR UN TEXTE EN ANGLAIS ISSU D'UNE REVUE OU D'UNE DOCUMENTATION INFORMATIQUE

***N.B : Seules les copies des candidats admissibles seront corrigées.***

### *AVERTISSEMENTS IMPORTANTS*

Veillez à bien indiquer sur votre copie le nombre d'intercalaires utilisés (la copie double n'est pas décomptée).

L'usage de tout matériel autre que le matériel usuel d'écriture et de tout document autre que le support fourni est **interdit**.

L'usage du dictionnaire est interdit.

La copie ne saurait comporter de **nom, initiales, paraphe, signature ou tout autre signe distinctif**, susceptibles de permettre l'identification du candidat. Le non-respect de cette consigne entraînera l'exclusion du concours.

**Toute fraude ou tentative de fraude** constatée par la commission de surveillance entraînera **l'exclusion du concours.**

Il vous est interdit de quitter définitivement la salle d'examen **avant le terme d'épreuve**.

Le présent document comporte **3 pages** numérotées.

*Tournez la page, SVP*

**Après avoir procédé à la traduction des deux extraits surlignés ci-dessous :**

**1er passage :** de « The language syntax... » à « ...for new functionality. »

**2ème passage :** de «The design of the language... » à « ...the preferred usage »

**Vous répondrez ensuite <u>en anglais</u> aux deux questions suivantes**
**(il est demandé de rédiger au moins quinze lignes pour chaque réponse).**

**Question 1 :** What is the purpose and main features of the R language ?

**Question 2 :** Which are the interactions between R and C ?

## *R Language Definition*

R is a system for statistical computation and graphics. It provides, among other things, a programming language, high level graphics, interfaces to other languages and debugging facilities. This manual details and defines the R language.
The R language is a dialect of S which was designed in the 1980s and has been in widespread use in the statistical community since. Its principal designer, John M. Chambers, was awarded the 1998 ACM Software Systems Award for S.
The language syntax has a superficial similarity with C, but the semantics are of the FPL (functional programming language) variety with stronger affinities with Lisp and APL. In particular, it allows “computing on the language”, which in turn makes it possible to write functions that take expressions as input, something that is often useful for statistical modeling and graphics.
It is possible to get quite far using R interactively, executing simple expressions from the command line. Some users may never need to go beyond that level, others will want to write their own functions either in an ad hoc fashion to systematize repetitive work or with the perspective of writing add-on packages for new functionality.
The purpose of this manual is to document the language *per se*. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. Major subsystems for specific tasks, such as graphics, are only briefly described in this manual and will be documented separately.
Although much of the text will equally apply to S, there are also some substantial differences, and in order not to confuse the issue we shall concentrate on describing R.
The design of the language contains a number of fine points and common pitfalls which may surprise the user. Most of these are due to consistency considerations at a deeper level, as we shall explain. There are also a number of useful shortcuts and idioms, which allow the user to express quite complicated operations succinctly. Many of these become natural once one is familiar with the underlying concepts. In some cases, there are multiple ways of performing a task, but some of the techniques will rely on the language implementation, and others work at a higher level of abstraction. In such cases we shall indicate the preferred usage.
Some familiarity with R is assumed. This is not an introduction to R but rather a programmers’ reference manual. (...)

In every computer language variables provide a means of accessing the data stored in memory. R does not provide direct access to the computer's memory but rather provides a number of specialized data structures we will refer to as objects. These objects are referred to through symbols or variables. In R, however, the symbols are themselves objects and can be manipulated in the same way as any other object. This is different from many other languages and has wide ranging effects.
In this chapter we provide preliminary descriptions of the various data structures provided in R. More detailed discussions of many of them will be found in the subsequent chapters. The R specific function *typeof* returns the *type* of an R object. Note that in the C code underlying R, all objects are pointers to a structure with typedef SEXPREC; the different R data types are represented in C by SEXPTYPE, which determines how the information in the various parts of the structure is used.
(…)
Function mode gives information about the *mode* of an object in the sense of Becker, Chambers & Wilks (1988), and is more compatible with other implementations of the S language. Finally, the function storage.mode returns the *storage mode* of its argument in the sense of Becker et al. (1988). It is generally used when calling functions written in another language, such as C or FORTRAN, to ensure that R objects have the data type expected by the routine being called. (In the S language, vectors with integer or real values are both of mode "numeric", so their storage modes need to be distinguished.)
(…)
R objects are often coerced to different types during computations. There are also many functions available to perform explicit coercion. When programming in the R language the type of an object generally doesn't affect the computations, however, when dealing with foreign languages or the operating system it is often necessary to ensure that an object is of the correct type.

*https://cran.r-project.org/manuals.html*
*The R language definition*