```
   Checking for foreign routines
   FRICAS="/usr/local/lib/fricas/target/x86_64-linux-gnu"
   spad-lib="/usr/local/lib/fricas/target/x86_64-linux-gnu//lib/libspad.so"
   foreign routines found
   openServer result -2
                          FriCAS Computer Algebra System
              Version: FriCAS 2024-04-15 built with sbcl 2.2.9.debian
                     Timestamp: Di 28 Mai 2024 21:49:04 CEST
   ------------------------------------------------------------------------------
      Issue )copyright to view copyright notices.
      Issue )summary for a summary of useful system commands.
      Issue )quit to leave FriCAS and return to shell.
   ------------------------------------------------------------------------------

      Function declaration sixel : TexFormat -> Void has been added to
         workspace.
   Value = #<INTERPRETED-FUNCTION NIL {10020A026B}>
```

(6) -> $X + 2$

```
      There are no library operations named math
         Use HyperDoc Browse or issue
                                  )what op math
         to learn if there is any operation containing " math " in its
         name.

      Cannot find a definition or applicable library operation named math
         with argument type(s)
                                       String

         Perhaps you should use "@" to indicate the required return type,
         or "$" to specify which version of the function you need.
```

(6) -> x

x

Type: Variable(x)

(8) -> math(s) == interpretString(s)$TemplateUtilities

```
      Compiled code for math has been cleared.
      1 old definition(s) deleted for function or rule math
```

Type: Void

(8) -> $X1''$

```
      There are no exposed library operations named interpretString but
         there is one unexposed operation with that name. Use HyperDoc
         Browse or issue
                           )display op interpretString
         to learn more about the available operation.
      Cannot find a definition or applicable library operation named
         interpretString with argument type(s)
                                       String

         Perhaps you should use "@" to indicate the required return type,
         or "$" to specify which version of the function you need.
      FriCAS will attempt to step through and interpret the code.
      There are no exposed library operations named interpretString but
```

```
      there is one unexposed operation with that name. Use HyperDoc
      Browse or issue
                          )display op interpretString
      to learn more about the available operation.

   Cannot find a definition or applicable library operation named
      interpretString with argument type(s)
                                  String

      Perhaps you should use "@" to indicate the required return type,
      or "$" to specify which version of the function you need.
```

```
(8) -> X + 2
```

```
   There are no exposed library operations named interpretString but
      there is one unexposed operation with that name. Use HyperDoc
      Browse or issue
                          )display op interpretString
      to learn more about the available operation.
   Cannot find a definition or applicable library operation named
      interpretString with argument type(s)
                                  String

      Perhaps you should use "@" to indicate the required return type,
      or "$" to specify which version of the function you need.
   FriCAS will attempt to step through and interpret the code.
   There are no exposed library operations named interpretString but
      there is one unexposed operation with that name. Use HyperDoc
      Browse or issue
                          )display op interpretString
      to learn more about the available operation.

   Cannot find a definition or applicable library operation named
      interpretString with argument type(s)
                                  String

      Perhaps you should use "@" to indicate the required return type,
      or "$" to specify which version of the function you need.
```

```
(8) -> )d op interpretString

  There is one unexposed function called interpretString :
     [1] String -> Any from TemplateUtilities
```

```
(9) -> x + 2
```

```
   Compiling function math with type String -> Any
```

```
(9) -> x - 1
```

$$x - 1$$

```
                                                  Type: Polynomial(Integer)
```

```
(11) -> r := x + 2
```

$$x + 2$$

Type: Polynomial(Integer)

(12) -> $2r+y$

$$y+2x+4$$

Type: Polynomial(Integer)

(14) -> $S:=\sum_{i=1}^{10} i$

$$55$$

Type: Fraction(Polynomial(Integer))

(19) -> $\int_0^1 \sin(x)\,dx$

```
integrate(sin(x)*d x=0..1)
```

Type: Void

(15) -> `math(s) == output s`

```
Compiled code for math has been cleared.
1 old definition(s) deleted for function or rule math
```

Type: Void

(16) -> x

```
Compiling function math with type String -> Void
x
```

Type: Void

(18) -> $x+'2$

```
x+2
```

Type: Void

(20) -> $f:=\frac{1}{(x-1)^2}$

```
f := (1/((x-1)^2))
```

Type: Void

(21) -> `)d op output`

```
There are 3 exposed functions called output :
   [1] String -> Void from OutputPackage
   [2] OutputForm -> Void from OutputPackage
   [3] (String,OutputForm) -> Void from OutputPackage
```

(21) -> `)q`

Busy...

(21) ->