

```

> restart :
> motVide := proc(x :: string)
  local ch, i;
  ch := "";
  for i from 1 to length(x) do
    ch := cat(ch, "-") :
  od:
  return ch :
end:
> print(motVide("maple")) :
                                     "-----"
(1)

> estDedans := proc(ch :: string, let :: string)
  local i;
  for i from 1 to length(ch) do
    if ch[i] = let then
      return true :
    fi:
  od:
  return false :
end:
print(estDedans("maple", "a")) :
print(estDedans("maple", "b")) :
                                     true
                                     false
(2)

> replace := proc(ch1 :: string, ch2 :: name, let :: string)
  local i;
  for i from 1 to length(ch1) do
    if ch1[i] = let then
      ch2 := cat(substring(eval(ch2), 1..i - 1), let, substring(eval(ch2), i + 1
      ..length(eval(ch2)))) :
    fi:
  od:
end:
> st := "-c-pl--" :
  replace("acaplua", 'st', "a") : print(st) :
                                     "acapl-a"
(3)

> pendu := proc(ch :: string)
  local chaine, i, let,
  chaine := motVide(ch) :
  for i from 1 to 10 do
    let := readstat("une lettre") :
    if estDedans(ch, let) then
      replace(ch, 'chaine', let) :
      print(chaine);
    else
      print(cat(let, " n'est pas dans le mot")) :
    fi:
  if ch = chaine then return "gagné!" : fi:

```

```
od:  
return "perdu!" :  
end:
```

```
> print( pendu("treet") ) :
```

```
"t---t"  
"tr--t"  
"u n'est pas dans le mot"  
"treet"  
"gagné!"
```

(4)

```
> print( pendu("treet") ) :
```

```
"a n'est pas dans le mot"  
"b n'est pas dans le mot"  
"c n'est pas dans le mot"  
"d n'est pas dans le mot"  
"--ee-"  
"f n'est pas dans le mot"  
"g n'est pas dans le mot"  
"h n'est pas dans le mot"  
"j n'est pas dans le mot"  
"k n'est pas dans le mot"  
"perdu!"
```

(5)

```
>
```