CORREO SEGURO

```
> restart;
```

El objetivo es enviar el correo electronico de alicia@unican.es a bernardo@unican.es
Utilizaremos el DES como criptosistema simetrico y el RSA para autenticar y cifrar la clave se sesion.

CIFRANDO EL MENSAJE

El mensaje en texto claro que se desea enviar es:

```
> message := " No cabe duda que la información es la mayor fuente de
  poder que ha conocido la humanidad y que la criptografía es una
  herramienta fundamental para su control. Se pretende presentar un
  desarrollo histórico de esta ciencia moderna, con el objetivo de que
  se conozcan sus ventajas e inconvenientes, sus peligros y leyendas.
  Saludos. x";
```

message := " No cabe duda que la información es la mayor fuente de poder que ha conocido la humanidad y que la criptografía es una herramienta fundamental para su control. Se pretende presentar un desarrollo histórico de esta ciencia moderna, con el objetivo de que se conozcan sus ventajas e inconvenientes, sus peligros y leyendas. Saludos. x"

Primero cifraremos el mensaje con el criptosistema DES en mode CBC(Cipher Block Chaining)

```
> read `DES.m`;
```

Necesitamos preparar el mensaje para cifrarlo con DES, lo dividimos en hex palabras de 64 bits. Cada palabra corresponde a 8 caracteres ASCII, por lo tanto añadiremos ceros hasta completarlo para que sea divisible por 8. Anotamos el número de palabras.

```
> messLength := length(message);

  messWords := ceil(messLength/8);

  messPadding := 8*messWords - messLength;
```

messLength := 332

messWords := 42

messPadding := 4

```
> messList := convert(message, bytes);

  messList := [op(messList),seq(0,i=1..messPadding)];
```

messList := [32, 78, 111, 32, 99, 97, 98, 101, 32, 100, 117, 100, 97, 32, 113, 117, 101, 32, 108, 97, 32, 105, 110, 102, 111, 114, 109, 97, 99, 105, 243, 110, 32, 101, 115, 32, 108, 97, 32, 109, 97, 121, 111, 114, 32, 102, 117, 101, 110, 116, 101, 32, 100, 101, 32, 112, 111, 100, 101, 114, 32, 113, 117, 101, 32, 104, 97, 32, 99, 111, 110, 111, 99, 105, 100, 111, 32, 108, 97, 32, 104, 117, 109, 97, 110, 105, 100, 97, 100, 32, 121, 32, 113, 117, 101, 32, 108, 97, 32, 99, 114, 105, 112, 116, 111, 103, 114, 97, 102, 237, 97, 32, 101, 115, 32, 117, 110, 97, 32, 104, 101, 114, 114, 97, 109, 105, 101, 110, 116, 97, 32, 102, 117, 110, 100, 97, 109, 101, 110, 116, 97, 108, 32, 112, 97, 114, 97, 32, 115, 117, 32, 99, 111, 110, 116, 114, 111, 108, 46, 32, 83, 101, 32, 112, 114, 101, 116, 101, 110, 100, 101, 32, 112, 114, 101, 115, 101, 110, 116, 97, 114, 32, 117, 110, 32, 100, 101, 115, 97, 114, 114, 111, 108, 108, 111, 32, 104, 105, 115, 116, 243, 114, 105, 99, 111, 32, 100, 101, 32, 101, 115, 116, 97, 32, 99, 105, 101, 110, 99, 105, 97, 32, 109, 111, 100, 101, 114, 110, 97, 44, 32, 99, 111, 110, 32, 101, 108, 32, 111, 98, 106, 101, 116, 105, 118, 111, 32, 100, 101, 32, 113, 117, 101, 32, 115, 101, 32, 99, 111, 110, 111, 122, 99, 97, 110, 32, 115, 117, 115, 32, 118, 101, 110, 116, 97, 106, 97, 115, 32, 101, 32, 105, 110, 99, 111, 110, 118, 101, 110, 105, 101, 110, 116, 101, 115, 44, 32, 115, 117, 115, 32, 112, 101, 108, 105, 103, 114, 111, 115, 32, 121, 32, 108, 101, 121, 101, 110, 100, 97, 115, 46, 32, 83, 97, 108, 117, 100, 111, 115, 46, 32, 120]

messList := [32, 78, 111, 32, 99, 97, 98, 101, 32, 100, 117, 100, 97, 32, 113, 117, 101, 32, 108, 97, 32, 105, 110, 102, 111, 114, 109, 97, 99, 105, 243, 110, 32, 101, 115, 32, 108, 97, 32, 109, 97, 121, 111, 114, 32, 102, 117, 101, 110, 116, 101, 32, 100, 101, 32, 112, 111, 100, 101, 114, 32, 113, 117, 101, 32, 104, 97, 32, 99, 111, 110, 111, 99, 105, 100, 111, 32, 108, 97, 32, 104, 117, 109, 97, 110, 105, 100, 97, 100, 32, 121, 32, 113, 117, 101, 32, 108, 97, 32, 99, 114, 105, 112, 116, 111, 103, 114, 97, 102, 237, 97, 32, 101, 115, 32, 117, 110, 97, 32, 104, 101, 114, 114, 97, 109, 105, 101, 110, 116, 97, 32, 102, 117, 110, 100, 97, 109, 101, 110, 116, 97, 108, 32, 112, 97, 114, 97, 32, 115, 117, 32, 99, 111, 110, 116, 114, 111, 108, 46, 32, 83, 101, 32, 112, 114, 101, 116, 101, 110, 100, 101, 32, 112, 114, 101, 115, 101, 110, 116, 97, 114, 32, 117, 110, 32, 100, 101, 115, 97, 114, 114, 111, 108, 108, 111, 32, 104, 105, 115, 116, 243, 114, 105, 99, 111, 32, 100, 101, 32, 101, 115, 116, 97, 32, 99, 105, 101, 110, 99, 105, 97, 32, 109, 111, 100, 101, 114, 110, 97, 44, 32, 99, 111, 110, 32, 101, 108, 32, 111, 98, 106, 101, 116, 105, 118, 111, 32, 100, 101, 32, 113, 117, 101, 32, 115, 101, 32, 99, 111, 110, 111, 122, 99, 97, 110, 32, 115, 117, 115, 32, 118, 101, 110, 116, 97, 106, 97, 115, 32, 101, 32, 105, 110, 99, 111, 110, 118, 101, 110, 105, 101, 110, 116, 101, 115, 44, 32, 115, 117, 115, 32, 112, 101, 108, 105, 103, 114, 111, 115, 32, 121, 32, 108, 101, 121, 101, 110, 100, 97, 115, 46, 32, 83, 97, 108, 117, 100, 111, 115, 46, 32, 120, 0, 0, 0, 0]

```
> intTo2Hex := intVal -> substring(convert(intVal+256,hex),2..3):

> messHexList := map(intTo2Hex,messList):

  messHexWordList :=
  [seq(cat(seq(messHexList[8*i+j],j=1..8)),i=0..messWords-1)];
```

messHexWordList := [204E6F2063616265, 2064756461207175, 65206C6120696E66, 6F726D616369F36E, 206573206C61206D, 61796F7220667565, 6E74652064652070, 6F64657220717565, 20686120636F6E6F, 6369646F206C6120, 68756D616E696461, 6420792071756520, 6C61206372697074, 6F67726166ED6120, 657320756E612068, 657272616D69656E, 74612066756E6461, 6D656E74616C2070, 6172612073752063, 6F6E74726F6C2E20, 5365207072657465, 6E64652070726573, 656E74617220756E, 206465736172726F, 6C6C6F2068697374, F37269636F206465, 2065737461206369, 656E636961206D6F, 6465726E612C2063, 6F6E20656C206F62, 6A657469766F2064, 6520717565207365, 20636F6E6F7A6361, 6E20737320766565, 6E74616A61732065, 20696E636F6E7665, 6E69656E7465732C, 207375732070656C, 6967726F73207920, 6C6579656E646173, 2E2053616C75646F, 732E207800000000]

Para cifrar el mensaje con DES en modo CBC se necesita un valor inicial. Elegimos la clave aleatoriamente. No nos preocupamos del control de paridad.

```
> randKey := substring(convert(2^64+rand(2^64)(),hex),2..17);
```

randKey := 7778EEFFF432E596

Guardamos la clave: 7778EEFFF432E596 y utilizamos la funcion de expansion del DES.

```
> randKey := "7778EEFFF432E596":

  key := keyexpander(randKey);
```

key := ["111111111010111110001110110101110011011100001101",
"111000110011111111101111001011011001011011010111",
"111110011111111011100101010111111110010011100111",
"110100011111111111111010001011101100110111001101",
"111101001110111111101111000101011110101110101111",
"111101111101111101000111111011111100011110100001",
"011010111111001111110111111011010010011110100101 1",
"101111011101010111111111111011110110100110001 1100",
"110011111101111111011010100011011000110011111110",
"011111101111101111111011010011011111111011010101",
"101111111111110101001011001110111100010011111101",
"011010110110111111011111100010111011101100000111",
"011111011111111011001110110001110011001111011010 1",
"010101111010110111111011111111011010010111100010 1",
"111111111110110010110111111010010110000111001101 1",
"100111011111111100101111101101011010110111011110"]

El valor inicial sera la palabra todos ceros cifrada con el DES y con la clave fija.

```
> allZeroes := substring(convert(2^64+0,hex),2..17);

  IV := qdDEShex(allZeroes,key);
```

allZeroes := 0000000000000000

IV := "04844CB2337C0CC2"

Ahora ciframos con DES en modo CBC. La primera palabra utiliza el valor inicial y la primera palabra del mensaje. El resto del mensaje lo ciframos con un bucle:

```
> cipherTable[1] := qdDEShex(xor64hex(IV,messHexWordList[1]),key);
  for i from 2 to messWords do
  cipherTable[i] :=
  qdDEShex(xor64hex(cipherTable[i-1],messHexWordList[i]),key):
  od:
```

$cipherTable_1$:= 86565D261132EE81

Por ajustes técnicos, lo convertimos a una lista:

```
> cipherList := convert(cipherTable,list);
```

cipherList := [86565D261132EE81, 4B0D890D95C865A9, D52AFD001FCCFFCC, 78600F93392BFA07, 5AB1CDD776D02ADD, 7BB68AA2DB6A3C1B, B6759F27A4284CD7, FD53F9D90DB177AA, 6516ECC1222224B3, 23C34DB34EF39F63, 124097634060239F, 898310A6318C7A92, AEF132759F7D5397, 28BB9900E6CC162C, "0B75D46D238C10F0", E1E2D7EB76A4AF0E, 7BF31A25D57B348F, C8B796FCBFDE1DC7, 914EEA6D28757828, 927EF9F61E6AC10E, 242A5DBEA8163FBB, 68B0AB12088CAB50, 7FE49BEDEDD5D145, 6B7653EED0B771AA, C2893218316B5F75, 1F9D1FAC611652A3, 2EF90036717AFD4E, A1AB2F7B447AC501, F16B493899CBE259, 6D01D4F19E583637, 7C841AD4082C9C43, 94C3236916FE2DD1, 9B776229BE1194E0, 67ED95F9A3C58436, A8D70F6128457B32, 3F104AEEDC282829, C8D71EACBA8EAE6D, D9049802E24F3896, 3ED65191800E8BF5, DF511D402762740F, F37E53B702950F16, 9EDB145613D71FF2]

Luego este es el mensaje cifrado

FIRMA del resumen (MAC) y cifrando la clave de sesión:

Para el propósito de la firma digital, considermos la ultima palabra del mensaje cifrado, es decir, el MAC.

```
> hash := cipherList[messWords];
```

hash := 9EDB145613D71FF2

Utilizaremos el RSA para dos tareas. Primero cifrar la clave de sesión, para que el receptor Bernardo, pueda descrifrarlo. Segundo, firmar el MAC, para autenticar el mensaje.

```
> p := nextprime(rand(10^80)());;

  q := nextprime(rand(10^80)());;
```

p := 330736974742561435635584587189767467538305380320622220857229741217686043056140 73

q := 800374092598119526553100754871637971794904570391695941600884305716749604988341 31

```
> Senderp :=
  330736974742561435635584587189767467538305380320622220857229741217686043056140
  73:

  Senderq :=
  800374092598119526553100754871637971794904570391695941600884305716749604988341
  31:

  Sendern := Senderp*Senderq;

  Sendere := 2^16+1;

  Senderd := (1/Sendere) mod ((Senderp-1)*(Senderq-1)):
```

Sendern := 26471330604822478554767314516351292234851316805313644160782027833452681945007387335502458484705940337644698956306946519364873408770884309110728695859345263255635563

Sendere := 65537

Publicamos (Sendern, Sendere) como clave publica. Para firmar el resumen, lo ciframos con nuestra clave privada:

```
> digestNum := convert(hash, decimal, hex);

  signature := Power(digestNum,Senderd) mod Sendern;

  sigHex := convert(signature, hex);
```

digestNum := 11446765237824856050

signature := 11862932621957864817868582045035660635958731894976135851319449015536066768663781650594928506692190792615343888857545036357302803731629349681995966807619012123121

sigHex := 1599DCB27215FB1B314EEEC88017D2B824C993DA079F92E38486E44EC40225A85319E1B14DEE621B6E7D8BCC9D0028796A16144E770F080EE66273CCA59BA3B2A8CD1

Para cifrar la clave de sesion, necesitamos la clave pública del receptor, (Receivern, Receivere).

```
> p := nextprime(rand(10^80)());;

  q := nextprime(rand(10^80)());;
```

p := 20457916453747019461644031395307920624947349951053530086146486307198155590763471

q := 92673709525428510973272600608981219760099374675982933766845473509473676470788363

```
> Receiverp :=
  20457916453747019461644031395307920624947349951053530086146486307198155590763471:

  Receiverq :=
  92673709525428510973272600608981219760099374675982933766845473509473676470788363:

  Receivern := Receiverp*Receiverq;

  Receivere := 2^16+1:

  Receiverd := (1/Receivere) mod ((Receiverp-1)*(Receiverq-1));
```

Receivern := 1895911006930035821113645529767884057259283818477739524604309139408303387651699653989993189557809168785274595534506647543454372607114206130735004084228432287973

Receiverd := 16934378827787423424378684105398842935112067407121155422324343256890499062568501414890382495591472377924317562036902860068902428584575182269194793894572340\83833

```
> keyNum := convert(randKey, decimal, hex);

  encryptKey := Power(keyNum, Receivere) mod Receivern;
  encryptKeyHex := convert(encryptKey, hex);
```

keyNum := 8608893470799750550

encryptKey := 9309148769655280501063089801724909873150946243346389358425110650722385931\
91263649684047171551528285742909238884912207243150277191281484815781536653568560\
503902

encryptKeyHex := 10F36CDAE57F665A367984588EC59ACACFD57D31338F7C119F57E5AFE25B13\
E06BB669508542CE257DB955EE4BD8840830EE5C728A470EEC8E267BBCB113000F29C5E

EL MENSAJE CIFRADO COMPLETO

```
>
```

Ponenmos todo junto y obtenemos el mensaje.

cipherList := [`86565D261132EE81`, `4B0D890D95C865A9`, D52AFD001FCCFFCC, `78600F93392BFA07`, `5AB1CDD776D02ADD`, `7BB68AA2DB6A3C1B`, B6759F27A4284CD7, FD53F9D90DB177AA, `6516ECC1222224B3`, `23C34DB34EF39F63`, `124097634060239F`, `898310A6318C7A92`, AEF132759F7D5397, `28BB9900E6CC162C`, "0B75D46D238C10F0", E1E2D7EB76A4AF0E, `7BF31A25D57B348F`, C8B796FCBFDE1DC7, `914EEA6D28757828`, `927EF9F61E6AC10E`, `242A5DBEA8163FBB`, `68B0AB12088CAB50`, `7FE49BEDEDD5D145`, `6B7653EED0B771AA`, C2893218316B5F75, `1F9D1FAC611652A3`, `2EF90036717AFD4E`, A1AB2F7B447AC501, F16B493899CBE259, `6D01D4F19E583637`, `7C841AD4082C9C43`, `94C3236916FE2DD1`, `9B776229BE1194E0`, `67ED95F9A3C58436`, A8D70F6128457B32, `3F104AEEDC282829`, C8D71EACBA8EAE6D, D9049802E24F3896, `3ED65191800E8BF5`, DF511D402762740F, F37E53B702950F16, `9EDB145613D71FF2`];

hash := `9EDB145613D71FF2`;

sigHex := `1599DCB27215FB1B314EEEC88017D2B824C993DA079F92E38486E44EC40225A85319E1B14DEE621B6E7D8BCC9D0028796A16144E770F080EE66273CCA59BA3B2A8CD1`;

encryptKeyHex := `10F36CDAE57F665A367984588EC59ACACFD57D31338F7C119F57E5AFE25B13E06BB669508542CE257DB955EE4BD8840830EE5C728A470EEC8E267BBCB113000F29C5E`;

Sendern := 26471330604822478554767314516351292234851316805313644160782027833452681945007387335502458484705940337644698956306946519364873408770884309110728695859345263255 63;

Sendere := 65537;

```
>
```

LEYENDO EL MENSAJE

Nos ponemos en posicion del receptor. Comenzamos una nueva sesion, para borrar todo de la memoria. El receptor conoce la clave publica del emisor y su clave privada. Obviamente, tambien, conoce el mensaje cifrado.

```
> restart;

  read `DES.m`:
```

```
> Receivern :=
  189591100693003582111364552976788405725928381847773952460430913940830338765\
  169965398999318955780916878527459553450664754345437260711420613073500408422843\
  22\
  87973;

  Receiverd :=
  169343788277874234243786841053988429351120674071211554223243432568904990625\
  685014148903824955914723779243175620369028600689024285845751822691947938945723\
  40\
  83833;

  Sendern :=
  264713306048224785547673145163512922348513168053136441607820278334526819450\
  073873355024584847059403376446989563069465193648734087708843091107286958593452\
  632\
  5563;

  Sendere := 65537;
```

Receivern := 189591100693003582111364552976788405725928381847773952460430913940830338765\
169965398999318955780916878527459553450664754345437260711420613073500408422843\
22\
87973

Receiverd := 169343788277874234243786841053988429351120674071211554223243432568904990625\
685014148903824955914723779243175620369028600689024285845751822691947938945723\
40\
83833

Sendern := 264713306048224785547673145163512922348513168053136441607820278334526819450\
073873355024584847059403376446989563069465193648734087708843091107286958593452\
632\
5563

Sendere := 65537

```
> cipherList := [`86565D261132EE81`, `4B0D890D95C865A9`,
  D52AFD001FCCFFCC, `78600F93392BFA07`, `5AB1CDD776D02ADD`,
  `7BB68AA2DB6A3C1B`, B6759F27A4284CD7, FD53F9D90DB177AA,
  `6516ECC1222224B3`, `23C34DB34EF39F63`, `124097634060239F`,
  `898310A6318C7A92`, AEF132759F7D5397, `28BB9900E6CC162C`,
  "0B75D46D238C10F0", E1E2D7EB76A4AF0E, `7BF31A25D57B348F`,
  C8B796FCBFDE1DC7, `914EEA6D28757828`, `927EF9F61E6AC10E`,
  `242A5DBEA8163FBB`, `68B0AB12088CAB50`, `7FE49BEDEDD5D145`,
  `6B7653EED0B771AA`, C2893218316B5F75, `1F9D1FAC611652A3`,
  `2EF90036717AFD4E`, A1AB2F7B447AC501, F16B493899CBE259,
  `6D01D4F19E583637`, `7C841AD4082C9C43`, `94C3236916FE2DD1`,
  `9B776229BE1194E0`, `67ED95F9A3C58436`, A8D70F6128457B32,
  `3F104AEEDC282829`, C8D71EACBA8EAE6D, D9049802E24F3896,
  `3ED65191800E8BF5`, DF511D402762740F, F37E53B702950F16,
  `9EDB145613D71FF2`];

  sigHex :=
  `1599DCB27215FB1B314EEEC88017D2B824C993DA079F92E38486E44EC40225A85319E
  1B14DEE621B6E7D8BCC9D0028796A16144E770F080EE66273CCA59BA3B2A8CD1`;

  encryptKeyHex
```

```
:=`10F36CDAE57F665A367984588EC59ACACFD57D31338F7C119F57E5AFE25B13E06BB
669508542CE257DB955EE4BD8840830EE5C728A470EEC8E267BBCB113000F29C5E`;
```

cipherList := [86565D261132EE81, 4B0D890D95C865A9, D52AFD001FCCFFCC,
78600F93392BFA07, 5AB1CDD776D02ADD, 7BB68AA2DB6A3C1B, B6759F27A4284CD7,
FD53F9D90DB177AA, 6516ECC1222224B3, 23C34DB34EF39F63, 124097634060239F,
898310A6318C7A92, AEF132759F7D5397, 28BB9900E6CC162C, "0B75D46D238C10F0",
E1E2D7EB76A4AF0E, 7BF31A25D57B348F, C8B796FCBFDE1DC7, 914EEA6D28757828,
927EF9F61E6AC10E, 242A5DBEA8163FBB, 68B0AB12088CAB50, 7FE49BEDEDD5D145,
6B7653EED0B771AA, C2893218316B5F75, 1F9D1FAC611652A3, 2EF90036717AFD4E,
A1AB2F7B447AC501, F16B493899CBE259, 6D01D4F19E583637, 7C841AD4082C9C43,
94C3236916FE2DD1, 9B776229BE1194E0, 67ED95F9A3C58436, A8D70F6128457B32,
3F104AEEDC282829, C8D71EACBA8EAE6D, D9049802E24F3896, 3ED65191800E8BF5,
DF511D402762740F, F37E53B702950F16, 9EDB145613D71FF2]

sigHex := 1599DCB27215FB1B314EEEC88017D2B824C993DA079F92E38486E44EC40225A85319E1\
B14DEE621B6E7D8BCC9D0028796A16144E770F080EE66273CCA59BA3B2A8CD1

encryptKeyHex := 10F36CDAE57F665A367984588EC59ACACFD57D31338F7C119F57E5AFE25B13\
E06BB669508542CE257DB955EE4BD8840830EE5C728A470EEC8E267BBCB113000F29C5E

VERIFICADO DE LA FIRMA.

```
> signature := convert(sigHex, decimal, hex);

  sigD := Power(signature,Sendere) mod Sendern;

  hashRecover := substring(convert(2^64+sigD,hex),2..17);
```

signature := 11862932621957864817868582045035660635958731894976135851319449015536066768\
6378165059492850669219079261534388885754503635730280373162934968199596680761901\
223121

sigD := 11446765237824856050

hashRecover := 9EDB145613D71FF2

EL RECEPTOR COMPUTA LA CLAVE DE SESION.

```
> encryptKey := convert(encryptKeyHex,decimal,hex);

  keyNum := Power(encryptKey, Receiverd) mod Receivern:
  sessionKey := substring(convert(keyNum + 2^64, hex),2..17);
```

encryptKey := 9309148769655280501063089801724909873150946243346389358425110650722385931\
9126364968404717155152828574290923888491220724315027719128148481578153665356856\
0503902

sessionKey := 7778EEFFF432E596

AHORA, EL RECPTOR COMIENZA DESCIFRAR EL MENSAJE CON DES en modo CBC.

```
> key := keyexpander(sessionKey);

  allZeroes := substring(convert(2^64+0,hex),2..17);

  IV := qdDEShex(allZeroes,key);
```

```
key := ["1111111110101111100011101101011100110111000011101",
        "1110001100111111111011110010110110010110110101111",
        "1111100111111110111001010101111111100100111001111",
        "1101000111111111111110100010111011001101110011101",
        "1111010011110111111101111000101011110101110101111",
        "1111011111011111010001111110111111000111101000001",
        "0110101111110011111101111101101001001111010010111",
        "1011110111010101111111111101111011010011000111100",
        "1100111111011111110110101000110110001100111111110",
        "0111111011111011111110110100110111111111011010101",
        "1011111111111101010010110011101111000100111111101",
        "0110101101101111110111111000101111011101100000111",
        "0111110111111101100111011000111001100111110110101",
        "0101011110101101111110111111110110100101111000101",
        "1111111111101100101101111101001011000011100110111",
        "1001110111111111001011111011010110101101110111110"]
```

allZeroes := 0000000000000000

IV := "04844CB2337C0CC2"

```
> messWords := linalg[vectdim](cipherList);
```

messWords := 42

```
> plainTable[1] := xor64hex(unDEShex(cipherList[1],key), IV):
  for i from 2 to messWords do
  plainTable[i] := xor64hex(unDEShex(cipherList[i],key),
  cipherList[i-1]);
  od:
```

```
> plainList := convert(plainTable,list);
```

plainList := [204E6F2063616265, 2064756461207175, 65206C6120696E66, 6F726D616369F36E,
206573206C61206D, 61796F7220667565, 6E74652064652070, 6F64657220717565,
20686120636F6E6F, 6369646F206C6120, 68756D616E696461, 6420792071756520,
6C61206372697074, 6F67726166ED6120, 657320756E612068, 657272616D69656E,
74612066756E6461, 6D656E74616C2070, 617261207375206 3, 6F6E74726F6C2E20,
5365207072657465, 6E64652070726573, 656E74617220756E, 206465736172726F,
6C6C6F2068697374 , F37269636F206465, 206573746120636 9, 656E636961206D6F,
6465726E612C2063, 6F6E20656C206F62, 6A657469766F2064, 6520717565207365,
20636F6E6F7A6361, 6E20737357207665, 6E74616A61732065, 20696E636F6E7665,
6E69656E7465732C, 207375732070656C, 6967726F73207920, 6C6579656E646173,
2E2053616C75646F, 732E207800000000]

FINALMENTE, SE CONVIERTE A UNA CADENA DE CARACTERES ASCII,

```
> plainListBytes := map(x ->
  seq(convert(substring(x,2*i+1..2*i+2),decimal,hex),i=0..7),
     plainList):
```

```
> received := convert(plainListBytes,bytes);
```

received := " No cabe duda que la información es la mayor fuente de poder que ha conocido la humanidad y que la criptografía es una herramienta fundamental para su control. Se pretende presentar un desarrollo histórico de esta ciencia moderna, con el objetivo de que se conozcan sus ventajas e inconvenientes, sus peligros y leyendas. Saludos. x"

LO PASAMOS DE UNA CADENA A PROPIAMENTE TEXTO.

```
> convert(received, name);
```

No cabe duda que la información es la mayor fuente de poder que ha conocido la humanidad y que la criptografía es una herramienta fundamental para su control. Se pretende presentar un desarrollo histórico de esta ciencia moderna, con el objetivo de que se conozcan sus ventajas e inconvenientes, sus peligros y leyendas. Saludos. x

```
>
>
```