> (Gen-Decoder-A '((h e l l o) (t h i s) (i s) (t e s a)))

(#t #t #t #f)

> (define add5 (encode-n 5))

> (encode-d document add5)

(((y m n x) (h t z w x j) (h t a j w x) (y t u n h x) (n s) (u w t l w f r r n s l) (q f s l z f l j x) (f s i) (h t r u n q j w x))

((f y y w n g z y j) (l w f r r f w x) (f s i) (y m j n w) (z x j) (n s) (x d s y f c) (i n w j h y j i) (y w f s x q f y n t s))

((r t i j q x) (t k) (u w t l w f r r n s l) (q f s l z f l j) (x j r f s y n h x))

((n s y j w r j i n f y j) (w j u w j x j s y f y n t s x) (t k) (u w t l w f r x) (u f w f q q j q) (u w t l w f r r n s l) (r t i j q x)))

> (encode-d test-document add5)

(((m j q q t)) ((y m n x) (n x) (y j x y)))

> (Gen-Decoder-A '((m j q q t)) ((y m n x) (n x) (y j x y)))

. . y: undefined;

cannot reference undefined identifier

> (Gen-Decoder-A '((m j q q t) (y m n x) (n x) (y j x y)))

(#f #f #f #f)

> (test1 '(m j q q t))

(l i p p s)

> (Gen-Decoder-A '((m j q q t)))

((m j q q t))

> (test '(m j q q t))

(Gen-Decoder-A '((m j q q t) (y m n x) (n x) (y j x y)))

> (define decoderSP1 (Gen-Decoder-A '((m j q q t))))

> (Code-Breaker document decoderSP1)

(apply-distance '(m j q q t) (find-max-letter (paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)(find-max-frequent(paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha))) most-frequent-letters )

> (define decoderSP1 (Gen-Decoder-A '((m j q q t))))

> (Code-Breaker document decoderSP1)

(((o c d n) (x j p m n z) (x j q z m n) (o j k d x n) (d i) (k m j b m v h h d i b) (g v i b p v b z n) (v i y) (x j h k d g z m n))

((v o o m d w p o z) (b m v h h v m n) (v i y) (o c z d m) (p n z) (d i) (n t i o v s) (y d m z x o z y) (o m v i n g v o d j i))

((h j y z g n) (j a) (k m j b m v h h d i b) (g v i b p v b z) (n z h v i o d x n))

((d i o z m h z y d v o z) (m z k m z n z i o v o d j i n) (j a) (k m j b m v h n) (k v m v g g z g) (k m j b m v h h d i b) (h j y z g n)))

> (Code-Breaker another-document decoderSP1)

(((h e l l o)) ((t h i s) (i s) (t e s t)))

> (apply-distance '(m j q q t) (find-max-letter (paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)(find-max-frequent(paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha))) most-frequent-letters )

(h e l l o)

> (define decoderSP1 (Gen-Decoder-A '((m j q q t))))

> (Code-Breaker document decoderSP1)

(((o c d n) (x j p m n z) (x j q z m n) (o j k d x n) (d i) (k m j b m v h h d i b) (g v i b p v b z n) (v i y) (x j h k d g z m n))

((v o o m d w p o z) (b m v h h v m n) (v i y) (o c z d m) (p n z) (d i) (n t i o v s) (y d m z x o z y) (o m v i n g v o d j i))

((h j y z g n) (j a) (k m j b m v h h d i b) (g v i b p v b z) (n z h v i o d x n))

((d i o z m h z y d v o z) (m z k m z n z i o v o d j i n) (j a) (k m j b m v h n) (k v m v g g z g) (k m j b m v h h d i b) (h j y z g n)))

> (Code-Breaker another-document decoderSP1)

(((h e l l o)) ((t h i s) (i s) (t e s t)))

> (apply-distance '(m j q q t) (find-max-letter (paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)(find-max-frequent(paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha))) most-frequent-letters )

(h e l l o)

> (find-max-letter '((a 2) (b 0) (c 1) (d 0) (e 4) (f 0) (g 1) (h 1) (i 2) (j 0) (k 0) (l 0) (m 1) (n 0) (o 0) (p 0) (q 0) (r 1) (s 5) (t 2) (u 0) (v 0) (w 0) (x 0) (y 0) (z 50)) (find-max-frequent '((a 2) (b 0) (c 1) (d 0) (e 4) (f 0) (g 1) (h 1) (i 2) (j 0) (k 0) (l 0) (m 1) (n 0) (o 0) (p 0) (q 0) (r 1) (s 5) (t 2) (u 0) (v 0) (w 0) (x 0) (y 0) (z 50))))

z

> (paragraph-helper-for-counting'( (t h i s) (i s) (a) (s e c r e t) (m e s s a g e)) alpha)

((a 2) (b 0) (c 1) (d 0) (e 4) (f 0) (g 1) (h 1) (i 2) (j 0) (k 0) (l 0) (m 1) (n 0) (o 0) (p 0) (q 0) (r 1) (s 5) (t 2) (u 0) (v 0) (w 0) (x 0) (y 0) (z 0))

> (count-frequency '(h e l l o) alpha)

((a 0) (b 0) (c 0) (d 0) (e 1) (f 0) (g 0) (h 1) (i 0) (j 0) (k 0) (l 2) (m 0) (n 0) (o 1) (p 0) (q 0) (r 0) (s 0) (t 0) (u 0) (v 0) (w 0) (x 0) (y 0) (z 0))

> (count-frequency '((m j q q t) (y m n x) (n x) (y j x y)) alpha)

((a 0) (b 0) (c 0) (d 0) (e 0) (f 0) (g 0) (h 0) (i 0) (j 0) (k 0) (l 0) (m 0) (n 0) (o 0) (p 0) (q 0) (r 0) (s 0) (t 0) (u 0) (v 0) (w 0) (x 0) (y 0) (z 0))

> (paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)

((a 0) (b 0) (c 0) (d 0) (e 0) (f 0) (g 0) (h 0) (i 0) (j 2) (k 0) (l 0) (m 2) (n 2) (o 0) (p 0) (q 2) (r 0) (s 0) (t 1) (u 0) (v 0) (w 0) (x 3) (y 3) (z 0))

> (find-max-frequent'((a 0) (b 0) (c 0) (d 0) (e 0) (f 0) (g 0) (h 0) (i 0) (j 2) (k 0) (l 0) (m 2) (n 2) (o 0) (p 0) (q 2) (r 0) (s 0) (t 1) (u 0) (v 0) (w 0) (x 3) (y 3) (z 0)))

3

> (find-max-letter '((a 0) (b 0) (c 0) (d 0) (e 0) (f 0) (g 0) (h 0) (i 0) (j 2) (k 0) (l 0) (m 2) (n 2) (o 0) (p 0) (q 2) (r 0) (s 0) (t 1) (u 0) (v 0) (w 0) (x 3) (y 3) (z 0))(find-max-frequent'((a 0) (b 0) (c 0) (d 0) (e 0) (f 0) (g 0) (h 0) (i 0) (j 2) (k 0) (l 0) (m 2) (n 2) (o 0) (p 0) (q 2) (r 0) (s 0) (t 1) (u 0) (v 0) (w 0) (x 3) (y 3) (z 0))))

x

> (find-max-frequent(paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha))

3

> (find-max-letter (paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)(find-max-frequent(paragraph-helper-for-counting '((m j q q t) (y m n x) (n x) (y j x y)) alpha)))

x