

Encryption

Problem description

This is a trivial encryption program which works by shuffling 70 selected characters in sequence:

<space> character after '0'

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 .,-!?()

The encryption scheme is controlled by 3 integers: **Width**(≤ 70), **Height**(≤ 70) and **Skip**. The basic idea is that the scheme wraps the 70 characters (*in sequence*) above to a rectangle of dimension **W** \times **H** in clockwise order starting from the upper-left corner. (You may assume $70 \leq W \times H$, no checking is needed). As an example, if **W**=12, **H**=6, **S**=0:

A	B	C	D	E	F	G	H	I	J	K	L
f	g	h	i	j	k	l	m	n	o	p	M
e	4	5	6	7	8	9	0	.	q	N	
d	3)	(?	!	-	,	r	O
c	2	1	z	y	x	w	v	u	t	s	P
b	a	Z	Y	X	W	V	U	T	S	R	Q

Then the characters are extracted *in columns* from left to right, top to bottom order (and skipping all unused blanks):

AfedcbBg432aCh51ZDi6zYEj7)yXfk8(xWG19?wVHm0!vUIn -uTJo.,tSKpqrsRLMNOPQ

The encryption works by replacing a character from the original sequence to a character in the new sequence at the same position.

Original	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 .,-!?()
New	AfedcbBg432aCh51ZDi6zYEj7)yXfk8(xWG19?wVHm0!vUIn -uTJo.,tSKpqrsRLMNOPQ

For example, 'B' (2nd character in the original sequence) is changed to 'f' (2nd character in the new sequence). Likewise, the message "Hello" will be changed to "g8??H".

The scheme, however has one problem – 'A' (upper-left) is always mapped to 'A'. Therefore the integer **S** is introduced. Instead of wrapping the 70 characters directly, each character is considered as being preceded by **S** unused blanks. (The original scheme with no blank can be considered as $S=0$). As an example, if **W**=36, **H**=6, **S**=2:

	A		B		C		D		E		F		G		H		I		J		K		L
	a		b		c		d		e		f		g		h		i		j		k		l
	y		z		1		2		3		4		5		6		7		8		9		
Z)		(?		!		-		,		.				0		M
	x		w		v		u		t		s		r		q		p		o		n		m
	Y		X		W		V		U		T		S		R		Q		P		O		N

Since $S=2$, character 'A' is preceded by 2 blanks and therefore it's no longer the first character. The new sequence (*ignoring unused blanks*) will be:

ZaxYAybwXBzcvWC1)duVD2(etUE3?fsTF4!grSG5-hqRH6,ipQI7.joPJ8 knOK90lmNLM
 We assume $70 \cdot (1+S) \leq W \cdot H$, no checking is needed.

Implementation

We will develop a program based on the template file.

In the file, there is a Codec class with the following private members:

int W , H , S	Width, Height, Skip values for the scheme.
char C [71]	The 70 selected characters in sequence.
char M [70][70]	The 2 dimensional matrix used for shuffling.

The task is to complete the implementation of the following public items:

Codec()	Default constructor with no shuffling (W =70, H =1, S =0)
Codec (int w, int h, int s)	Parameterized constructor taking W , H , S
void config (int w, int h, int s)	Change the parameters of the encryption system. Could possibly be called multiple times before the program finish.
void showSetting ()	Display the encoding parameters, 2D matrix and the new character sequence in the same format as sample output. The printed matrix should be in dimension of W * H (<i>i.e. no extra trailing spaces before <enter></i>) and unused blanks in the 2D matrix should be printed as space characters.
void encode (char in[], char out[])	Encode a given cString in [] to another cString out [] using the current codec settings. Characters outside sequence C (<i>e.g. symbols like the '@' mark</i>) will be copied without change.
void decode (char in[], char out[])	Decode a given cString in [] to another cString out [] using the current codec settings. Characters outside sequence C (<i>e.g. symbols like the '@' mark</i>) will be copied without change.

You may add in new variables and functions but you're not allowed to change the existing items in the template (*e.g. changing the type or count of function input, or changing public/private settings are not allowed*). `<string>` library and `string` class are not allowed. You may use `<iostream>`, `<iomanip>` and `<cstring>` whenever appropriate.

The program will keep reading single-character commands (*in uppercase*) until 'Q' (quit) is encountered.

'C' command is used to configure the encryption system, with integers **W**, **H** and **S** on the same line.

'S' command is used to display the current setting (which could possibly be changed by the 'C' command). Besides showing **W**, **H**, and **S**, it also displays the 2D matrix and the new character sequence (*of the 70 characters*).

'E' and 'D' commands are used for encryption and decryption respectively. The message is separated from the 'E' / 'D' command by exactly one space character. The message length is less than 1000 character and it may contain spaces or characters outside the selected sequence of 70 characters.

Sample Input / Output:

Example 1: (User Input is underlined)

```
Command: S
W=70, H=1, S=0
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 .,-!()?
Seq: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 .,-!()?

Command: C 12 6 0

Command: S
W=12, H=6, S=0
ABCDEFGHIJKL
fghijklmnopM
e4567890 .qN
d3 )(?!-,r0
c21zyxwvutsP
baZYXWVUTSRQ
Seq: AfedcbBg432aCh51ZDi6zYEj7)yXfk8(xWGl9?wVHm0!vUIn -uTJo.,tSKpqrsRLMNOPQ

Command: E Hello@
g8?H@

Command: Q
Quit
```

Example 2: (User Input is underlined)

```
Command: C 36 6 2

Command: S
W=36, H=6, S=2
  A B C D E F G H I J K L
a b c d e f g h i j k l
y z 1 2 3 4 5 6 7 8 9
Z      ) ( ? ! - , . 0 M
x w v u t s r q p o n m
Y X W V U T S R Q P O N
Seq: ZaxYAybwXBzcvWC1)duVD2(etUE3?fsTF4!grSG5-hqRH6,ipQI7.joPJ8 knOK90lmNLM

Command: D (sSSKY-5sm
Well Done!

Command: Q
Quit
```