Swinburne University Of Technology

Faculty of Information and Communication Technologies

ASSIGNMENT COVER SHEET

Subject Code:	HIT3303/8303
Subject Title:	Data Structures & Patterns
Assignment number and title:	2 - Arrays, Indexers, and Dynamic Link Libraries
Due date:	**March 30, 2011, 10:30 a.m., on paper**
Lecturer:	Dr. Markus Lumpe

Your name:____________________

Marker's comments:

Problem	Marks	Obtained
1	99	
2	25	
3	2	
4	12	
Total	138	

Extension certification:

This assignment has been given an extension and is now due on ___________

Signature of Convener:________________________

Problem Set 2: Arrays, Indexers, and Dynamic Link Libraries

Around 1550 Blaise de Vigenère, a French diplomat from the court of Henry III of France, developed a new scrambling technique that uses 26 alphabets to cipher a text. The *Vigenère Cipher* is a polyalphabetic substitution technique based on the following *tableau:*

Key\Letter	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

The Vigenère cipher uses this table together with a keyword to encode a message.

To illustrate the use of this encryption method, suppose we wish to scramble the following message (Hamlet 3/1):

```
To be, or not to be: that is the question:
```

using the keyword `Relations`. First, we notice that the table provides only a mapping for upper case characters. But this is not really a problem. The mapping is identical for upper and lower case characters. We just rewrite the keyword to consist of upper case characters only. When encoding the message we convert each character to an upper case one, perform the corresponding encryption function, and output the result in either upper case or lower case depending on the original spelling. All characters not covered in the Vigenère cipher remain unchanged in the output. No keyword character is consumed in this case!

We begin by writing the keyword, repeated as many times as necessary, above the message. To derive the encoded text using the tableau, for each letter in the message, one finds the intersection of the row given by the corresponding keyword letter and the column given by the message letter itself to pick out the encoded letter.

```
Keyword:           RE LA  TI ONS RE LA  TION SR ELA TIONSREL
Message:           To be, or not to be: that is the question:
Scrambled Message: Lt nf, ia ccm lt nf: nqph bk ytf kdtgmatz:
```

Decoding of an encrypted message is equally straightforward. One writes the keyword repeatedly above the message:

```
Keyword:           RE LA  TI ONS RE LA  TION SR ELA TIONSREL
Scrambled Message: Lt nf, ia ccm lt nf: nqph bk ytf kdtgmatz:
Decoded Message:   To be, or not to be: that is the question:
```

This time one uses the keyword letter to pick a row of the table and then traces the row to the column containing the encoded letter. The index of that column is the decoded letter.

Problem 1: Dynamic Link Library

Implement a C++ dynamic link library called Vigenere (Windows file name Vigenere.dll, Unix file name libVigenere.so, and MacOS file name libVigenere.dylib). That is, define a header file Vigenere.h defining the class Vigenere and a C++ file Vigenere.cpp that implements this class:

```
#include <string>

class Vigenere
{
private:
  char fCharacterMap[26][26];
  std::string fKey;
  unsigned int fKeyIndex;
  int fSourceFrequency[26];
  int fTargetFrequency[26];
  int fTotalEncoded;

public:
  Vigenere( char* aKey );

  void resetFrequencies();
  char encode( char aCharacter );
  char decode( char aCharacter );
  std::string encode( const std::string& aString );
  std::string decode( const std::string& aString );

  friend std::ostream& operator<<( std::ostream& aOStream,
                                   const Vigenere& aScrambler );
};
```

There are two sets of encryption methods: one for characters and one for strings. The character-based methods implement the Vigenère cipher process. That is, the character-based methods encode and decode perform the encryption for the given argument aCharacter. More precisely, both methods check first whether aCharacter is a letter and whether it is a lower case character. Then the methods proceed by using fKey, fKeyIndex, and aCharacter to determine the Vigenère mapping, which is returned as the result. Both methods advance the key index, fKeyIndex, if an encryption has occurred. We use a "round-robin" approach to select the next key character from fKey.

The string-based methods encode and decode drive the encryption of strings. Both methods iterate over the contents of aString and return a new string, whose contents has been computed by the character-based methods encode and decode.

The class Vigenere also supports character frequency analysis. This feature provides you with the ability to assess the quality of the encryption. The variables fSourceFrequency contains the input counts, whereas fTargetFrequency records the output counts for characters. The variable fTotalEncoded represents the total number of characters processed. These variables can be used to compute the percentage of the occurrence of a given letter in text both clear and encoded. For example, if a text contains fTotalEncoded characters, then G's percentage is fSourceFrequency[6]*100/fTotalEncoded, where 6 = 'G' - 'A' is the index of the 7th letter in the 26 character English alphabet.

Problem 2: Scramble

Using the dynamic link library `Vigenere` implement the C++ console application `scramble` that takes two arguments `key` and `file_name` and encodes the text file named `file_name`:

```
scramble “Too many secrets” sample.txt
```

generates the file `sample.txt.secure.txt`, the encoded version of `sample.txt`. In addition, at the end `scramble` has to print the character frequencies to the console using the operator <<.

The encoding of a text file has to be implemented in a while loop:

```
while ( getline( lReader, lLine ) )
{
    …
}
```

Remember to set the system-specific environment to locate the `Vigenere` shared library (Windows: `PATH`, Linux: `LD_LIBRARY_PATH`, and MacOS: `DYLD_LIBRARY_PATH`).

One note on the keyword: In the above example, the keyword is “`Too many secrets`.” In order to use this string as a Vigenère cipher key, we need to convert it to an upper case string and remove all non-letter characters: TOOMANYSECRETS. This should be done in the initialization of the `Vigenere` constructor.

Sample console output:

```
Scrambling 'sample.txt' using key: "Too many secrets"
Frequency distribution:
Char:      A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
Input:     8   1   2   5  11   2   1   5   5   0   0   3   3   6   7   1   0   6   7   8   4   0   1   0   1   0
Output:    3   3   3   3   2   5   5   5   4   3   2   4   3   4   2   3   2   3   4   5   4   3   4   5   3   2
```

Problem 3: Unscramble

Using the dynamic link library `Vigenere` implement the C++ console application `unscramble` that takes two arguments `key` and `file_name` and decodes the text file named `file_name`:

```
unscramble "Too many secrets" sample.txt.secure.txt
```

produces the file `sample.txt.secure.txt.public.txt`, the decoded version of `sample.txt.secure.txt`. In addition, at the end `unscramble` has to print the character frequencies to the console using the operator `<<`.

The decoding of a text file has to be implemented in a while loop:

```
while ( getline( lReader, lLine ) )
{
    …
}
```

Sample output:

```
Unscrambling 'sample.txt.secure.txt' using key: "Too many secrets"
Frequency distribution:
Char:     A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
Input:    3   3   3   3   2   5   5   5   4   3   2   4   3   4   2   3   2   3   4   5   4   3   4   5   3   2
Output:   8   1   2   5  11   2   1   5   5   0   0   3   3   6   7   1   0   6   7   8   4   0   1   0   1   0
```

Sample.txt (Richard III)

```
                  ACT I

                   SCENE I

              London. A Street.

              Enter Gloucester.

Gloucester. Now is the winter of our discontent
      Made glorious summer by this sun of York;
      And all the clouds that lour'd upon our house
      In the deep bosom of the ocean buried.
      Now are our brows bound with victorious wreaths;
      Our bruised arms hung up for monuments;
      Our stern alarums changed to merry meetings;
      Our dreadful marches to delightful measures.
      Grim-visag'd war hath smooth'd his wrinkled front;
      And now, - instead of mounting barbed steeds,
      To fright the souls of fearful adversaries, -
      He capers nimbly in a lady's chamber
      To the lascivious pleasing of a lute.
      But I, that am not shap'd for sportive tricks,
      Nor made to court an amorous looking-glass;
      I, that am rudely stamp'd, and want love's majesty
      To strut before a wanton ambling nymph;
      I, that am curtail'd of this fair proportion,
      Cheated of feature by dissembling nature,
      Deform'd, unfinish'd, sent before my time
      Into this breathing world, scarce half made up,
      And that so lamely and unfashionable
      That dogs bark at me, as I halt by them;
      Why, I, in this weak piping time of peace,
      Have no delight to pass away the time,
      Unless to see my shadow in the sun
      And descant on mine own deformity:
      And therefore, since I cannot prove a lover,
      To entertain these fair well-spoken days,
      I am determined to prove a villain,
      And hate the idle pleasures of these days.
      Plots have I laid, inductions dangerous,
      By drunken prophecies, libels, and dreams,
      To set my brother Clarence and the king
      In deadly hate the one against the other:
      And if King Edward be as true and just
      As I am subtle, false, and treacherous,
      This day should Clarence closely be mew'd up,
      About a prophecy, which says, that G
      Of Edward's heirs the murderer shall be.
      Dive, thoughts, down to my soul: here Clarence comes.

      Brother, good day: what means this armed guard
      That waits upon your Grace?
```

Sample.txt.secure.txt (“Too many secrets”)

```
                  URI V

                   TQDGJ L

              Dthwic. P Fufdxy.

              Hfyyk Aadhdsrmju.

Yqinwthgff. Mhb lk ybx qxcgff ny txj iclwdcgfbs
      Ffgw lfhlxdht gtfrhj gs mbxh fvb ny Drjp;
      Ugx pay uvd vqrmim mbpi ypiq'w zsgs inl wdhts
      Hg ykw iyxj qdfpa ny ykw twxuc qhswdw.
      Sro flx ijg oscvl grmsx pciw ijqshwlgzm pltpgig;
      Nnw ejzclys peng gnsj mu zhl bdavadgyv;
      Gzl lntga bzzkzpk hbthvtq uc lxwuq ryxnxctt;
      Ctk iuwfxyoa bnsqgxx wg iyecvwggik fjdkzlxm.
      Vgvn-jhlfj'v buk bpiu tanhyk'v mcl qgxalzdw kugsn;
      Ths cbx, - wmlyhsi iy gdjauwmz gdjgyw mitreg,
      Sh kualbm nwt fpikl ti xjukzja nejdkxdjnyl, -
      Bt rnqsql slegfr cc p ybrx'l hksrvxl
      Id gis ktxfaachoh eyforbsj gk u eoit.
      Ovh H, mmdl fg gii hubd'c ytu kuiknxkr ufhvpv,
      Ftl fust gp qnnww ss ufigdht znhplfl-aeuhh;
      V, uvzm fp jzxxfn hgbao'w, fqv bugn adif'g ltohkys
      Mi hievh axkrjj u pucibo oluqlfl hrgew;
      V, uvzm fp uzlmuxa'q pt sanv xfck jgdcpfsbtq,
      Umytnts bg tdtyxjj vr xxhffaaenqy sumogt,
      Qftnkr'g, mszbhxhu'e, gdgy ewkiky bn gjad
      Bswg ybbm qgrbhgbsj otlex, hrnsqd afox ruwy je,
      Nor safw kt ftgtal bbc nsisxbbicpoms
      Safw vtal vpgx bh lx, fv A muen qn gisl;
      Pmb, A, nh mbxh jfoj insasa mcbt bg ddthh,
      Zfpx hd srmwfay wg uulm plnz hgx ylej,
      Ogfthf uc rxj pq xbtxdl vo hgx xxf
      Fhw xthpbbs hs pasy hqc srgcqfnwq:
      Fhw nwteftnkj, vaswx C rpaocs iwrnj u eikte,
      Uc dgyhjyubh iwrts etnu ojfe-medxfb ctdv,
      A fg wyitenwmxi wg ulhpt p ijzktnq,
      Ssx auit gis hwqh hqytmjgrt ce mmhkj xtsh.
      Eyphr afyw N ftcs, xaeibmnrfx xthvtepir,
      Ud gjzhdyc eepdgxhlwx, fbvtaf, bbc wwhsrm,
      Mi htg nm aktwzjl Vfpgroqd tsg lmy dccv
      Vo rdtioq mumy iwr pbd tldasmm nwt buvdk:
      Fqv nz Dccv Rekzki ew fm mljt nor inxw
      Sx C tg hjouzd, yfokj, ugx igrbqgxwrmx,
      Nach snz gghzov Hftltcpf qkhxhdd vx gtl'q vd,
      Zutxl f jkiewrdm, vanfz xurm, iwnu U
      Ny Jgoflw'm wtvsg saj pmwxxltg fioke gh.
      Vnpx, nwdhhvsl, iros nh gn hbvz: gxwh Uqukycrr dclxx.

      Ejtnayg, vbpr ctd: zzfn fypcf uvhl fuejx zopgq
      Uvzm bdaym njdc lpiq Zwduj?
```

Problem 4: Indexer

Define a Vigenère indexer adhering to the following class specification:

```
#include "Vigenere.h"

class VigenereIndexer
{
private:
  Vigenere fCipher;
  bool fMode;

public:
  VigenereIndexer( char* aKey, bool aMode );

  char operator[]( const char aChar );

  friend std::ostream& operator<<( std::ostream& aOStream,
                                   const VigenereIndexer& aIndexer );
};
```

A Vigenère indexer is an object that is initialized with a code word aKey and an encryption modus aMode (i.e., aMode == true for encoding, aMode == false for decoding). The indexer defines an on-the-fly encryption mechanism. Each consecutive use of the [] operator will yield the corresponding encoded or decoded character.

Build a program using the Vigenère indexer and change the main function to contain a while loop as follows (lScrambler is the corresponding indexer):

```
char lChar;
while ( (lChar = lReader.get()) != EOF )
{
      lOutput << lScrambler[lChar];
}
```

The C++ console application VigenereIndexer that takes three arguments mode, key and file_name and decodes the text file named file_name:

```
VigenereIndexer encode “Too many secrets” sample.txt
```

encodes the file sample.txt and emits a new file, sample.txt.encode.txt, the encoded version of sample.txt.

Decoding is similar:

```
VigenereIndexer decode “Too many secrets” sample.txt.encode.txt
```

decodes the file sample.txt.encode.txt and emits a new file, sample.txt.encode.txt.decode.txt, with the original content of sample.txt.

Submission deadline: Wednesday, March 30, 2010, 10:30 a.m.

Submission procedure: on paper, in class.