

Swinburne University of Technology
Faculty of Science, Engineering and Technology

MIDTERM COVER SHEET

Subject Code: COS30008
Subject Title: Data Structures and Patterns
Assignment number and title: Midterm, Solution Design, Design Pattern, and Iterators
Due date: April 27, 2022, 23:59
Lecturer: Dr. Markus Lumpe

Your name: _____ **Your student ID:** _____

Check Tutorial	Mon 10:30	Mon 14:30	Tues 08:30	Tues 10:30	Tues 12:30	Tues 14:30	Tues 16:30	Wed 08:30	Wed 10:30	Wed 12:30	Wed 14:30

Marker's comments:

Problem	Marks	Obtained
1	68	
2	120	
3	56	
4	70	
Total	314	

Problem 1 – KeyProvider.cpp

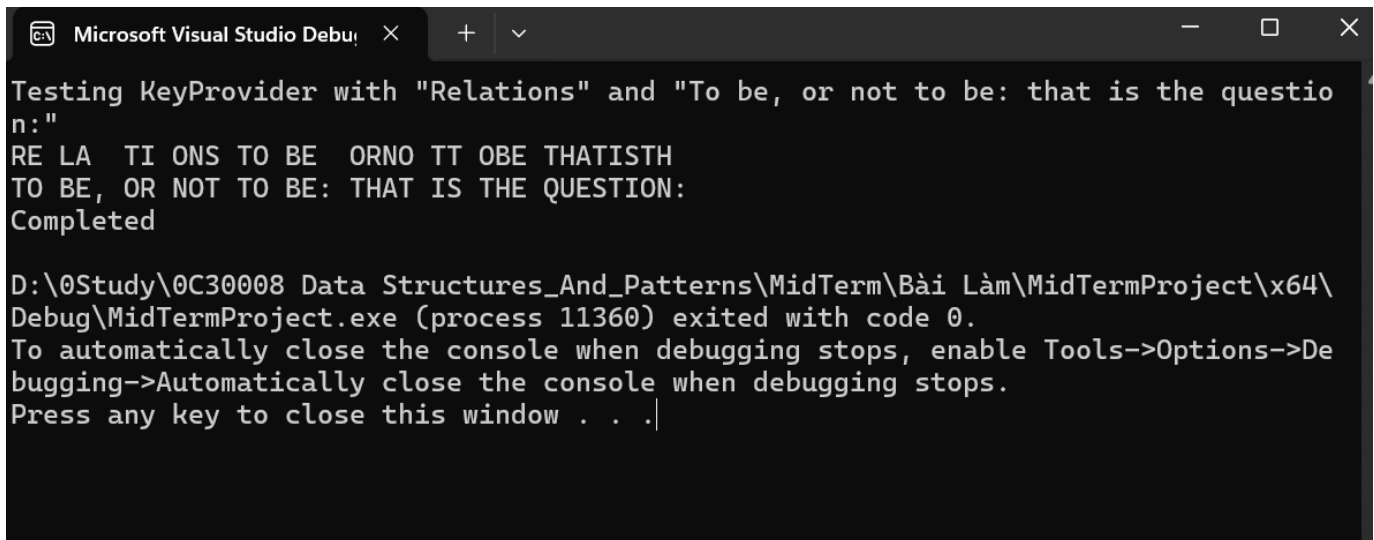
...Bài Làm\MidTermProject\MidTermProject\KeyProvider.cpp

1

```
1 #include "KeyProvider.h"
2 #include <cctype> // Thư viện để sử dụng hàm toupper()
3
4 using namespace std;
5
6 // Khởi tạo đối tượng KeyProvider, thiết lập ban đầu cho từ khóa
7 KeyProvider::KeyProvider(const string& aKeyword) :
8     fKeyword(nullptr), // Đặt con trỏ fKeyword thành nullptr
9     fSize(0),          // Đặt kích thước từ khóa ban đầu là 0
10    fIndex(0)           // Đặt chỉ mục từ khóa là 0
11 {
12     initialize(aKeyword); // Gọi hàm khởi tạo từ khóa với aKeyword
13 }
14
15 // Destructor giải phóng bộ nhớ đã cấp phát cho từ khóa
16 KeyProvider::~KeyProvider()
17 {
18     delete[] fKeyword; // Giải phóng bộ nhớ đã cấp phát
19 }
20
21 // Hàm khởi tạo lại từ khóa với chuỗi mới
22 void KeyProvider::initialize(const string& aKeyword)
23 {
24     delete[] fKeyword; // Xóa từ khóa cũ nếu tồn tại
25
26     fSize = aKeyword.length(); // Cập nhật độ dài từ khóa mới
27     fKeyword = new char[fSize]; // Cấp phát bộ nhớ cho từ khóa mới
28     fIndex = 0;                // Đặt lại chỉ mục về 0
29
30     for (size_t i = 0; i < fSize; ++i) // Duyệt qua từng ký tự của từ khóa mới
31     {
32         fKeyword[i] = toupper(aKeyword[i]); // Chuyển mỗi ký tự thành chữ in hoa
33     }
34 }
35
36 // Toán tử dereference trả về ký tự hiện tại của từ khóa
37 char KeyProvider::operator*() const
38 {
39     return fKeyword[fIndex]; // Trả về ký tự tại vị trí fIndex hiện tại
40 }
41
42 // Toán tử dịch trái, thêm một ký tự mới và di chuyển chỉ mục
43 KeyProvider& KeyProvider::operator<<(char aKeyCharacter)
44 {
45     fKeyword[fIndex] = toupper(aKeyCharacter); // Thay thế ký tự hiện tại bằng ký tự mới
46 }
```

```
47     ++fIndex; // Tăng chỉ mục lên 1
48
49     if (fIndex == fSize) // Nếu chỉ mục vượt quá độ dài từ khóa
50     {
51         fIndex = 0; // Đặt lại chỉ mục về 0 (vòng lặp tuần hoàn)
52     }
53
54     return *this; // Trả về chính đối tượng để hỗ trợ xâu chuỗi
55 }
56
```

Output:

The image shows a screenshot of the Microsoft Visual Studio Debug Console. The window title is "Microsoft Visual Studio Debug Console". The output text is as follows:

```
Testing KeyProvider with "Relations" and "To be, or not to be: that is the question:"
RE LA TI ONS TO BE ORNO TT OBE THATISTH
TO BE, OR NOT TO BE: THAT IS THE QUESTION:
Completed

D:\0Study\0C30008 Data Structures_And_Patterns\MidTerm\Bài Làm\MidTermProject\x64\
Debug\MidTermProject.exe (process 11360) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->De
bugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

Problem 2 – Vigenere MT.cpp

...Bài Làm\MidTermProject\MidTermProject\Vigenere MT.cpp

1

```
1 #include "Vigenere.h"
2 #include <cctype> // Thư
   viện để sử dụng các hàm kiểm tra ký tự
3
4 using namespace std;
5
6 // Hàm khởi tạo Vigenere, thiết lập từ khóa và chuẩn bị bảng mã hóa
7 Vigenere::Vigenere(const string& aKeyword) :
8     fKeyword(aKeyword), // Gán
   từ khóa ban đầu
9     fKeywordProvider(aKeyword) // Khởi
   tạo KeyProvider với từ khóa
10 {
11     initializeTable(); // Gọi
   hàm khởi tạo bảng mã hóa
12 }
13
14 // Trả về từ khóa hiện tại, được cung cấp bởi KeyProvider
15 string Vigenere::getCurrentKeyword()
16 {
17     string currentKeyword; //
   Chuỗi chứa kết quả
18
19     for (size_t i = 0; i < fKeyword.size(); ++i) // Lặp
   qua từng ký tự của từ khóa
20     {
21         char keyCharacter = *fKeywordProvider; // Lấy
   ký tự hiện tại từ KeyProvider
22
23         currentKeyword += keyCharacter; // Thêm
   ký tự vào chuỗi kết quả
24         fKeywordProvider << keyCharacter; // Đẩy
   ký tự và cập nhật KeyProvider
25     }
26
27     return currentKeyword; // Trả
   về từ khóa hiện tại
28 }
29
30 // Đặt lại KeyProvider về trạng thái ban đầu
31 void Vigenere::reset()
32 {
33     fKeywordProvider.initialize(fKeyword); // Khởi
   tạo lại từ khóa trong KeyProvider
34 }
35
36 // Mã hóa ký tự dựa trên từ khóa và bảng mã hóa
37 char Vigenere::encode(char aCharacter)
38 {
```

```

...Bài Làm\MidTermProject\MidTermProject\Vigenere MT.cpp 2
39     if (isalpha(aCharacter)) // Kiểm tra
    tra nếu ký tự là chữ cái
40     {
41         bool isLower = islower(aCharacter); // Kiểm tra
    tra ký tự có phải là chữ thường không
42         aCharacter = toupper(aCharacter); // Chuyển
    ký tự thành chữ in hoa
43
44         // Tính toán chỉ số hàng và cột
45         char rowIndex = *fKeywordProvider - 'A'; // Lấy
    hàng tương ứng trong bảng mã hóa
46         char columnIndex = aCharacter - 'A'; // Lấy
    cột tương ứng trong bảng mã hóa
47
48         fKeywordProvider << aCharacter; // Cập nhật
    KeyProvider với ký tự hiện tại
49
50         aCharacter = fMappingTable[rowIndex][columnIndex]; // Mã
    hóa ký tự dựa trên bảng
51
52         if (isLower) // Nếu
    ký tự ban đầu là chữ thường
53         {
54             aCharacter = tolower(aCharacter); // Chuyển
    ký tự đã mã hóa thành chữ thường
55         }
56     }
57
58     return aCharacter; // Trả
    về ký tự đã mã hóa
59 }
60
61 // Giải mã ký tự dựa trên từ khóa và bảng mã hóa
62 char Vigenere::decode(char aCharacter)
63 {
64     if (isalpha(aCharacter)) // Kiểm tra
    tra nếu ký tự là chữ cái
65     {
66         bool isLower = islower(aCharacter); // Kiểm tra
    tra ký tự có phải là chữ thường không
67         aCharacter = toupper(aCharacter); // Chuyển
    ký tự thành chữ in hoa
68
69         char rowIndex = *fKeywordProvider - 'A'; // Lấy
    hàng tương ứng trong bảng mã hóa
70
71         for (char columnIndex = 0; columnIndex < CHARACTERS; +
    columnIndex) // Duyệt qua cột để tìm ký tự đã mã hóa
72         {

```

```

...Bài Làm\MidTermProject\MidTermProject\Vigenere MT.cpp 3
73         if (aCharacter == fMappingTable[rowIndex][columnIndex]) // ➤
            Kiểm tra nếu ký tự khớp với cột hiện tại
74         {
75             aCharacter = 'A' + columnIndex; // ➤
            Khôi phục ký tự ban đầu
76             break; // ➤
            Thoát khỏi vòng lặp khi tìm thấy
77         }
78     }
79
80     fKeywordProvider << aCharacter; // Cập ➤
        nhật KeyProvider với ký tự hiện tại
81
82     if (isLower) // Nếu ➤
        ký tự ban đầu là chữ thường
83     {
84         aCharacter = tolower(aCharacter); // ➤
        Chuyển ký tự đã giải mã thành chữ thường
85     }
86 }
87
88 return aCharacter; // Trả ➤
    về ký tự đã giải mã
89 }

```

Output:

```

Microsoft Visual Studio Debug Console
Encoding "To be, or not to be: that is the question:" using "RELATIONS"
TO BE, OR NOT TO BE: THAT IS THE QUESTION:
Lt nf, ia ccm nd dj: izoi cm ij j kcfmbiv:
Completed
Decoding "Lt nf, ia ccm nd dj: izoi cm ij j kcfmbiv:" using "RELATIONS"
LT NF, IA CCM ND DJ: IZOI CM IJJ KCFMCBIV:
To be, or not to be: that is the question:
Completed

D:\0Study\0C30008 Data Structures_And_Patterns\MidTerm\Bài Làm\MidTermProject\x64\
Debug\MidTermProject.exe (process 44924) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->De
bugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Problem 3 – iVigenereStream.cpp

...Lam\MidTermProject\MidTermProject\iVigenereStream.cpp

1

```
1  #include "iVigenereStream.h"
2
3  using namespace std;
4
5  // Constructor cho lớp iVigenereStream, khởi tạo cipher provider và mở tệp
6  iVigenereStream::iVigenereStream(Cipher aCipher,
7      const std::string& aKeyword,
8      const char* aFileName) :
9      fCipherProvider(aKeyword),
10     fCipher(aCipher)
11 {
12     open(aFileName);
13 }
14
15 // Destructor cho lớp iVigenereStream, đảm bảo tệp được đóng
16 iVigenereStream::~iVigenereStream()
17 {
18     close();
19 }
20
21 // Phương thức mở tệp với tên tệp được cung cấp
22 void iVigenereStream::open(const char* aFileName)
23 {
24     if (aFileName)
25         fIStream.open(aFileName, ifstream::binary);
26 }
27
28 // Phương thức đóng tệp hiện tại
29 void iVigenereStream::close()
30 {
31     if (is_open())
32     {
33         fIStream.close();
34     }
35 }
36
37 // Phương thức đặt lại trạng thái của cipher provider
38 void iVigenereStream::reset()
39 {
40     fCipherProvider.reset();
41
42     if (is_open())
43     {
44         seekstart();
45     }
46 }
47
48 // Kiểm tra trạng thái tốt của luồng tệp
49 bool iVigenereStream::good() const
```

```

50 {
51     return fIStream.good();
52 }
53
54 // Kiểm tra xem tệp có đang mở hay không
55 bool iVigenereStream::is_open() const
56 {
57     return fIStream.is_open();
58 }
59
60 // Kiểm tra xem đã đến cuối tệp chưa
61 bool iVigenereStream::eof() const
62 {
63     return fIStream.eof();
64 }
65
66 // Toán tử nhập liệu để lấy ký tự từ tệp và mã hóa nó
67 iVigenereStream& iVigenereStream::operator>>(char& aCharacter)
68 {
69     char lCharacter = 0;
70
71     if (fIStream.get(lCharacter))
72     {
73         aCharacter = fCipher(fCipherProvider, lCharacter);
74     }
75
76     return *this;
77 }
78

```

Output:



```

Microsoft Visual Studio Debug
Decoding "sample_3.txt" using "Relations".

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?

Completed.
D:\0Study\0C30008 Data Structures_And_Patterns\MidTerm\Bai Lam\MidTermProject\x64\Debug\MidTermProject.exe (process 6888) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

```


Problem 4 – VigenereForwardIterator.cpp

...ermProject\MidTermProject\VigenereForwardIterator.cpp

1

```
1 #include "VigenereForwardIterator.h"
2
3 using namespace std;
4
5 // Constructor cho lớp VigenereForwardIterator, khởi tạo iterator với luồng
   iVigenereStream
6 VigenereForwardIterator::VigenereForwardIterator(iVigenereStream&
   aIStream) :
7     fIStream
   (aIStream),
8     fEOF
   (false)
9 {
10     +
   // Tăng iterator để chuẩn bị cho ký tự đầu tiên
11 }
12
13 // Toán tử dereference, trả về ký tự hiện tại
14 char VigenereForwardIterator::operator*() const
15 {
16     return
   fCurrentChar;
17 }
18
19 // Toán tử tiền tố tăng, cập nhật ký tự hiện tại từ luồng
20 VigenereForwardIterator& VigenereForwardIterator::operator++()
21 {
22     fIStream >>
   fCurrentChar;
23     // Đọc ký
   tự mới từ luồng tệp
24     fEOF = fIStream.eof
   ();
25     // Cập nhật cờ
   EOF dựa trên trạng thái luồng
26     return
   *this;
27 }
28
29 // Toán tử hậu tố tăng, trả về giá trị trước khi tăng
30 VigenereForwardIterator VigenereForwardIterator::operator++(int)
31 {
32     VigenereForwardIterator temp =
   *this;
   // Lưu trữ bản sao hiện tại
```

```

...ermProject\MidTermProject\VigenereForwardIterator.cpp 2
33
34     +                                     ↗
        +(*this);                                     ↗
        // Tăng iterator để cập nhật ký tự
35
36     return                                     ↗
        temp;                                         // ↗
        Trả về bản sao trước khi tăng
37 }
38
39 // So sánh hai iterator để kiểm tra tính tương đương
40 bool VigenereForwardIterator::operator==(const VigenereForwardIterator& ↗
    aOther) const
41 {
42     return &fIStream == &aOther.fIStream && fEOF == ↗
        aOther.fEOF;                                // So sánh địa chỉ luồng và cờ EOF
43 }
44
45 // So sánh hai iterator để kiểm tra tính không tương đương
46 bool VigenereForwardIterator::operator!=(const VigenereForwardIterator& ↗
    aOther) const
47 {
48     return !(*this == ↗
        aOther);                                     // Sử dụng toán ↗
        tử == để kiểm tra không tương đương
49 }
50
51 // Phương thức bắt đầu iterator, đặt lại trạng thái
52 VigenereForwardIterator VigenereForwardIterator::begin() const
53 {
54     VigenereForwardIterator Result = ↗
        *this;                                       // Tạo bản sao của iterator
55
56     Result.fIStream.reset ↗
        ();                                         // Đặt lại luồng ↗
        để bắt đầu từ đầu
57     Result.fEOF = ↗
        false;                                     // Đặt cờ ↗
        EOF về false
58
59     return + ↗
        +Result;                                   // ↗
        Tăng và trả về iterator bắt đầu
60 }
61
62 // Phương thức kết thúc iterator, đánh dấu vị trí kết thúc
63 VigenereForwardIterator VigenereForwardIterator::end() const
64 {
65     VigenereForwardIterator Result = ↗

```

```

        *this; // Tạo bản sao của iterator
66
67     Result.fEOF =
        true; // Đặt cờ EOF về true
68
69     return
        Result; //
        Trả về iterator kết thúc
70 }
71

```

Output:

```

Microsoft Visual Studio Debug Console
Forward Iterator Encoding "sample_4.txt" using "Relations".

SHF J
MLTBX J

Oiwrrs. O Xcdtsx.
Tbuxl Yqtohsmywy.

Sajxhxmywy. Zdr lx mbj owcqnk in tra rcxudtizfx
Vtgt uftfcvbw xbybwa qt madf fzf qe Swad;
Tir prk izp dzsvpe npfw xdpv'w oxph ajm zspit
Wc ozm szxu kcmwr sk yxg dvtnc hozntg.
Spk cmw xzv pglxk gdpff oxqa xxxhsorico fuyplqh;
Jno twvcaxs vjok cdgl yq xbk ujbhbutik;
Bjf ngjfh tavjngx uvbzhwy gh pmsff riyivsyk;
Nhw ilnokyjg eejhiy oa qfdlommzjp rjzclkn.
Senn-odkfz'k oje djmi zqlplp'e bql jgxhspmm yogwh;
Lzi ruo, - xbmuser dc vcnhyjrv hngwsx bhlgek,
Vt jkclmx mbt ygdsa iz njtgage pjbjskgmufw, -
Dj utqwax gqremo nf t zjqa'e bqonnfv
Sh wpf ycxuckcwe qehjorcb hv m qvmn.
Pbi O, utvn fo iic mpbj'e scg mwsjmbt lkyrcm,
Wkw gsmh eh qdmeu es ubrgjmm mcpxxfv-bemhh;
T, cvha mn knmytz munek'h, fzc puog bsws'w jbxyeiu
Yh fubzm vdzky s rupyuc scijba cmncj;
U, cvhh zz scanijf'e bi ozct ommg vlwyhxurgd,
Uwuplym dt imfuowi qe jntmzegnhp gtyhtq,
Msmcsg'y, msjntxku'h, nstc pnywvx rm nkrk
Xfyb sbrf gasuibqwx ygwmx, aloyzt zmpy pbvh zx,
Bzj giey ne moqytz ugs gosfegjcrvprf
Mpji rpie guzl ux bl, tu J zlmm od uanu;
Xts, K, hh bmvp edjt dcxrgd yjxu xv yshwn,
Ufkk dt ehqqhdy hd tfeb heus ix f mbnb,
Vmfmxm cb xzs yd laushb na sam tyc
Xwr xmxvvbu cr qngh pkh sssxfrxqm:
Esj izrayepfi, mqsuj O rsgxh swxyf o zdpuj,
Ik joftnyscc yvyxw zbrf qmqe-xvptwk imkl,
Y px isxfqfrorh yi ujbes f zcabsxj,
Foz qmff cvf whtf jqyixdvqx er yixnw itny.
Jttmx lbux Y xpcw, qozzlfjxrb reijyadil,
Fz ryzfzgz rqshcsnnsi, dxrmqv, jsw pagfyl,
Uc wil rz okiaajl Pkcjthkj sqp uzj yls
Wr xmfoum ojhi yii amm bafcvxi hmf vuqsk:
Uhl nu Eqsy Frajxo ks hx xovw epi knml
Vx J oq cpunmx, obylz, chp yxfmvmffsok,
Yila isn naicuw Gmzkmcxq goatwqm ej pql'w zb,
Zdthy x tmeqjtxs, xxarx afbr, qpjw O
Hg Dwqisx'z wknvp uzi fcmwkyz xuvdp gw.
Iaom, utawllco, iiec ov us lsji: vygr Bepmqvhw hryfk.

Gfrykte, lhqv sug: bzhi bieor qpjm nwnsw acjke
Lufx dvjlw oxph vpdL Zmqrz?

Completed.

D:\0Study\0C30008 Data Structures_And_Patterns\MidTerm\Bài Làm\MidTermProject\x64\Debug\MidTermProject.exe (process 36908) ex
ited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console wh
en debugging stops.

```