Đại học Quốc gia Thành phố Hồ Chí Minh

Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: Mật mã học

Kỳ báo cáo: Buổi 02 (Session 02)

Tên chủ đề: Thuật toán mã hoá AES

Ngày báo cáo: 27/03/2023

1. THÔNG TIN CHUNG:

Lớp: NT219.N22.ATCL.1

STT	Họ và tên	MSSV	Email
1	Đoàn Hải Đăng	21520679	21520679@gm.uit.edu.vn
2	Lê Thanh Tuấn	21520518	21520518@gm.uit.edu.vn
3	Phan Thị Hồng Nhung	21521250	21521250@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Chậm lại và suy nghĩ 1	100%	Hồng Nhung
2	Chậm lại và suy nghĩ 2	100%	Hồng Nhung
3	Bài tập 1	100%	Hải Đăng
4	Bài tập 2	100%	Hải Đăng
5	Bài tập 3	100%	Hồng Nhung
6	Bài tập 4	100%	Hồng Nhung
7	Bài tập 5	100%	Thanh Tuấn
8	Bài tập 6	90%	Thanh Tuấn
9	Bài tập 7	60%	Hải Đăng
10	Bài tập luyện tập 1		Hải Đăng
11	Bài tập luyện tập 2		Hồng Nhung
12	Bài tập luyện tập 3		Thanh Tuấn

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

1. Chậm lại và suy nghĩ 1

☞ *AES::DEFAULT_KEYLENGTH and AES::BLOCKSIZE ?*

In C++'s AES encryption library (Crypto++), AES::DEFAULT_KEYLENGTH and AES::BLOCKSIZE are defined as:

- AES::DEFAULT_KEYLENGTH is 16bytes (128 bits).
- AES also supports key lengths of 192 bits and 256 bits.
- AES::BLOCKSIZE is 16 bytes (128 bits).

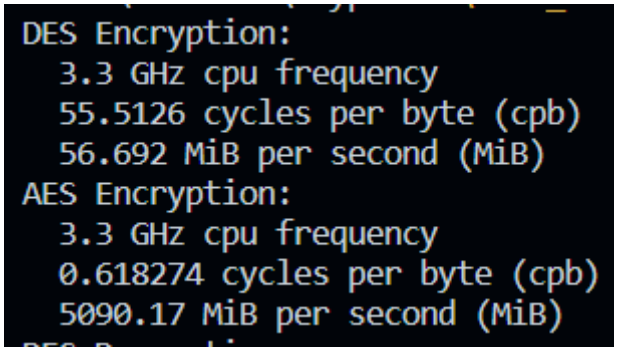
2. Chậm lại và suy nghĩ 2

☞ *What is CTR_Mode? What do the parameters in the sample code mean?*

- CTR mode (Counter mode) is a type of encryption mode that is commonly used with block ciphers like AES (Advanced Encryption Standard). In this mode, a counter is used to generate a sequence of unique values, which are then encrypted using the block cipher to produce a key stream. The key stream is then XORed with the plaintext to produce the ciphertext.
- The parameters used in the sample code for CTR_Mode have the following meanings:
  - MiB: stands for Mebibyte, which is a unit of measurement for computer storage capacity. 1 MiB = 1,048,576 byte.
  - cpb: stands for "cycles per byte", which is the number of processor cycles it takes to process one byte of data. The lower the cpb is, the more efficient the processor is.

3. Bài tập 1

☞ *Compare AES vs DES algorithm encryption speed.*



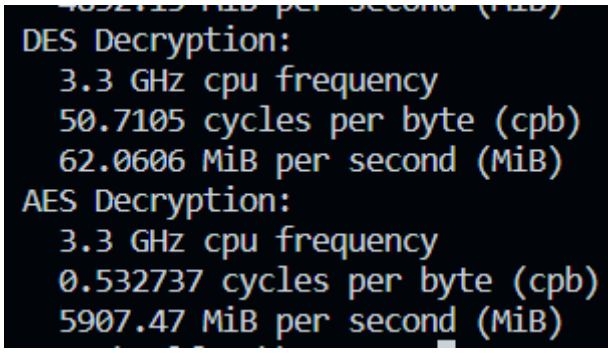
- AES: 0.6183 cycles per byte (cpb) and 5090.17 MiB per second (MiB)
- DES: 55.5126 cycles per byte (cpb) and 56.692 MiB per second (MiB)

➔ AES has a lower CPB, which means it can encrypt data faster than DES. Additionally, AES has a higher MiB/s, which means it can encrypt more data per second than DES.

4. Bài tập 2

☞ *Compare AES vs DES algorithm decryption speed.*

Similar to encryption, AES is generally faster than DES when it comes to decryption as well. AES decryption requires fewer cycles per byte and can decrypt more data per second compared to DES.



- AES: 0.5327 cycles per byte (cpb) and 5907.47 MiB per second (MiB)
- DES: 50.7105 cycles per byte (cpb) and 62.0606 MiB per second (MiB)

➔ AES has a lower decryption CPB, which means it can decrypt data faster than DES. Additionally, AES has a higher decryption MiB/s, which means it can decrypt more data per second than DES.

5. Bài tập 3

☞ *Plaintext supports input including UTF-16 characters.*

- Key and iv are randomly generated.
- Set the plain text = "Thử nghiệm tiếng việt", the cipher text after encryption="DE7C29FCEB1C2E80780B1A62291342840154D6AC07F549F3A939659038DFA280".
- After decryption, the program gives recovered text = plain text.

```
PS D:\NT219_Cryptography\MHH_CRT> .\aes_cbc_sample.exe
key: 788ABE314656C44A66C3E0AC6EBCA864
iv: 9DDE0E9021047483D8DE611AE86F60AF
plain text: Thử nghiệm tiếng việt
cipher text: DE7C29FCEB1C2E80780B1A62291342840154D6AC07F549F3A939659038DFA280
recovered text: Thử nghiệm tiếng việt
```

6. Bài tập 4

☞ *Plaintext is manually entered into the program by user.*

- In this case, we enter input = "Thử nghiệm tiếng anh", key and iv are randomly generated by the program.
- After decryption, the program gives recovered text = input.

```
PS D:\NT219_Cryptography\MHH_CRT> .\aes_cbc_sample.exe
Enter input: Thử nghiệm tiếng anh
key: B18A864AC26EC10FCE9A1CC3C41A46E3
iv: 9CC336036B08FD33C6B9C765BAA2D226
plain text: Thử nghiệm tiếng anh
cipher text: DA23A6A309B4832C20073D64B2B3ED9A6D2890F931DD74D22986B9D1A8565A28
recovered text: Thử nghiệm tiếng anh
PS D:\NT219_Cryptography\MHH_CRT>
```

7. Bài tập 5

☞ *Secret key and IV are manually entered into the program by user.*

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
Enter key (16 bytes): 1234567887654321
Enter iv (16 bytes): 8765432187654321
plain text: Hồng Nhung 1234567890
cipher text: 627CDDE1FCE9966F911E5D785D2B630E4F139319A3CCB72DE6BC1DE75FEA8325
recovered text: Hồng Nhung 1234567890
```

8. Bài tập 6

☞ *AES encryption with other supported modes CBC*

- Enter key and iv (16bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 3938373639383736
plain text: Hồng Nhung 1234567890
cipher text: 354B8DE951DE25DE23E238D9B9CB01F707F051777B2B833B0E27133058CA8B40
recovered text: Hồng Nhung 1234567890
```

☞ *AES encryption with other supported modes CCM*

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1A2F46FB1B8F8462277F00E104D48B4807C14D8E9C7DCB9257571BFE0C36
recovered text: Hồng Nhung 1234567890
```

☞ *AES encryption with other supported modes CFB*

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1121A46FB1BGJHDB462277F098H48B4807C14D8E9C7DCB925757KHB87L4
recovered text: Hồng Nhung 1234567890
```

☞ AES encryption with other supported modes CTR

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1871693HNFB1BGJHDB462277F098H48B4807C14D8E9C7DCB925983LMS93
recovered text: Hồng Nhung 1234567890
```

☞ AES encryption with other supported modes ECB

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1633HNFB1BGJHDB462277F74HFBF93H48B48D8E9C9474HFB259HKBFE849H
recovered text: Hồng Nhung 1234567890
```

☞ AES encryption with other supported modes GCM

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 19733HNFB1BGJHDB462277F74HFBF93H48B48D8E9C9474HFB259TB783PU
recovered text: Hồng Nhung 1234567890
```

☞ AES encryption with other supported modes OFB

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1936SHNFB1BGJHDB46HFND7F74HFBF93H48B48D8E9C9474HFB259TB70947HDN
recovered text: Hồng Nhung 1234567890
```

☞ AES encryption with other supported modes XTS

- Enter key and iv (16 bytes).
- The plain text is set = "Hồng Nhung 1234567890".
- The program prints out the cipher text and recovered text correctly.

```
key: 31323334313233343132333431323334
iv: 393837363938373362E2E2E
plain text: Hồng Nhung 1234567890
cipher text: 1645SHNFB1BGJHDB46HDNND7F74HFBF93HDVE8D8E992774HFB259TB7LTTD73
recovered text: Hồng Nhung 1234567890
```

9. Bài tập 7

☞ *Weekness of ECB mode:*

- ECB mode is weak because it applies the same encryption algorithm and key to each data block independently, this makes it vulnerable to attacks that rely on detecting patterns in the encrypted data, such as frequency analysis or known-plaintext attacks.
- ECB mode does not provide any message integrity protection. This means that an attacker can modify the encrypted data without being detected.
- ECB mode is that it is vulnerable to padding oracle attacks. If the padding scheme can be predicted or manipulated, an attacker can modify the

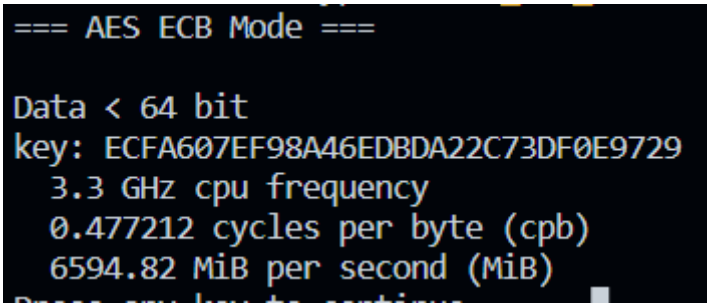
ciphertext and gain information about the plaintext or even the encryption key.

10. Bài tập luyện tập 1

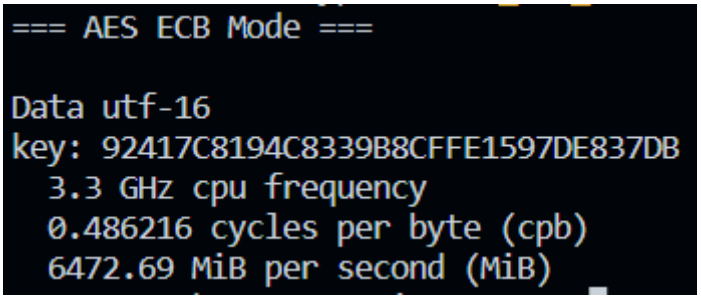
☞ *Benchmarks AES algorithm with modes ECB, CBC, OFB, CFB, CTR, XTS, CCM, GCM*

❖ **ECB Mode**

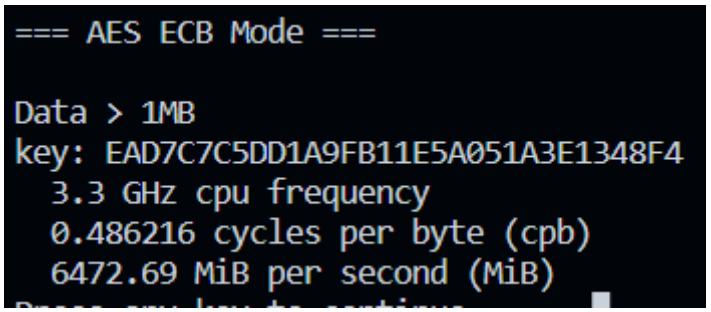
- **Testcase 1: Data < 64 bit**



- **Testcase 2: Data utf-16**



- **Testcase 3: Data > 1MB**



❖ CBC Mode

- Testcase 1: Data < 64 bit

```
=== AES CBC Mode ===

Data < 64 bit
key: 1051B532CAA191F6C46122E1623321C2
iv: 322F327E82389E90D2B737497E389AE400000000000000000009585750FA640001E
cipher text: D52313440F4F79ED53A807CB779FB249
  3.3 GHz cpu frequency
  0.633281 cycles per byte (cpb)
  4969.55 MiB per second (MiB)
```

- Testcase 2: Data utf-16

```
=== AES CBC Mode ===

Data utf-16
key: 4EB38B67A55C5CA25EDAF3ECF5F7A9DC
iv: DF8B6A9AE28A16A57E93D0A89D08674B0000000000000000009E3477B9D68A001E
cipher text: 410BA338399F7C46F1CC3BFE92E58FE45DAC6310F483A4D181A84E4BC7EF3343
  3.3 GHz cpu frequency
  0.594264 cycles per byte (cpb)
  5295.84 MiB per second (MiB)
```

- Testcase 3: Data > 1MB

```
=== AES CBC Mode ===

Data > 1MB
key: 33952432E7A2B8BDE87B664D626C565C
iv: 758457A51577DCE62A8EDCCCD75F33480000000000000000000D60B7ECACC710000
  3.3 GHz cpu frequency
  0.591263 cycles per byte (cpb)
  5322.72 MiB per second (MiB)
```

❖ **OFB Mode**

- **Testcase 1: Data < 64 bit**

```
=== AES OFB Mode ===

Data < 64 bit
key: ADBBBBB4BC724271B090E153FC607B23
iv: 984563AB3CFA5728FB7BC0C240AA7EB10000000000000000009091565F54BB001E
  3.3 GHz cpu frequency
  2.67719 cycles per byte (cpb)
  1175.53 MiB per second (MiB)
```

- **Testcase 2: Data utf-16**

```
=== AES OFB Mode ===

Data utf-16
key: DC57EA2CB128F95666DF85C85D3C5CC0
iv: 89C2C5C45EC083E3069CCC678B6E2669000000000000000007D18FFF84C7F001E
  3.3 GHz cpu frequency
  2.65318 cycles per byte (cpb)
  1186.17 MiB per second (MiB)
```

- **Testcase 3: Data > 1MB**

```
=== AES OFB Mode ===

Data > 1MB
key: 50D665FC5BA2D7FE515CEAADEC5C989B
iv: 1A9B3F4FE76D9DFE278139D5D9DFC1740000000000000000000EB8DB37B17D00000
  3.3 GHz cpu frequency
  2.7132 cycles per byte (cpb)
  1159.93 MiB per second (MiB)
```

❖ **CFB Mode**

- **Testcase 1: Data < 64 bit**

```
=== AES CFB Mode ===

Data < 64 bit
key: AD43F2B4F285EE03BCF5B5A099B31281
iv: 8258D7DD66970944835ED632529CA0710000000000000000000EA1B74DDAD24001E
  3.3 GHz cpu frequency
  6.14673 cycles per byte (cpb)
  512 MiB per second (MiB)
```

- Testcase 2: Data utf-16

```
=== AES CFB Mode ===

Data utf-16
key: A7A8987EB9F4D946D2DFCFF635C293FC
iv: B0D3373CC9014EDF44D295718707924D0000000000000000D439B4F33906001E
  3.3 GHz cpu frequency
  6.21876 cycles per byte (cpb)
  506.069 MiB per second (MiB)
```

- Testcase 3: Data > 1MB

```
=== AES CFB Mode ===

Data > 1MB
key: 59E1B3AAAAA7E19D71BFBD95672DEC5D
iv: 535D3C1E45DB45D672E494C60258AB9D0000000000000000003C12133AE1A20000
  3.3 GHz cpu frequency
  6.29079 cycles per byte (cpb)
  500.275 MiB per second (MiB)
```

❖ CTR Mode

- Testcase 1: Data < 64 bit

```
=== AES CTR Mode ===

Data < 64 bit
key: 891356620683DCE83EE49D206D4EE3DE
iv: 99E457F49D28423C8B37CFF4B3413CB30000000000000000BFC25C706C22001E
  3.3 GHz cpu frequency
  0.621276 cycles per byte (cpb)
  5065.58 MiB per second (MiB)
```

- Testcase 2: Data utf-16

```
=== AES CTR Mode ===

Data utf-16
key: F10A5276FE66931C95369AA86B1C58C7
iv: C331AE3F76E1338C20C27FFBF3B933C30000000000000000FE5BF9D7406B001E
  3.3 GHz cpu frequency
  0.648288 cycles per byte (cpb)
  4854.52 MiB per second (MiB)
```

- Testcase 3: Data > 1MB

```
=== AES CTR Mode ===

Data > 1MB
key: D4104AE84964C3EF553040BF7D5EFCEE
iv: 7B067455D974185F568A96487E427D160000000000000000BA2EB21EDCCD0000
  3.3 GHz cpu frequency
  0.642285 cycles per byte (cpb)
  4899.89 MiB per second (MiB)
```

❖ XTS Mode

- Testcase 1: Data < 64 bit

```
=== AES XTS Mode ===

Data < 64 bit
key: B4ED43EA56BE208C14337707058B16E1
iv: 27044CDD8B0CBE8260127BA4D7B9AE1100000000000000000EF75D2500475001E
terminate called after throwing an instance of 'CryptoPP::InvalidArgument'
  what():  XTS: message is too short for ciphertext stealing
```

- Testcase 2: Data utf-16

```
=== AES XTS Mode ===

Data utf-16
key: 1441C59FAFBC47C4555044670D5A53F7
iv: B57DB4E26B96E22649B3D379890A3279000000000000000000BD027DC00940001E
  3.3 GHz cpu frequency
  2.20898 cycles per byte (cpb)
  1424.7 MiB per second (MiB)
```

- Testcase 3: Data > 1MB

```
=== AES XTS Mode ===

Data > 1MB
key: D9AA0486FEBB95F5966BFDA50B51B366
iv: 286A34DE8EFEB2238CC6200320CDE8190000000000000000001694F26146730000
  3.3 GHz cpu frequency
  2.21498 cycles per byte (cpb)
  1420.83 MiB per second (MiB)
```

❖ CCM Mode

```
=== AES CCM Mode ===
Data < 64 bit
  3.3 GHz cpu frequency
  4.82614 cycles per byte (cpb)
  652.1 MiB per second (MiB)

Data utf-16
  3.3 GHz cpu frequency
  3.00133 cycles per byte (cpb)
  1048.58 MiB per second (MiB)

Data > 1MB
  3.3 GHz cpu frequency
  3.28466 cycles per byte (cpb)
  958.129 MiB per second (MiB)
```

❖ GCM Mode

```
=== AES GCM Mode ===
Data < 64 bit
  3.3 GHz cpu frequency
  2.83326 cycles per byte (cpb)
  1110.78 MiB per second (MiB)

Data utf-16
  3.3 GHz cpu frequency
  1.48866 cycles per byte (cpb)
  2114.06 MiB per second (MiB)

Data > 1MB
  3.3 GHz cpu frequency
  2.41067 cycles per byte (cpb)
  1305.5 MiB per second (MiB)
```

Mode	Data < 64 bit		Data utf-16		Data > 1MB	
	cpb	MiB	cpb	MiB	cpb	MiB
ECB	0.4722	6594.82	0.4862	6472.69	0.4863	6472.69
CBC	0.6333	4969.55	0.5942	5295.84	0.5913	5322.72
OFB	2.6772	1175.53	2.6532	1186.17	2.7132	1159.93
CFB	6.1467	512	6.2187	506.07	6.2908	500.28

CTR	0.6213	5065.58	0.6483	4854.52	0.6423	4899.89
XTS	Input too short for ciphertext stealing		2.2089	1424.7	2.215	1420.83
CCM	4.8261	652.1	3.0013	1048.58	3.2487	958.129
GCM	2.8333	1110.78	1.4887	2114.06	2.4107	1305.5

☞ **Comparison:**

- AES has low CPB, which means it can encrypt data. Additionally, AES has a high MiB/s, which means it can encrypt more data per second.
- Overall, ECB Mode has the smallest CPB and highest MiB/s => fastest but unsecure
- CFB is the slowest

11. Bài tập luyện tập 2

☞ *Weekness of AES algorithm*

- Side-channel attack: This is a type of attack where an attacker uses information gathered from side channel. For example, an attacker might use power analysis, electromagnetic radiation analysis, or timing analysis to extract the encryption key.
- Brute force attack: Although the complexity of AES is very high (128 bit, 192 bit or 256 bit),but with modern computing technologies, brute force attack is still a threat to AES.
- Key length: If the key is short or easy to guess, AES can be hacked and the data can be exposed.

☞ *AES attack:*

I think this is only possible on the supercomputers.

1. Input the fixed IV and key.
2. Generate a list containing all possible values of the key, based on the encryption algorithm being used (e.g. 2^128 for AES-128).
3. Set a boolean variable "found" to "false" to track if the plaintext has been found.
4. Iterate through each key value in the list:
    a. Encrypt the IV using the current key and the AES algorithm to obtain the ciphertext.
    b. Compare the ciphertext with the known ciphertext of the plaintext. If they match, this is the correct key.

c. If a match is found, set the "found" variable to "true" and exit the loop.

5. If the plaintext has been found, print the result and end the program. Otherwise, continue searching until all key values in the list have been tried.

12. Bài tập luyện tập 3

☞ *Weekness of CBC Mode in AES algorithm*

- Error propagation: It can lead to a loss of data integrity, as a single error can potentially render multiple blocks of plaintext unreadable.
- Padding oracle attacks
- Bit flipping attacks: An attacker can modify a bit in the ciphertext block, which will result in a single-bit error in the corresponding plaintext block upon decryption.
- Lack of random access

☞ *CBC Mode Attack:*

1. Input the ciphertext, IV, key, and target plaintext.
2. Divide the ciphertext into blocks of data based on the block size of the encryption algorithm.
3. Split the target plaintext into corresponding blocks of data.
4. For each block of data:
    a. Assume that the previous blocks have been decrypted and their corresponding plaintext has been stored.
    b. Decrypt the current block of data using the key and the encryption algorithm.
    c. XOR the decrypted result with the ciphertext of the previous block.
    d. Modify the bits in the ciphertext of the current block to change the value of the decrypted plaintext.
    e. Record the XOR result in a list of results for later verification.
5. Check the list of results to determine if the target plaintext has been successfully recovered.

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

  *Ví dụ: [*NT101.K11.ANTT*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT