MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

CNG242 Programming Language Concepts – Spring 2021 – Assignment 3 – C++

Date handed out: 31 April 2021, Monday
Submission due: 8 June 2021, Tuesday 23:55

Please Read This Page Carefully

Submission Rules

1. **You need to write a comment on the first line of your file**, stating that you read the rules specified here and the submission is your own work. **Submissions without this statement will not be graded. Example,** "/* Zekican Budin – 1234567
   I read and accept the submission rules and the extra rules specified in each question. This is my own work that is done by myself only */".
2. As explained in the syllabus[1] provided for CNG 242, attempting any academic dishonesty[2], breaking professionalism and ethics[3] rules may result in the following:
   - You might be asked to perform an oral test to confirm your submission.
   - You may receive a "zero" grade for this submission.
   - **You may receive "zero" from all future submissions.**
   - You may receive a failing letter grade for the course, and this case might be forwarded to the discipline committee.
3. You **cannot use** someone 'else's code. You **cannot hire** someone to write the code for you. **You cannot share your code with someone else.**
4. You need to be ready to demonstrate your own work, answer related questions, and have short coding sessions if necessary.
5. You need to compress your .cpp and .h files and submit a single rar or zip file **named with your student id only**. Your compressed file should not contain any .exe or any project related 3rd extensions. Only .cpp and .h files will be evaluated. For example, **1234567.rar or 1234567.zip**
6. Late submission within the first 24 hours is accepted with %20 penalty. After the first 24 hours, any late submission will not be accepted or graded.
7. You cannot share this worksheet with any third parties. Upon doing so, any detected action will directly be sent to the disciplinary committee. The assignment **should not be shared publicly in any manner, at any time.** The assignment cannot be disclosed or disseminated to anyone before, during, or after the submission.
8. You should read the questions fully and follow the directions listed. Only the **functions, operators, classes and/or structures with the same name will be graded.**
9. You can only get full marks, if your files fully compile, run, and generate correct output for all possible cases. **Non-general static solutions will not be graded!**
10. **If you are added to another section as a guest student**, you will still have to submit your submission under the main section you are registered. You can check your registered section by trying to see your grades in ODTUClass. Only submit to the section that you see your grades.
11. Your final rar/zip file should not contain more than 3 files. **Implement your functions, overload operators in your header files.** You can create an extra helper header.
12. You can only use <iostream>, std::ostream, std::istream, std::cout, std::cin, std::endl, std::strcpy, std::strlen, std::strcpy_s, and std::copy. **To make strcpy and strlen work, you can include <cstring>, however, do not use any other function from cstring** library! **Do not use following** keywords; asm, auto, export, goto, mutable, register, signed, unsigned, union, volatile, wchar_t, dynamic_cast, static_cast.

[1] Page 3&4 (Course rules, #1,2,3)
[2] Taking unfair advantage in assessment is considered a serious offence by the university, which will take action against any student who contravenes the regulation through negligence or deliberate intent.
[3] For a comprehensive cheating definition, please refer to: https://ncc.metu.edu.tr/res/academic-code-of-ethics . When a breach of the code of ethics occurs (cheating, plagiarism, deception, etc.), the student will be added to the BLACKLIST.

CNG 242 Programming Language Concepts – Assignment 3 - C++

Operator Overloading, IPv6

In this work we are going to establish some classes and operators which can be used for operations on IPv6 addressing.

Learning Outcomes: On successful completion of this assignment, a student will:

- Write a C++ program that utilizes operator overloading.
- Appreciate the ability to program using notation closer to the target domain.
- Establish a class that can be used for arithmetic operations on binary numbers as well as IPv6.

Some parts of this assignment will be imaginary. There are no practical implementations like these for IPv6, but they are useful for practising your operator overloading skills.

Internet Protocol version 6 (IPv6) is the state of the art version of the Internet Protocol (IP), which can provide identification and location system for computers on networks and routes traffic across the Internet. IPv6 is expected to replace IPv4 eventually. It uses a **128**-bit address, theoretically allowing 2^{128} addresses. IPv6 addresses are represented as eight groups, separated by colons, of four hexadecimal digits.

Example:

1080:0000:0000:0000:0008:0800:200C:417A

The 128 bits of an IPv6 address have

- 3-bit Format Prefix (FP): Used to identify the type of address
- 13-bits Top-Level Aggregation Identifier (TLA ID): Used to identify the authority responsible for the address at the highest level of the routing hierarchy.
- 8-bits Reserved field (Res): Reserved for the future
- 24-bits Next-Level Aggregation Identifier (NLA ID): Used to identify internet service providers (ISPs).
- 16-bits Site Level Aggregation Identifier (SLA ID): Used by an individual organization to create its own local addressing hierarchy and to identify subnets.
- 64- bits Interface ID: The Interface ID field is used to identify individual interfaces on a link.

as illustrated in figure 1.

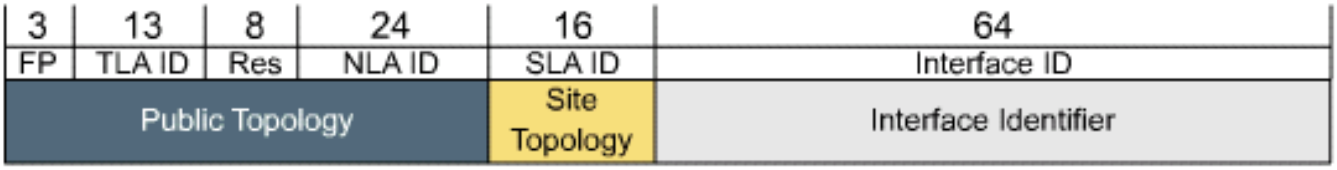


Figure 1: IPv6 format

For this assignment, you are going to establish an "Address" class to represent IPv6 addresses. The Address class will have private data members for the hexadecimal and binary representations separately. We are going to assume that when the FP field of an IPv6 address is 001, it is a unicast address. If the FP field is 000, it is an anycast address. Apart from these two classifications, if the decimal representation of the FP field is greater than one but smaller than or equal to four, it is an anycast address. If it is greater than four, it is a multicast address. The FP dependent classification can be summarized in Table 1 as follows:

FP field	Class
000	Anycast
001	Unicast
1< FP <=4	Anycast
4< FP	Multicast

Table 1: Classification Details

Write the following operators in Table 2 for your Address class:

ID	Operator	Operation
1	~	This unary operator will print the result of classification for the address object (anycast, unicast, multicast).
2	!	This unary operator will print the TLA ID both in binary and hexadecimal.
3	++	This unary operator will print the NLA ID both in binary and hexadecimal.
4	&	This unary operator will print the SLA ID both in binary and hexadecimal.
5	+	Perform **bitwise logical disjunction**(or) operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
6	*	Perform **bitwise logical conjunction**(and) operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
7	-	Perform bitwise **NOR** operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
8	/	Perform bitwise **NAND** operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
9	%	Perform bitwise **XOR** operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
10	^	Perform bitwise **XNOR** operation between the **interface identifier field** of two IPv6 addresses and return the result to another address object.
11	=	Copy the contents of one address object to another one.
12	==	Compare two address objects, return true if they are the same. Otherwise, return false. It will also print the sections which are same (FP, TLA, NLA, SLA, Interface)
13	!=	Compare two address objects, return false if they are the same. Otherwise, return true. It will also print the sections which are not same (FP, TLA, NLA, SLA, Interface)
14	<<	Print out an address object both in hexadecimal and binary format (separate lines)
15	>>	Read an address from the user (read in hexadecimal and convert to binary.

Table 2: Operators

For operations with ID: 5 to 10, FP, TLA, Res, NLA, SLA fields should be the same with the address object on the left (Please check sample runs).
Additionally, your program should also include a basic user interface to use all of these options. Please check the sample run.

Grading Policy:

Item	Mark (out of 20)
Address Class	**3**
Operator ID 1	**1**
Operator ID 2	**1**
Operator ID 3	**1**
Operator ID 4	**1**
Operator ID 5	**1**
Operator ID 6	**1**
Operator ID 7	**1**
Operator ID 8	**1**
Operator ID 9	**1**
Operator ID 10	**1**
Operator ID 11	**1**
Operator ID 12	**1**
Operator ID 13	**1**
Operator ID 14	**1**
Operator ID 15	**1**
User Interface	**2**

Submission Instructions:

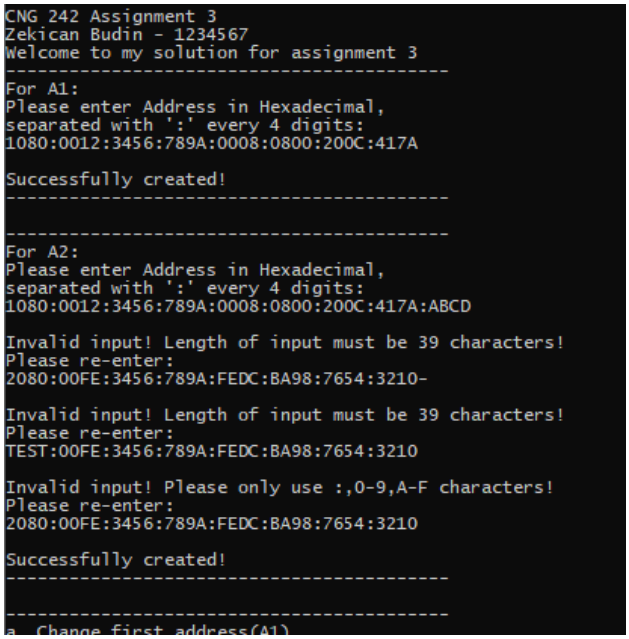
- Implement your functions/Overload operators in your **header files.**
- **Both parties of the cheating get zero from all of the assignments.**
- **Code modularity, style, and redundancies will also be graded.**
- **Try to replicate the sample run as much as possible. Cleary show your work and input/output formats if you change anything.**
- Function names, operators must be the same as the provided information.
- If you are using visual studio, you can submit the project without .vs folder as shown on May 31, 2021 lab first 15 minutes. Otherwise, please submit only the .h and .cpp files.

Sample Run:

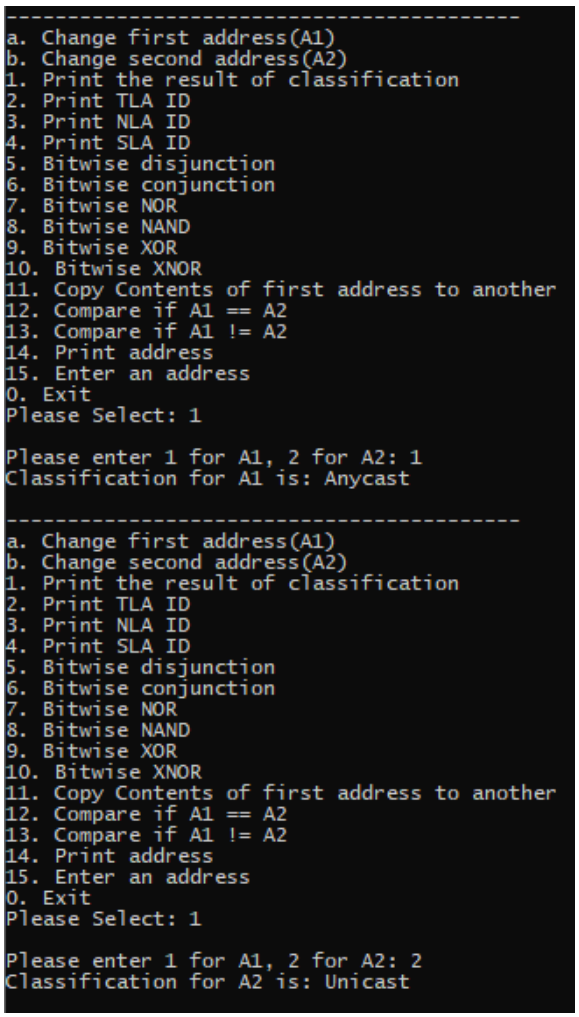
This is a single sample run, the screenshots cut in parts and linked with a table of contents for easier access.

Contents

Initial



Operator ID 1



Operator ID 2, 3, 4

```
----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 2

Please enter 1 for A1, 2 for A2: 1
TLA ID of A1
Hexadecimal: 1080
Binary: 1 0000 1000 0000

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 3

Please enter 1 for A1, 2 for A2: 2
NLA ID of A2
Hexadecimal: FE3456
Binary: 1111 1110 0011 0100 0101 0110

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 4

Please enter 1 for A1, 2 for A2: 1
SLA ID of A1
Hexadecimal: 789A
Binary: 0111 1000 1001 1010

----------------------------------------
```

Operator ID 5 and 6

```
----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 5

Result of A1 + A2;

Hexadecimal
1080:0012:3456:789A:FEDC:BA98:765C:737A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
1111 1110 1101 1100:1011 1010 1001 1000:0111 0110 0101 1100:0111 0011 0111 1010

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 6

Result of A1 * A2;

Hexadecimal
1080:0012:3456:789A:0008:0800:2004:0010

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0000 0000 1000:0000 1000 0000 0000:0010 0000 0000 0100:0000 0000 0001 0000
```

Operator ID 7 and 8

```
----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 7

Result of A1 - A2;

Hexadecimal
1080:0012:3456:789A:0123:4567:89A3:8C85

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0001 0010 0011:0100 0101 0110 0111:1000 1001 1010 0011:1000 1100 1000 0101

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 8

Result of A1 / A2;

Hexadecimal
1080:0012:3456:789A:FFF7:F7FF:DFFB:FFEF

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
1111 1111 1111 0111:1111 0111 1111 1111:1101 1111 1111 1011:1111 1111 1110 1111
```

Operator ID 9 and 10

```
----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 9

Result of A1 % A2;

Hexadecimal
1080:0012:3456:789A:FED4:B298:5678:736A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
1111 1110 1101 0100:1011 0010 1001 1000:0101 0110 0111 1000:0111 0011 0110 1010

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 10

Result of A1 ^ A2;

Hexadecimal
1080:0012:3456:789A:012B:4D67:A987:8C95

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0001 0010 1011:0100 1101 0110 0111:1010 1001 1000 0111:1000 1100 1001 0101

----------------------------------------
```

Operator ID 11

```
-----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 11

Creating a new object (A3)!
A3 is created!

Contents of A3:

Hexadecimal
0000:0000:0000:0000:0000:0000:0000:0000

Binary
FP,TLA,Res,NLA,SLA
0000 0000 0000 0000:0000 0000 0000 0000:0000 0000 0000 0000:0000 0000 0000 0000
Interface Identifier
0000 0000 0000 0000:0000 0000 0000 0000:0000 0000 0000 0000:0000 0000 0000 0000

Contents of A1:

Hexadecimal
1080:0012:3456:789A:0008:0800:2000C:417A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0000 0000 1000:0000 1000 0000 0000:0010 0000 0000 1100:0100 0001 0111 1010

After A3 = A1;
Contents of A3:

Hexadecimal
1080:0012:3456:789A:0008:0800:2000C:417A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0000 0000 1000:0000 1000 0000 0000:0010 0000 0000 1100:0100 0001 0111 1010

Contents of A1:

Hexadecimal
1080:0012:3456:789A:0008:0800:2000C:417A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0000 0000 1000:0000 1000 0000 0000:0010 0000 0000 1100:0100 0001 0111 1010

-----------------------------------------
```

Operator ID 12 and ID 13

```
----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 12

Result of A1==A2: false
Same sections:
SLA

----------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 13

Result of A1!=A2: true
Different sections:
FP
TLA
NLA
Interface

----------------------------------------
```

Operator ID 14, 15, and Exit

```
------------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 14

Please enter 1 for A1, 2 for A2: 1

Hexadecimal
1080:0012:3456:789A:0008:0800:2000C:417A

Binary
FP,TLA,Res,NLA,SLA
0001 0000 1000 0000:0000 0000 0001 0010:0011 0100 0101 0110:0111 1000 1001 1010
Interface Identifier
0000 0000 0000 1000:0000 1000 0000 0000:0010 0000 0000 1100:0100 0001 0111 1010

------------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 15

Please enter 1 for A1, 2 for A2: 2
Please enter Address in Hexadecimal,
separated with ':' every 4 digits:
A242:B213:C140:D111:E315:F334:0336:1350

Successfully created!

------------------------------------------
a. Change first address(A1)
b. Change second address(A2)
1. Print the result of classification
2. Print TLA ID
3. Print NLA ID
4. Print SLA ID
5. Bitwise disjunction
6. Bitwise conjunction
7. Bitwise NOR
8. Bitwise NAND
9. Bitwise XOR
10. Bitwise XNOR
11. Copy Contents of first address to another
12. Compare if A1 == A2
13. Compare if A1 != A2
14. Print address
15. Enter an address
0. Exit
Please Select: 0


Goodbye!
```