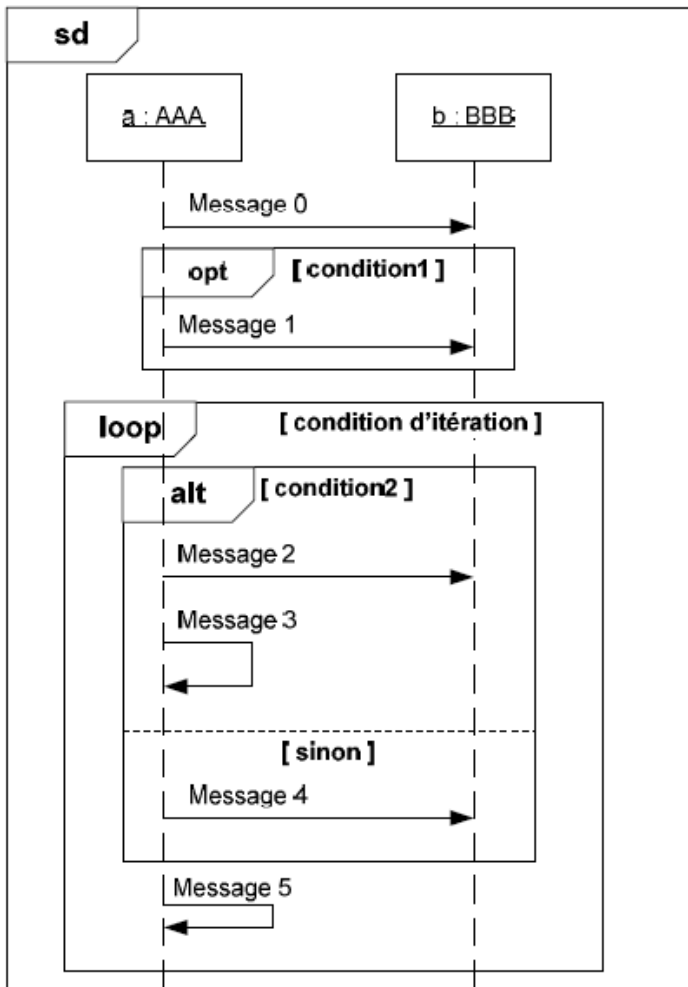


Ecrire l'équivalent en C++

[illegible]

- L'option (opt) correspond en pseudo code à l'instruction « Si condition ALORS ... FINSI »
- L'alternative (alt) correspond en pseudo code à l'instruction « Si condition ALORS ... SINON ... FINSI »
- La boucle (loop) correspond en pseudo code à l'instruction « TANT QUE condition ALORS ... FINTANTQUE »

Exercice 2

L'I.P.C. choisi utilise les sockets. On s'intéresse ici à l'encapsulation des services socket dans la classe `Socket` décrite ci-dessous. Cette classe sert de fondation à des classes comme `SocketClient`, `SocketServeur`, `SocketClientFtp`, etc. .

```

1.  #ifndef __SOCKET_H__
2.  #define __SOCKET_H__

3.  #include <WinSock2.h>
4.  #include <string>

5.  typedef std::string String;

    // Superclasse
6.  class Socket
7.  {
8.  public:

9.      enum TYPE { STREAM=SOCK_STREAM, DATAGRAM=SOCK_DGRAM };

10.     Socket(const Socket&);
11.     virtual ~CSocket();

12.     Socket& operator=(const Socket& sock);

13.     String  receiveLine();
14.     String  receiveBytes();
15.     int     receive(void* buf, unsigned size);

16.     void    close();

17.     void    sendLine (String s);
18.     void    sendBytes(const String& s);
19.     void    sendBytes(const unsigned char* buff);
20.     void    send      (const void* buff, unsigned n);

21. protected:
22.     friend class SocketServer;

23.     Socket(SOCKET s);
24.     Socket();

25.     SOCKET s_;                // le socket de la winsock
26.     int* refCounter_;

27. private:
28.     static void start();        // initialise la librairie des sockets
29.     static void end();         // termine l'utilisation des winsock
30.     static int  nofSockets_;    // nombre de sockets
31. };

32. #endif //__SOCKET_H__

```

Q-9.1

Le fichier de déclaration de la classe comporte des lignes précédées du caractère #.
Donner la signification de ce caractère.

.....

Q-9.2

Justifier les lignes 1, 2 et 32.

.....

Q-9.3

La ligne 4 inclut le fichier **string** de la stl. Que signifie le sigle stl ?

.....

Q-9.4

Que déclare la ligne 5.

.....

Q-9.5

A quelle ligne est déclaré un constructeur de copie.

.....

Q-9.6

Quel est l'impact sur la portée des attributs et des méthodes de la section **protected** déclarée à la ligne 21.

.....

Q-9.7

A la ligne 22, la classe **SocketServer** est déclarée amie de la classe **Socket**.
A quelle(s) proposition(s) correspond(ent) cette déclaration ?

Proposition	Vrai	Faux
Les membres <i>private</i> de la classe Socket sont visibles depuis la classe SocketServer		
Les membres <i>private</i> de la classe SocketServer sont visibles depuis la classe Socket		

Q-9.8

Les déclarations ci-dessous de la classe **ClientIntranet** utilisent un objet de la classe **Socket**.

Propositions
<pre>class ClientIntranet : public Socket { public: ClientIntranet(/*...*/); ... };</pre>
<pre>class ClientIntranet { Socket s; public: ClientIntranet(/*...*/); ... };</pre>

En vous aidant de la déclaration des constructeurs de la classe **Socket**, quelle(s) proposition(s) est (sont) correcte(s) ? Cocher les réponses correctes dans le tableau

Propositions	Code correct	Code faux
<pre>class ClientIntranet : public Socket { public: ClientIntranet(/*...*/); ... };</pre>		
<pre>class ClientIntranet { Socket s; public: ClientIntranet(/*...*/); ... };</pre>		

Q-9.10

La classe **ClientSocket** construit un socket permettant à un client de communiquer vers un serveur.

Le constructeur reçoit comme paramètre formel :

- l'adresse IP du serveur spécifiée sous la forme d'un pointeur constant vers un objet de type **String**.
- un numéro de ports.

Déclarer la classe **ClientSocket**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....