A) Programme et module :

Introduction :

À ce stade les notions de programme et module s'impose. Nous allons répondre aux questions suivantes :
Qu'est-ce qu'un programme ?
A quoi sert un programme ?
Qu'est-ce qu'un module ?

Qu'est-ce qu'un programme ?

Un programme est le cœur d'un programme EZ dans lequel on peut faire un traitement sur les données, les affichés, les lire en entrée etc..

Déclaration d'un programme :

Pour indiquer qu'un fichier EZ est un programme, il faut le déclarer comme suit :

program [program_name]

[…]

procedure [program_name]

[instructions..]

end procedure

Comment passés des arguments en ligne de commande ?
Il existe deux méthodes pour passé des arguments comme paramètre :
1- Envoie de valeurs en ligne de commandes :
Commande :
./program_name.exe --var1=arg1 --var2=arg2
2- Envoie d'un fichier de données en ligne de commandes :
Pour l'envoie d'un fichier data, il faut passer le fichier comme option lors de la compilation.
Commande :
ezc program_name.ez -o program_name.exe -d [data_file_path]
./program_name.exe

Comment récupérer la valeurs des arguments passés comme paramètre ?

La récupération des valeurs se fait automatiquement à partir de la ligne de commande.
Pour les arguments passés en paramètre la syntaxe est la suivante :

```
program hello_arg

arguments
x is integer as « --x »
c is string as « --c »
end arguments
global max_i is integer = 5
procedure showData (x is integer, c is string)
```

```
print x, « fois », c
end procedure
procedure hello_arg
i is integer =0

for i in 0.. max_i
do
showData(x,c)
end for

end procedure
```

Contraintes à respecter :

Les arguments passés doivent être uniquement :

- Entiers,
- Réels,
- Chaînes de caractères (string).

A quoi sert un programme ?

Un programme sert à utiliser les modules « prototype » déclarés et définir le comportement du programme.

Comment inclure un module dans un programme ?

Pour inclure un module EZ dans un programme il suffit d'écrire la ligne suivante :

```
import person
```

Pour inclure une bibliothèque cpp :

```
import "math.h"
```

Comment inclure du code C++ dans un programme EZ ?

Il est possible d'écrire du code C++ dans les fichiers EZ, il suffit de préciser qu'il s'agit du code C++ pour que le compilateur le prend en compte.
Exemple :

```
cppcode

double sum=0,0 ;

for (int i=0 ; i<10;i++)
     sum+=i*2 ;

end cppcode
```

La question qui se pose en ce moment c'est comment réutiliser la variable sum écrite en cpp ?
Pour réutiliser les variables écrite en cpp il faut préfixé le nom de la variable par un « cpp_[nom de la variable ez] » afin d'éviter des conflits éventuelles. Même chose pour l'utilisation des variables EZ dans un blok cpp « ez_[nom de la variable c++] »

Qu’est-ce qu’un module ?

Un module est programme ou l’on peut définir les fonctions et les classes (prototype) à réutiliser dans le programme principal.

```
program person

class person

name is string
id is integer
age is integer

function isOlderThan(old is integer) return boolean
return a<age
end function

end class
```

Les bibliothèques

Il existe plusieurs types de bibliothèques :
On distingue les modules écrits par l’utilisateur, les bibliothèques c++ qu’on réutilise en tant que module EZ et les bibliothèques c++ externes que l’utilisateur voudrai utiliser.

1- Les modules écrits par l’utilisateur :

Un module est considéré comme étant une bibliothèque EZ dans laquelle on définit le comportement des fonctions et procédures. L’utilisateur peut également faire appelle à des fonctions C++ qu’il faudra prendre en compte
Exemple :

```
function cos(x is integer) return cpp cos(x)
```

Dans ce cas il faut prendre en considération la bibliothèque math du C++.

2- Les bibliothèques c++ :

Les bibliothèques c++ sont déduit à partir du programme directement. Dans le cas où l’utilisateur voudrai utiliser une bibliothèque C++ particulière (non supportée par g++ ou gcc). Dans ce cas l’utilisateur doit définir le nom de la bibliothèque et le chemin vers le fichier comme suit :

```
import "usr/include/math.h"
liberary "/usr/lib/src -lm"
```