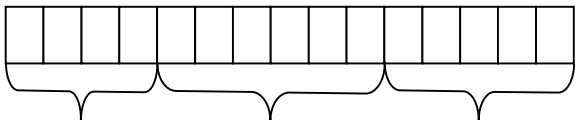
programmation
examen septembre

Afin d'utiliser au mieux un espace mémoire limité (par exemple sur une carte à puce), nous désirons pouvoir définir des zones de bits de tailles variables, et pouvoir remplir ces zones soit directement avec des valeurs binaires, soit avec des valeurs décimales. Une mémoire de 15 bits peut être découpée de la façon suivante :



Zone de 4 bits | Zone de 6 bits | Zone de 5 bits

Pour représenter une zone de bits nous utilisons les classes suivantes :

```
class ZoneException{
      AnsiString mess;
    public:
      ZoneException( AnsiString m)
      :mess(m){}
      AnsiString getMessage()
      { return mess; }
};

class Zone{
   friend class Zones;
   protected :
      int taille;
      bool * rep;
   public :
     Zone( int t);
     virtual ~Zone();
     virtual void remplir(AnsiString s)=0;
 };
```

```
class ZoneDec : public Zone{
   public :
     ZoneDec( int t);
     void remplir (AnsiString s);
};

class ZoneBin : public Zone{
   public :
     ZoneBin( int t);
     void remplir (AnsiString s);
};
```

taille est la taille de la zone exprimée en nombre de bits, *rep* est sa représentation sous forme d'un tableau de booléens (*true* représente le bit 1 et *false* le bit 0). *ZoneDec* est une zone pouvant être remplie à partir d'une valeur exprimée en décimal, *ZoneBin* est une zone pouvant être remplie à partir d'une valeur exprimée en binaire. Le paramètre *s* de la méthode remplir est la valeur en décimal ou en binaire, représentée sous forme de chaîne de caractères.

Question 1.(2 points)

Ecrire la définition des constructeurs et destructeurs des classes *Zone*, *ZoneDec* et *ZoneBin*.

Question 2.(5 points)

Ecrire la définition des méthodes *remplir* des classes *ZoneDec* et *ZoneBin*. Si la valeur donnée en paramètre a une taille trop grande pour entrer dans la zone ou n'est pas du bon type, il faut lever une exception, si la taille est plus petite, il faut la compléter avec des 0 à gauche.

La mémoire est constituée de plusieurs zones, de tailles différentes qui peuvent être remplies avec des valeurs soit binaires soit décimales. Un fichier contient la description de la mémoire : il est constitué d'autant de lignes que de zones mémoire, chaque ligne contient la taille de la zone suivie de 'D' si la zone est remplie avec une valeur décimale, et 'B' si la zone est remplie avec une valeur binaire. Un autre fichier contient la liste des valeurs.

```
class Memoire{
      int nombre;
      int nombreBits;
      Zone ** lesZones ;
      bool * memoire;
   public :
      Memoire ( char * nomFich);
      ~Memoire ();
      void lireLesZones( char * nomFich);
      void creerMemoire();
};
```

Nombre est le nombre de zones, *nombreBits* le nombre de bits de la mémoire. *lesZones* est un tableau de pointeur sur chacune des zones. *memoire* est un tableau de booléens représentant le contenu de la mémoire.

Question 3.(2 points)

Ecrire la définition du destructeur de la classe *Memoire*.

Question 4.(3 points)

Ecrire la définition du constructeur de la classe *Memoire*. Le fichier en paramètre contient la description de la mémoire : taille et type des zones.

Question 5.(4 points)

Ecrire la définition de la méthode *lireLesZones*, qui affecte au différentes zones de la mémoire les valeurs lues dans le fichier de valeurs.

Question 6.(4 points)

Ecrire la définition de la méthode *creerMemoire* qui rempli le tableau *memoire* à partir des différentes valeurs contenues dans les zones.