

Contrôle Continu - MIF16
Durée 45 min
AUCUN DOCUMENT AUTORISÉ

Barême : /1 , /2 , / 3 , /4 , /5 , /5

Question 1

Que représentent les symboles suivants dans un fichier pour **make** :

\$~

Chaîne de caractères composée des noms de toutes les dépendances.

\$<

Contient le nom de la dépendance permettant la génération de la cible.

Question 2

En C++, vous avez la classe **Vector** qui représente un tableau dynamique.
Écrivez la définition de la fonction **swap** :

```
template < class Type >
void Vector<Type>::Swap(
    const unsigned int idx1 ,
    const unsigned int idx2 )
{
    Type oTemp = m_oData[idx1];
    m_oData[idx1] = m_oData[idx2];
    m_oData[idx2] = oTemp;
}
```

Question 3

Proposez une redéfinition complète de la classe **Vector** pour des chaînes de caractères de type **String**. (On ne s'intéressera qu'à la déclaration de la classe, non à son code.)

```
template <
class Vector< std::string >
{
    public:
        void Swap(const unsigned int idx1, const unsigned int idx2);

    private:
        std::string m_oData;
};
```

Question 4

On veut un vecteur qui maintienne ses éléments triés. Pour cela, écrire une classe qui surcharge la classe **Vector** et l'algorithme qui insère un élément au bon endroit :

```
template <class Type>
class SortedVector : public Vector<Type>
{
    public:
        SortedVector() {}
        SortedVector( Vector<Type> )
        {
            Sort();
        }
        Type& operator[] ( const unsigned int idx ) const
        {
            return Vector<Type>::operator[] ( idx );
        }
        void Insert( const Type& oType )
        {
            Vector<Type>::push_back(oType);
            Sort();
        }
    private:
        void Sort( void )
        {
            // Algo de tri...
        }
};
```

Question 5

Donnez 4 outils de gestion de projet, expliquez leur fonctionnement ainsi que leur importance dans un projet de grande envergure. (Ces outils doivent apporter des solutions complémentaires)

- Make ;
- Wiki, Forum, Track, Mailing List, ...
- SVN, CVS, RCS, GIT, ...
- Diagramme de Gant, cahier des charges, compte rendu de réunions, ...
- Coding Rules, test unitaire, framework (multi-plateforme), standard interne, ...
- ...

Question 6

Makefile complet.

```
CC      = g++
LD      = g++

SUBDIR   = Divers Typage Symboles

SRCDIR   = src
INCDIR   = include
OBJDIR   = obj
DEPDIR   = dep
BINDIR   = bin
DOCDIR   = doc

RSRCDIR  = $(SUBDIR:%=$(SRCDIR)/%)
RDEPDIR  = $(SUBDIR:%=$(DEPDIR)/%)
ROBJDIR  = $(SUBDIR:%=$(OBJDIR)/%)

BIN      = $(BINDIR)/pascal-comp

SRCS     = $(foreach DIR,$(SUBDIR),$(wildcard $(SRCDIR)/$(DIR)/*.cpp)) $(wildcard $(SRCDIR)/*.cpp) $(SRCDIR)/pascal-lexer.cpp
          $(SRCDIR)/pascal-parser.cpp

OBJS     = $(SRCS:%$(SRCDIR)/%.cpp=$(OBJDIR)/%.o)
DEPS     = $(SRCS:%$(SRCDIR)/%.cpp=$(DEPDIR)/%.d)

CPPFLAGS = -W -Wall -ansi -pedantic -I$(INCDIR) -I$(SRCDIR)
LDFLAGS  = -lfl

all: $(RDEPDIR) $(ROBJDIR) $(BIN)

$(BIN): $(OBJS)
      $(LD) -o $@ $^ $(LDFLAGS)

$(SRCDIR)/%.cpp: analyse/%.l
      flex -o $@ $<
```

```
$(SRCDIR)/%.cpp: analyse/%.y
    bison --debug --verbose -d -o $@ $<

$(OBJDIR)/%.o: $(SRCDIR)/%.cpp
    $(CC) $(CPPFLAGS) -o $@ -c $<

$(DEPDIR)/%.d: $(SRCDIR)/%.cpp
    $(CC) $(CPPFLAGS) -MM -MD -o $@ $<

$(DEPDIR)/%:
    mkdir $@

$(OBJDIR)/%:
    mkdir $@

--include $(DEPS)

.PHONY: clean distclean doc

doc:
    cd $(DOCDIR) && doxygen

clean:
    rm -rf $(OBJS)

distclean: clean
    rm -rf $(BIN) $(DEPS) doc/html doc/latex
```