0.1 Test n° 1

Descriptif : Le test n° 1 réalise le premier exemple donné par le sujet.
Les fichiers analysés sont :

[h !] file1.cpp [linenos, gobble=2]cpp // affiche le message "Hello world" int main() cout ¡¡"Hello world"¡¡endl ; cout¡¡endl ; return 0 ;

[h !] file1.h [linenos,gobble=2]cpp int main() ;

[h !] key1.txt [linenos,gobble=2]cpp int world template

Résultat attendu : Nous lançons le programme avec le contexte suivant :

$$tp\_stl \ -e \ -k \ key1.txt \ file1.cpp \ file1.h$$

Nous devons obtenir le résultat ci-dessous après avoir trié la sortie : [h !] file1.res [linenos, gobble=2]pascal cout file1.cpp 3 4 endl file1.cpp 3 4 main file1.cpp 2 file1.h 1 return file1.cpp 5

0.2 Test n° 2

Descriptif : Le test n° 2 réalise le deuxième exemple donné par le sujet. Les fichiers analysés sont :

[h !] file2.cpp [linenos, gobble=2]cpp // affiche le message "Hello world" int main() cout ¡¡"Hello world"¡¡endl ; cout¡¡endl ; return 0 ;

[h !] file2.h [linenos,gobble=2]cpp int main() ;

[h !] key2.txt [linenos,gobble=2]cpp int world template

Résultat attendu : Nous lançons le programme avec le contexte suivant après avoir trié la sortie :

$$tp\_stl \ -k \ key2.txt \ file2.cpp \ file2.h$$

Nous devons obtenir le résultat ci-dessous : [h !] file2.res [linenos, gobble=2]pascal int file2.cpp 2 file2.h 1

0.3 Test n° 3

Descriptif : Le test n° 3 réalise le test sur le fichier main de notre programme. Les fichiers analysés sont :

[linenos, gobble=2]cpp //==========
// Name : Ref$_c$*roisee.cpp*//*Author* : //*Version* : //*Copyright* : *Yourcopyrightnotice*//*Description* : *HelloWorldinC++, Ansi−style*// ==========
include ¡iostream¿ include ¡vector¿

include "CmdLine/cmdLine.hpp" include "References/Referenceur.hpp" include "References/References.hpp"

using namespace std ; using namespace Reference$_c$*roisee*;

int main(int argc, char** argv) /**/

CmdLine : :Arguments args ; CmdLine : :Parser parser("Permet de referencer des mots clefs a travers des fichiers") ; parser.addOption("exclude,e", "Inverse le fonctionnement du programme") ; parser.addOption("keyword,k", "Specifie la liste des mots clefs a utiliser", true) ;

try parser.parse(argc, argv, args) ;

catch(exception e) cout ¡¡ "Une erreur c'est produit durant la recuperation de la ligne de commande : " ¡¡ endl ¡¡ e.what() ¡¡ endl ;

//———- // On charge les fichiers a referencer //———- vector¡string¿ ficsReferencer ;

if(args.count("$_a rgs_{"})) $fics Referencer = args.get<vector<string>>("_a rgs_{"});$

else cerr ¡¡ "Aucun fichier a referencer !" ¡¡ endl ; return 1 ;

//———- // On charge les mots clefs si ils sont fournis //———- string fichierMotClef ;

if(args.count("keyword")) fichierMotClef = args.get¡string¿("keyword") ;

//———- // L'etat dans lequel mettre le programme //———- bool mode(args.count("exclude")) ;

References refs ;

//———- // On effectue la reference croisee //———- try Referenceur referenceur(fichierMotClef, mode) ; referenceur.referencer(ficsReferencer, refs) ;

catch(exception e) cerr ¡¡ "Une erreur est survenue durant la reference croisee : " ¡¡ endl ; cerr ¡¡ e.what() ¡¡ endl ;

//———- // On affiche les resultats //———- refs.display(cout) ;

return 0 ; /**/

Résultat attendu : Nous lançons le programme avec le contexte suivant :

$$tp\_stl \ \ file3.cpp$$

Nous devons obtenir le résultat ci-dessous après avoir trié la sortie : [h !] file3.res [linenos, gobble=2]pascal bool file3.cpp 60 catch file3.cpp 30 72 char file3.cpp 19 cout file3.cpp 31 81 else file3.cpp 43 if file3.cpp 40 53 int file3.cpp 19 19 namespace file3.cpp 16 17 return file3.cpp 45 83 true file3.cpp 25 try file3.cpp 27 68 using file3.cpp 16 17

0.4 Test n° 4

Descriptif : Le fichier à analyser est le même que dans le test précédent, seul le contexte d'execution change.

Résultat attendu : Nous lançons le programme avec le contexte suivant :

$$tp\_stl\ \ file4.cpp$$

Nous devons obtenir le résultat ci-dessous après avoir trié la sortie : [h !] file4.res [linenos, gobble=2]cpp addOption file4.cpp 24 25 argc file4.cpp 19 28 args file4.cpp 22 28 40 41 53 54 60 Arguments file4.cpp 22 argv file4.cpp 19 28 cerr file4.cpp 44 73 74 CmdLine file4.cpp 22 23 count file4.cpp 40 53 60 display file4.cpp 81 e file4.cpp 30 32 72 74 endl file4.cpp 31 32 44 73 74 exception file4.cpp 30 72 fichierMotClef file4.cpp 51 54 69 ficsReferencer file4.cpp 38 41 70 get file4.cpp 41 54 main file4.cpp 19 mode file4.cpp 60 69 parse file4.cpp 28 Parser file4.cpp 23 parser file4.cpp 23 24 25 28 $\text{Reference}_c roisee file4.cpp 17 referencer file4.cpp 70 References file4.cpp 63 Referenceur file4.cpp 69 referenceu$