

Projet ASD2

Haithem essaied & Maher toujani



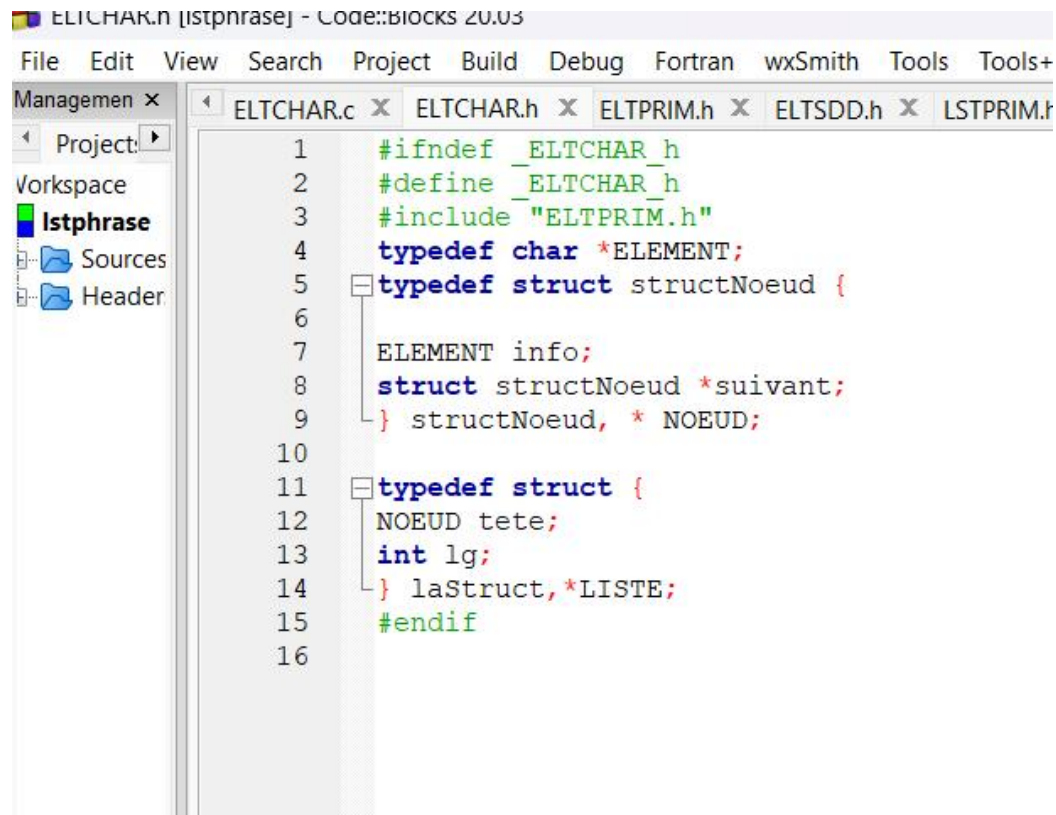
Plan du rapport

1. les structures de donnees utilisees.
2. Les fichiers .c et .h.
3. Les fonctions utilisees.

Les Listes:

Variables: Liste(liste de char contient des phrases), Liste1(liste d'une structure contient les phrases chiffrée)

Type d'element: Liste Char.

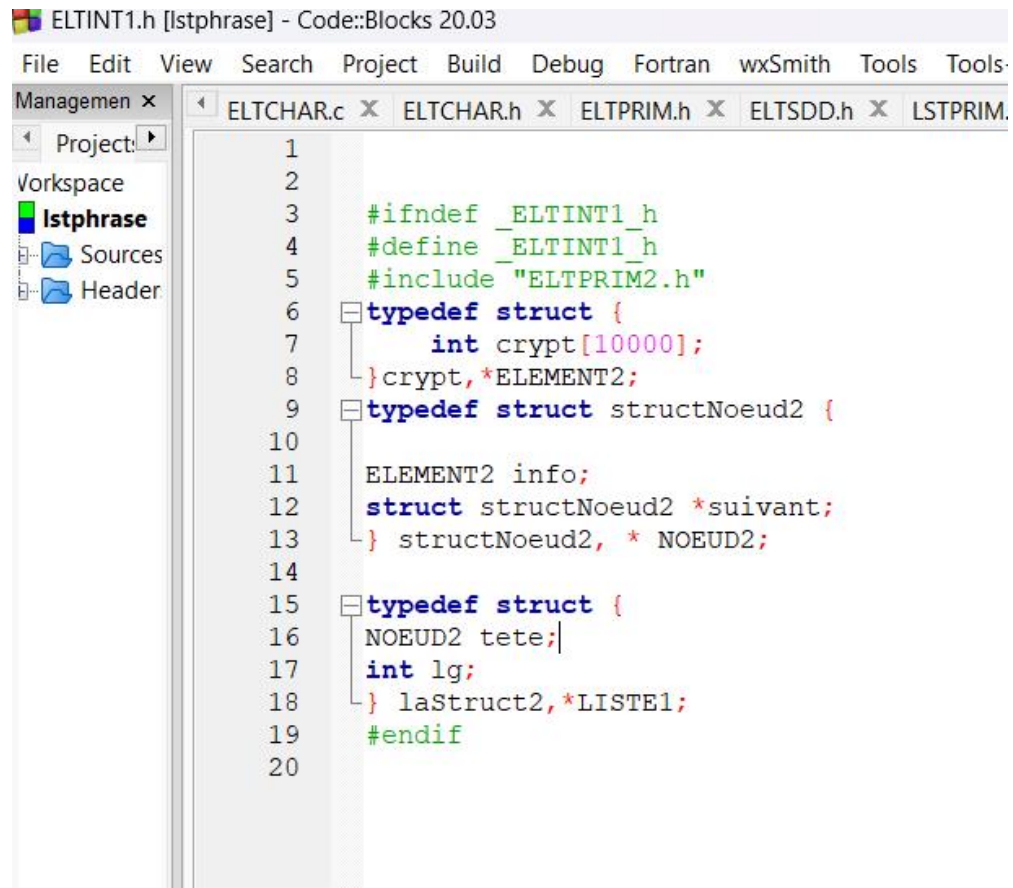


The screenshot shows a code editor window titled "ELTCHAR.h [lstphrase] - Code::Blocks 20.03". The editor displays the following C code:

```
1  #ifndef _ELTCHAR_h
2  #define _ELTCHAR_h
3  #include "ELTPRIM.h"
4  typedef char *ELEMENT;
5  typedef struct structNoeud {
6
7      ELEMENT info;
8      struct structNoeud *suivant;
9  } structNoeud, * NOEUD;
10
11 typedef struct {
12     NOEUD tete;
13     int lg;
14 } laStruct, *LISTE;
15 #endif
16
```

The left sidebar shows a project named "lstphrase" with a "Sources" folder and a "Header" folder. The "Header" folder is selected, and the "ELTCHAR.h" file is open in the editor.

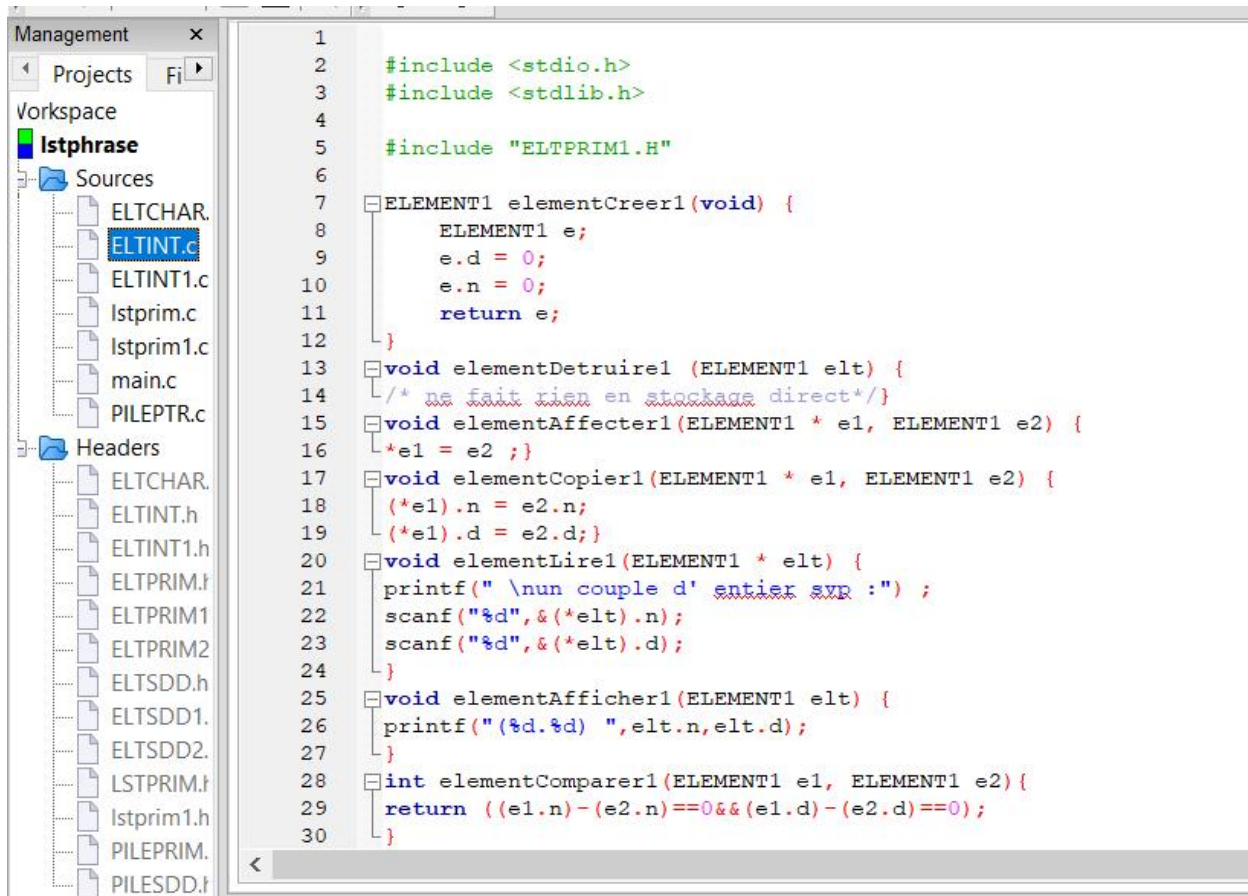
Liste1 structre qui contient a tableau de stockage direct.



```
1
2
3  #ifndef _ELTINT1_h
4  #define _ELTINT1_h
5  #include "ELTPRIM2.h"
6  typedef struct {
7      int crypt[10000];
8  } crypt, *ELEMENT2;
9  typedef struct structNoeud2 {
10
11      ELEMENT2 info;
12      struct structNoeud2 *suivant;
13  } structNoeud2, * NOEUD2;
14
15  typedef struct {
16      NOEUD2 tete;
17      int lg;
18  } laStruct2, *LISTE1;
19  #endif
20
```

Les fichier

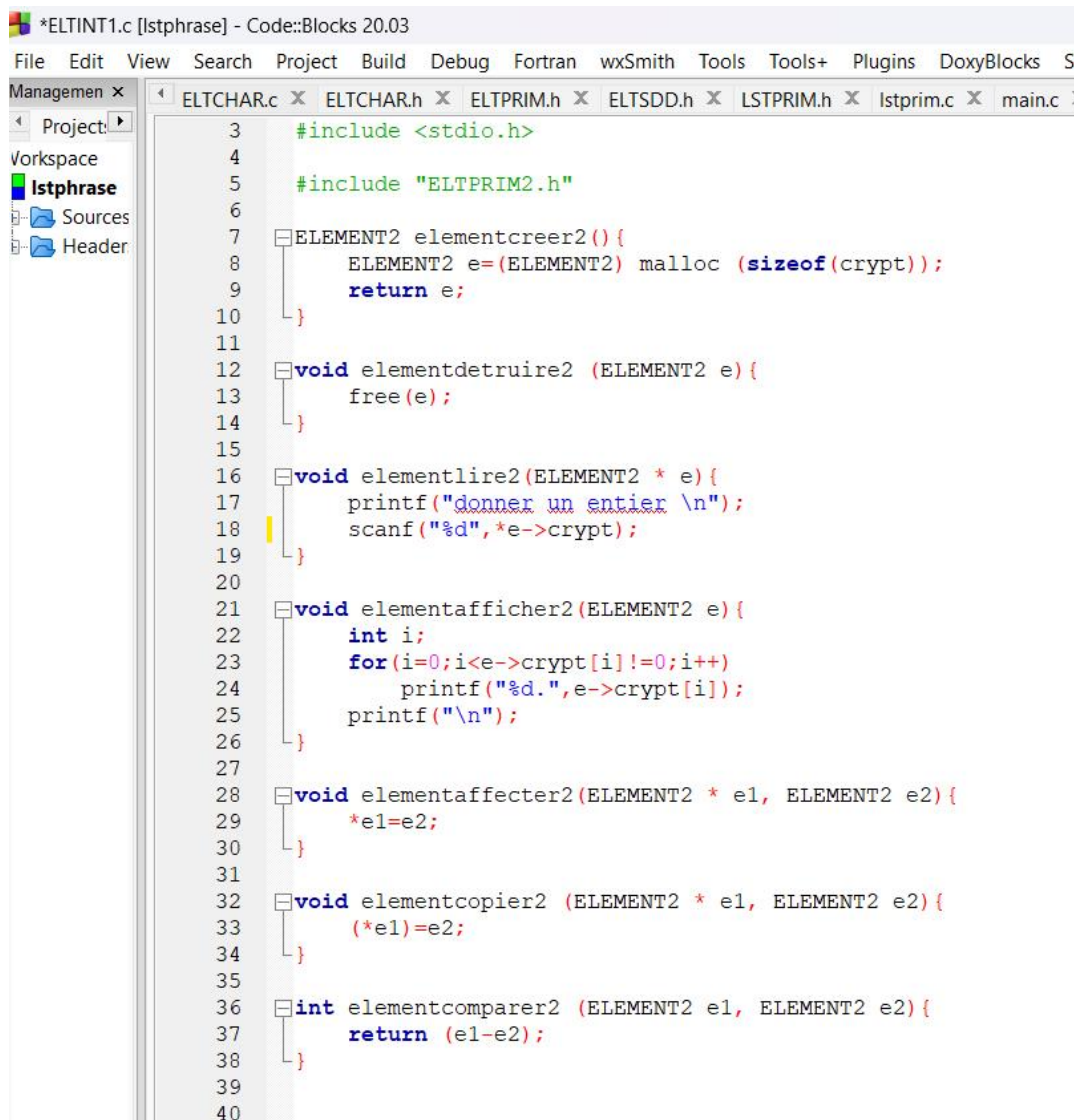
ELTINT.C: Définition des primitives en faisant les changements nécessaires a l'affichage de l'element couple .



```
1
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  #include "ELTPRIM1.H"
6
7  ELEMENT1 elementCreer1(void) {
8      ELEMENT1 e;
9      e.d = 0;
10     e.n = 0;
11     return e;
12 }
13
14 void elementDetruire1 (ELEMENT1 elt) {
15     /* ne fait rien en stockage direct */
16 }
17
18 void elementAffecter1(ELEMENT1 * e1, ELEMENT1 e2) {
19     *e1 = e2;
20 }
21
22 void elementCopier1(ELEMENT1 * e1, ELEMENT1 e2) {
23     (*e1).n = e2.n;
24     (*e1).d = e2.d;
25 }
26
27 void elementLire1(ELEMENT1 * elt) {
28     printf(" \nun couple d' entier svp :") ;
29     scanf("%d",&(*elt).n);
30     scanf("%d",&(*elt).d);
31 }
32
33 void elementAfficher1(ELEMENT1 elt) {
34     printf("(%d.%d) ",elt.n,elt.d);
35 }
36
37 int elementComparer1(ELEMENT1 e1, ELEMENT1 e2){
38     return ((e1.n)-(e2.n)==0&&(e1.d)-(e2.d)==0);
39 }
40
```

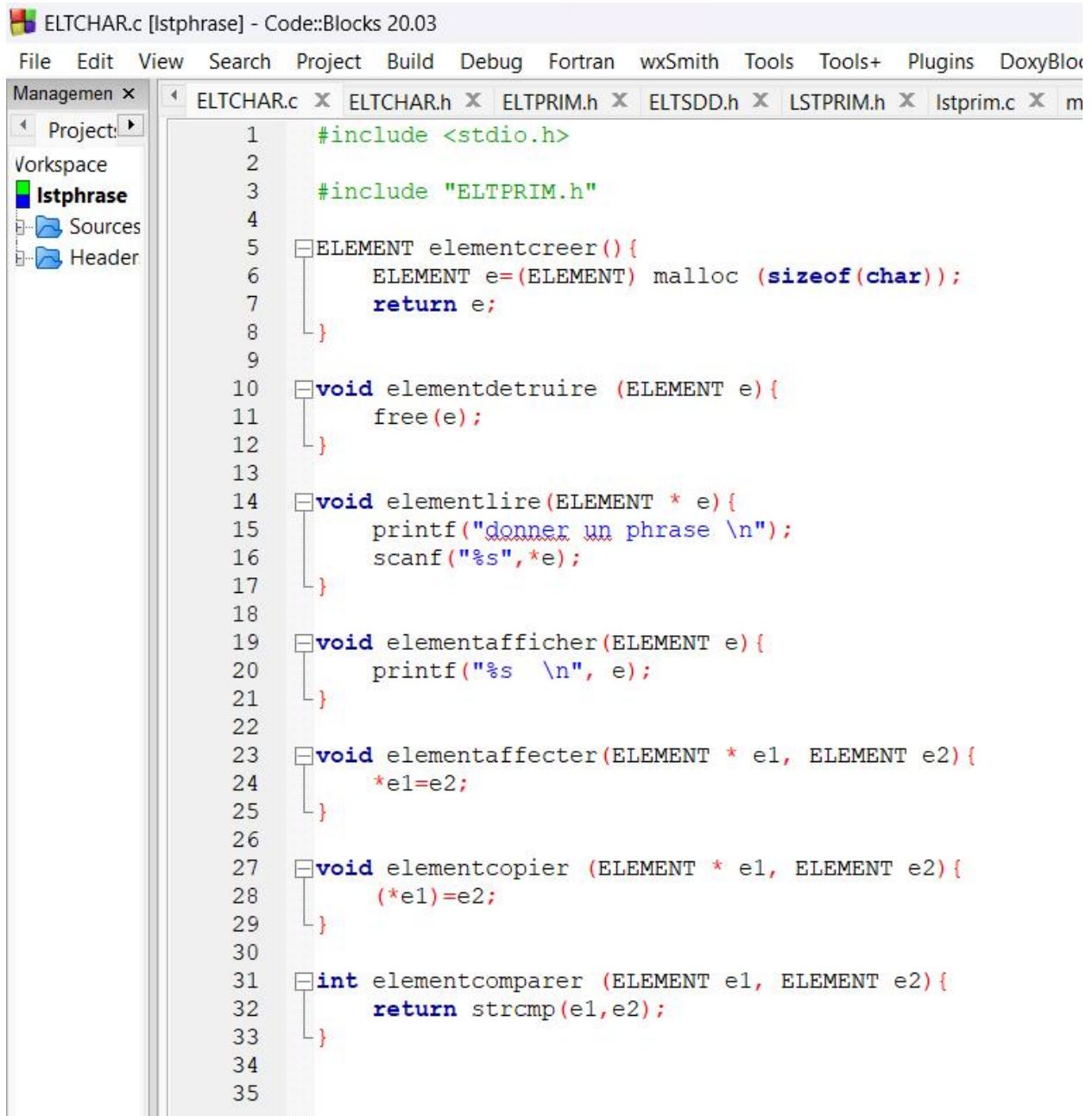
Les fichier

ELTINT1.C: Définition des primitives en faisant les changements nécessaires a l'affichage de l'element2 crypt.



```
3  #include <stdio.h>
4
5  #include "ELTPRIM2.h"
6
7  ELEMENT2 elementcreer2() {
8      ELEMENT2 e=(ELEMENT2) malloc (sizeof(crypt));
9      return e;
10 }
11
12 void elementdetruire2 (ELEMENT2 e) {
13     free(e);
14 }
15
16 void elementlire2(ELEMENT2 * e) {
17     printf("donner un entier \n");
18     scanf("%d",&e->crypt);
19 }
20
21 void elementafficher2(ELEMENT2 e) {
22     int i;
23     for(i=0;i<e->crypt[i]!=0;i++)
24         printf("%d.",e->crypt[i]);
25     printf("\n");
26 }
27
28 void elementaffecter2(ELEMENT2 * e1, ELEMENT2 e2) {
29     *e1=e2;
30 }
31
32 void elementcopier2 (ELEMENT2 * e1, ELEMENT2 e2) {
33     (*e1)=e2;
34 }
35
36 int elementcomparer2 (ELEMENT2 e1, ELEMENT2 e2) {
37     return (e1-e2);
38 }
39
40
```

ELTCHAR.C: Définition des primitives en faisant les changements nécessaires a l'affichage de l'element char qui contient un phrase.



```
1  #include <stdio.h>
2
3  #include "ELTPRIM.h"
4
5  ELEMENT elementcreer(){
6      ELEMENT e=(ELEMENT) malloc (sizeof(char));
7      return e;
8  }
9
10 void elementdetruire (ELEMENT e){
11     free(e);
12 }
13
14 void elementlire(ELEMENT * e){
15     printf("donner un phrase \n");
16     scanf("%s",*e);
17 }
18
19 void elementafficher(ELEMENT e){
20     printf("%s \n", e);
21 }
22
23 void elementaffecter(ELEMENT * e1, ELEMENT e2){
24     *e1=e2;
25 }
26
27 void elementcopier (ELEMENT * e1, ELEMENT e2){
28     (*e1)=e2;
29 }
30
31 int elementcomparer (ELEMENT e1, ELEMENT e2){
32     return strcmp(e1,e2);
33 }
34
35
```

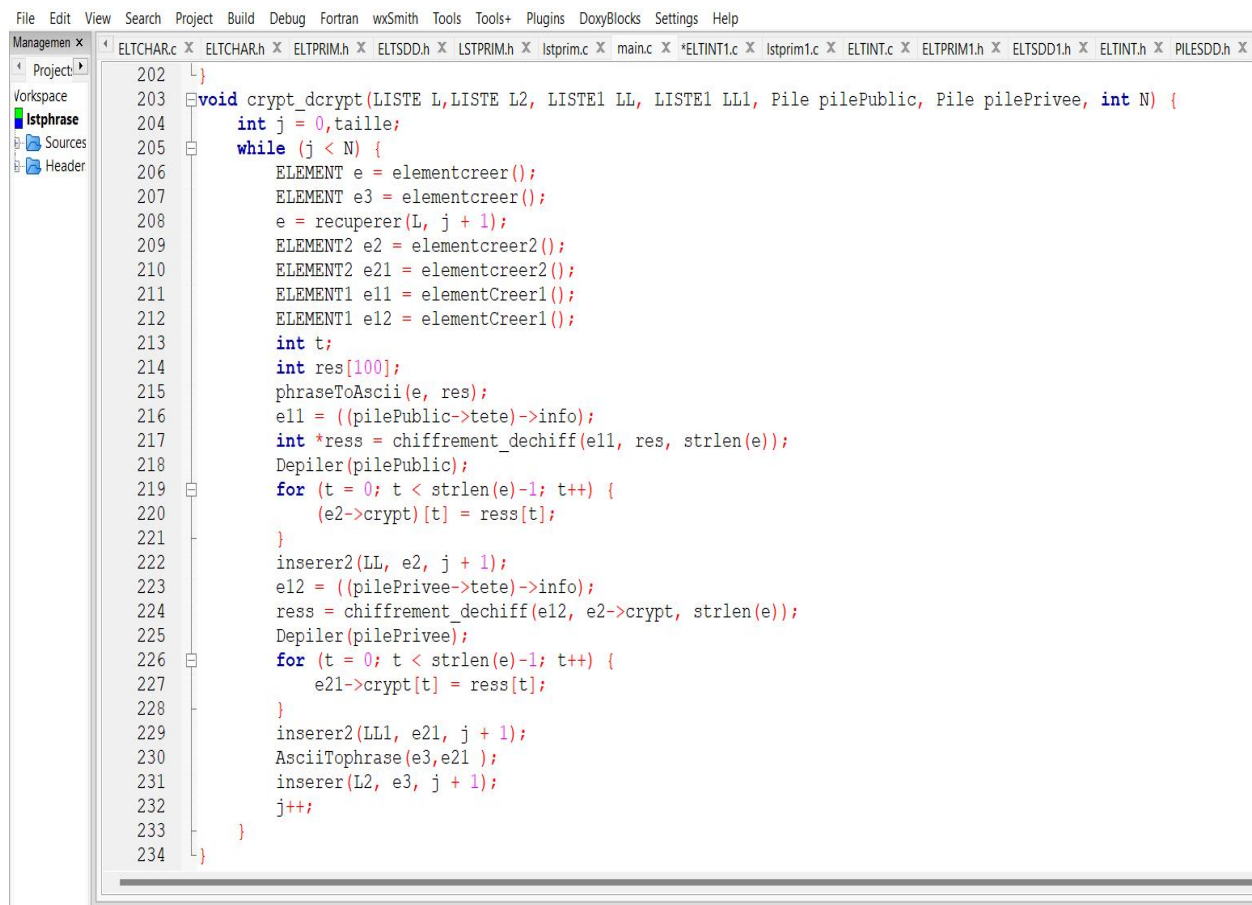

Les fonctions

void **crypt_decrypt**(LISTE L,LISTE L2, LISTE1 LL, LISTE1 LL1, Pile pilePublic, Pile pilePrivee, int N):

Ce fonction est utiliser pour chiffrer une phrase après la methode rsa et le stocker dans la liste1 ,et dechiffré la phrase chiffré a les codes ASCII de la phrases original et le stocker dans une autre listes de type liste1.

RQ: ce fonction utiliser la fonction chiffrement_dechiffement qui chiffrer un tableau d'entier avec la methode rsa.

code :



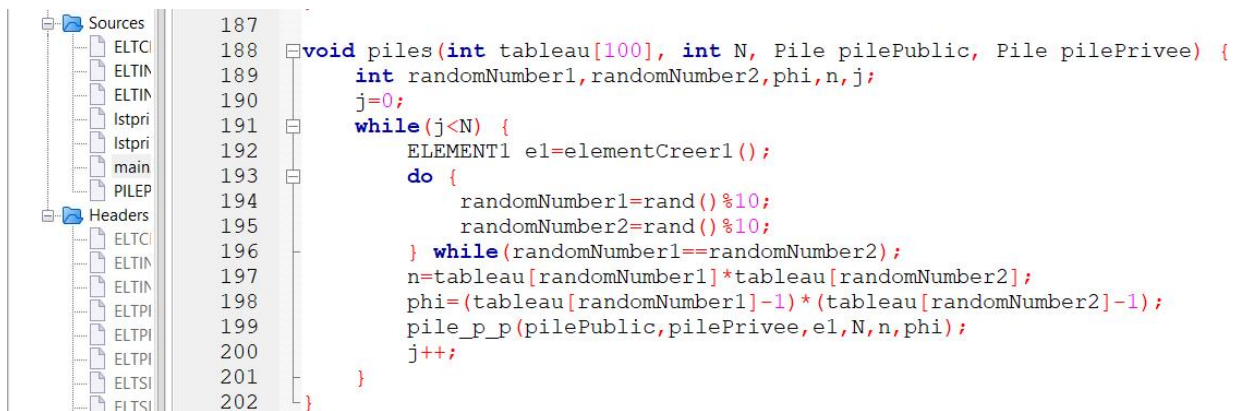
```
202 }
203 void crypt_decrypt(LISTE L,LISTE L2, LISTE1 LL, LISTE1 LL1, Pile pilePublic, Pile pilePrivee, int N) {
204     int j = 0, taille;
205     while (j < N) {
206         ELEMENT e = elementcreer();
207         ELEMENT e3 = elementcreer();
208         e = recuperer(L, j + 1);
209         ELEMENT2 e2 = elementcreer2();
210         ELEMENT2 e21 = elementcreer2();
211         ELEMENT1 e11 = elementCreer1();
212         ELEMENT1 e12 = elementCreer1();
213         int t;
214         int res[100];
215         phraseToAscii(e, res);
216         e11 = ((pilePublic->tete)->info);
217         int *ress = chiffrement_dechiff(e11, res, strlen(e));
218         Depiler(pilePublic);
219         for (t = 0; t < strlen(e)-1; t++) {
220             (e2->crypt)[t] = ress[t];
221         }
222         inserer2(LL, e2, j + 1);
223         e12 = ((pilePrivee->tete)->info);
224         ress = chiffrement_dechiff(e12, e2->crypt, strlen(e));
225         Depiler(pilePrivee);
226         for (t = 0; t < strlen(e)-1; t++) {
227             e21->crypt[t] = ress[t];
228         }
229         inserer2(LL1, e21, j + 1);
230         AsciiTophrase(e3, e21);
231         inserer(L2, e3, j + 1);
232         j++;
233     }
234 }
```

Les fonctions

void **piles**(int tableau[100], int N, Pile pilePublic, Pile pilePrivee) :

Utiliser pour remplir les piles prive et public après le choix des nombres premiers a partir du tableau et le calcul des variables (n,e,d,phi)

code :



```
187 void piles(int tableau[100], int N, Pile pilePublic, Pile pilePrivee) {
188     int randomNumber1, randomNumber2, phi, n, j;
189     j=0;
190     while(j<N) {
191         ELEMENT1 el=elementCreer1();
192         do {
193             randomNumber1=rand()%10;
194             randomNumber2=rand()%10;
195             while(randomNumber1==randomNumber2);
196             n=tableau[randomNumber1]*tableau[randomNumber2];
197             phi=(tableau[randomNumber1]-1)*(tableau[randomNumber2]-1);
198             pile_p_p(pilePublic,pilePrivee,el,N,n,phi);
199             j++;
200         }
201     }
202 }
```