Pratique du C **Bétisier**

Licence Informatique — Université Lille 1 Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 5 — 2013-2014

V-1 (06-12-2012)

#define MAX = 10 ;

Ce bout de code

int t[MAX], x = MAX;

provoque une erreur à

► fait ce qu'il devrait :

#define add (a,b) (a + b)

return add(1,2);

provoque une erreur à

► l'exécution : ► fait ce qu'il devrait :

▶ la compilation :

int main(void){

Ce bout de code

► la compilation : ► l'exécution :

V84 (06-12-2012)

V84 (06-12-2012)

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)

Préprocessing

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)

Pratique du C Bétisier

Règles du jeu

Les bouts de codes se prêtent aux commentaires.

- On cherche à savoir si ces codes provoquent une erreur à
 - ▶ la compilation,
 - l'exécution;
 - ▶ font ce qu'il devrait.

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
#define MAX = 10 ;
int t[MAX], x = MAX;
```

Ce bout de code

- provoque une erreur à
 - ▶ la compilation : oui
 - ► l'exécution :
- ► fait ce qu'il devrait :

Une macro n'est ni une déclaration ni une initialisation mais provoque une substitution textuelle.

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
#define add (a,b) (a + b)
int main(void){
  return add(1,2);
```

Ce bout de code

- provoque une erreur à
 - ▶ la compilation : oui
 - ► l'exécution :
- ► fait ce qu'il devrait :

L'espace est le séparateur entre l'identificateur de macro et la chaîne à substituer.

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)

```
#define CARRE(a) ((a) * (a))
                                                                                                           #define CARRE(a) ((a) * (a))
                 int main(void){
                                                                                                           int main(void){
                    int x = 2;
                                                                                                              int x = 2;
                    return CARRE(x++) ;
                                                                                                              return CARRE(x++) ;
                                                                                                           Ce bout de code
                 Ce bout de code
                   provoque une erreur à
                                                                                                             provoque une erreur à
                         ▶ la compilation :
                                                                                                                   ▶ la compilation : non
                         ▶ l'exécution :
                                                                                                                   ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                             ▶ fait ce qu'il devrait : non
                                                                                                           Même bien constituée une macro n'immunise pas contre l'effet
                                                                                                           latéral.
                                                         www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                   www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                           Pratique du C
Bétisier
                 /* On veut retourner 1 */
                                                                                                           /* On veut retourner 1 */
                 int main(void){
                                                                                                           int main(void){
                    int a=0, b=0, res;
                                                                                                              int a=0, b=0, res;
                                                                                          Lisibilité du code
Lisibilité du code
                    if (a)
                                                                                                              if (a)
                       if (b) res = !b;
                                                                                                                 if (b) res = !b;
                    else res = !a ;
                                                                                                              else res = !a ;
                    return res ;
                                                                                                              return res ;
                 Ce bout de code
                                                                                                           Ce bout de code
                   provoque une erreur à
                                                                                                             provoque une erreur à
                         ▶ la compilation :
                                                                                                                   ▶ la compilation : non
                                                                                                                   l'exécution : oui
                         l'exécution :
                   ► fait ce qu'il devrait :
                                                                                                             ► fait ce qu'il devrait :
                                                                                                           Il est difficile de savoir à quel if se vouer, alors autant utiliser
                                                                                                           des blocs.
                                                         www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
                                                                                                                                                   www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
                                                                                           Pratique du C
Bétisier
                 int main(void){
                                                                                                           int main(void){
Lisibilité du code
                                                                                          Lisibilité du code
                   char c ;
                                                                                                             char c ;
                   while (c = getchar() != EOF) putchar(c) ;
                                                                                                             while (c = getchar() != EOF) putchar(c) ;
                   return 0 ;
                                                                                                             return 0 ;
                 Ce bout de code
                                                                                                           Ce bout de code
                                                                                                             ▶ provoque une erreur à
                   provoque une erreur à
                         ▶ la compilation :
                                                                                                                   ▶ la compilation : non
                         ► l'exécution :
                                                                                                                   ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                             ▶ fait ce qu'il devrait : oui
                                                                                                           Il faut rendre lisible le code plutôt que de compter sur la
                                                                                                           priorité entre les opérateurs.
                                                         www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                   www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
```

```
/* On veut parcourir une cha\^ine de caract\'eres */
                                                                                                         /* On veut parcourir une cha\^ine de caract\'eres */
                main
                                                                                                         main
                (biov)
                                                                                                         (void)
                { int i ;
                                                                                                         { int i ;
Efficacité du code
                                                                                        Efficacité du code
                   char * bibi="La vie est belle" ;
                                                                                                           char * bibi="La vie est belle" ;
                   for(i=0;i<strlen(bibi);i++);</pre>
                                                                                                           for(i=0;i<strlen(bibi);i++);</pre>
                  return 0 :
                                                                                                           return 0 :
                Ce bout de code
                                                                                                         Ce bout de code
                   provoque une erreur à
                                                                                                           provoque une erreur à
                        ▶ la compilation :
                                                                                                                 ▶ la compilation : non
                                                                                                                 ▶ l'exécution : non
                        l'exécution :
                   ▶ fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : oui
                                                                                                         C'est un parcours en n^2!!!!!! À éviter impérativement.
                                                                                                         Utilisez les fonctions de la libraire standard à loisir après
                                                                                                         les avoir comprises (i.e. après votre cours de pratique des
                                                                                                         systèmes).
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                         Pratique du C
Bétisier
                int main(void){
                                                                                                         int main(void){
                   int a=1,b=1;
                                                                                                           int a=1,b=1;
                   if(a=0)
                                                                                                           if(a=0)
                      printf("Attention z\\'ero") ;
                                                                                                              printf("Attention z\\'ero") ;
Syntaxe
                                                                                        Syntaxe
                   else b /= a ;
                                                                                                           else b /= a ;
                                                                                                           return b ;
                  return b :
                Ce bout de code
                                                                                                         Ce bout de code
                   provoque une erreur à
                                                                                                           provoque une erreur à
                        ► la compilation :
                                                                                                                 ▶ la compilation : non
                        ► l'exécution :
                                                                                                                 l'exécution : non
                   ▶ fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : non
                                                                                                         L'affectation est un opérateur et non une instruction.
V84 (06-12-2012)
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                         Pratique du C
Bétisier
                /* on veut retourner 0 */
                                                                                                         /* on veut retourner 0 */
                int main(void){
                                                                                                         int main(void){
                    char a=1,b=0;
                                                                                                            char a=1,b=0;
                    switch(a){
                                                                                                            switch(a){
                      case 1 : a = b;
                                                                                                               case 1 : a = b;
                                                                                                               defau1t : return 1 ;
                       defau1t : return 1 ;
                    return a ;
                                                                                                            return a ;
                Ce bout de code
                                                                                                         Ce bout de code
                   provoque une erreur à
                                                                                                           provoque une erreur à
                        ▶ la compilation :
                                                                                                                 ▶ la compilation : non
                        ► l'exécution :
                                                                                                                 l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : non
                                                                                                         L'existence du goto et l'usage du break induisent cette er-
                                                                                                         reur difficilement détectable (sans l'aide des avertissements du
                                                                                                         compilateur).
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
```

```
int tab[12] = {
                                                                                                        int tab[12] = {
                   1, 2, 3, 4,
                                                                                                          1, 2, 3, 4,
                   5, 6, 7, 8,
                                                                                                           5, 6, 7, 8,
                   9,10,11,12
                                                                                                          9,10,11,12
                int main(void){ /* on veut retourner 3 */
                                                                                                        int main(void){ /* on veut retourner 3 */
                  int i=2,j=2;
                                                                                                          int i=2,j=2;
                  return tab[i++,j++];
                                                                                                         return tab[i++,j++];
                Ce bout de code
                                                                                                       Ce bout de code
                  ▶ provoque une erreur à
                                                                                                          ▶ provoque une erreur à
                        ▶ la compilation :
                                                                                                                ▶ la compilation : non
                        ► l'exécution :
                                                                                                                ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                          ▶ fait ce qu'il devrait : oui
                                                                                                        La virgule est l'opérateur qui délivre comme résultat
                                                                                                        l'opérande droit après avoir évalué l'opérande gauche.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                        Pratique du C
Bétisier
                /* On veut retourner 2 au shell */
                                                                                                        /* On veut retourner 2 au shell */
                int main(void){
                                                                                                        int main(void){
                    return fct(2);
                                                                                                            return fct(2);
Syntaxe
                                                                                       Syntaxe
                int fct(int i){
                                                                                                        int fct(int i){
                    return i ;
                                                                                                            return i ;
                Ce bout de code
                                                                                                       Ce bout de code
                   provoque une erreur à
                                                                                                          provoque une erreur à
                        ▶ la compilation :
                                                                                                               ▶ la compilation : oui
                        l'exécution :
                                                                                                                l'exécution :
                  ► fait ce qu'il devrait :
                                                                                                          ▶ fait ce qu'il devrait :
                                                                                                        La fonction fct devrait être déclarée par un prototype avant
                                                                                                        la fonction principale.
V84 (06-12-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                        Pratique du C
Bétisier
                int pair(int i){
                                                                                                        int pair(int i){
                   if (i)
                                                                                                           if (i)
                      return impair(i-1) ;
                                                                                                              return impair(i-1);
                    return 1;
                                                                                                           return 1;
                int impair (int i){
                                                                                                        int impair (int i){
                    if (i)
                                                                                                           if (i)
                      return pair(i-1);
                                                                                                              return pair(i-1);
                   return 0 ;
                                                                                                           return 0 ;
                                                                                                        Ce bout de code
                Ce bout de code
                  provoque une erreur à
                                                                                                          provoque une erreur à
                        la compilation :
                                                                                                                la compilation : oui
                                                                                                                ► l'exécution :
                        l'exécution :
                   ▶ fait ce qu'il devrait :
                                                                                                          ► fait ce qu'il devrait :
                                                                                                        Encore une fois, un prototype permet de fournir au compila-
                                                                                                        teur la déclaration qui lui manque.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
```

```
typedef int fct_t (int) ;
                                                                                                            typedef int fct_t (int) ;
                 fct_t fct ;
                                                                                                            fct_t fct ;
                 int main(void){
                                                                                                            int main(void){
                                                                                           Syntaxe
                      return fct(2) ;
                                                                                                                 return fct(2) ;
                 int fct(int i){
                                                                                                            int fct(int i){
                     return i ;
                                                                                                                return i ;
                 Ce bout de code
                                                                                                            Ce bout de code
                   ▶ provoque une erreur à
                                                                                                              provoque une erreur à
                         ▶ la compilation :
                                                                                                                    ▶ la compilation : non
                         ► l'exécution :
                                                                                                                    ▶ l'exécution : non
                   ▶ fait ce qu'il devrait :
                                                                                                              ▶ fait ce qu'il devrait : oui
                                                                                                            Un prototype de fonction est une déclaration.
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                     www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
                                                                                            Pratique du C
Bétisier
 Pratique du C
Bétisier
                                                                                                            int fct(int i){
                 int fct(int i){
                     return i ;
                                                                                                                return i ;
                 int main(void){
                                                                                                            int main(void){
Syntaxe
                                                                                           Syntaxe
                     fct(2) :
                                                                                                                 fct(2) :
                    return 0 ;
                                                                                                               return 0 ;
                 Ce bout de code
                                                                                                            Ce bout de code
                   provoque une erreur à
                                                                                                              provoque une erreur à
                         ▶ la compilation :
                                                                                                                    ▶ la compilation : non
                         ► l'exécution :
                                                                                                                    ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                              ▶ fait ce qu'il devrait : oui
                                                                                                            Même si la fonction fct retourne une valeur, rien ne nous
                                                                                                            force à la récupérer.
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
                                                                                                                                                     www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                            Pratique du C
Bétisier
                 int MaVar = 1 ;
                                                                                                            int MaVar = 1 ;
                 int main(void){
                                                                                                            int main(void){
                                                                                                               return ++Mavar ;
                    return ++Mavar ;
                 Ce bout de code
                                                                                                            Ce bout de code
                                                                                                              ▶ provoque une erreur à
                   provoque une erreur à
                         ► la compilation :
                                                                                                                    ▶ la compilation : oui
                         ▶ l'exécution :
                                                                                                                    l'exécution :
                   ▶ fait ce qu'il devrait :
                                                                                                              ► fait ce qu'il devrait :
                                                                                                            Les identificateurs sont sensibles à la casse.
                                                         www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                    www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
```

```
int var(void){ return 2;}
                                                                                                         int var(void){ return 2;}
                int main(void){
                                                                                                         int main(void){
                     int a = 10, var = 1, cinq = a/var;
                                                                                                             int a = 10, var = 1, cinq = a/var ;
                     return cinq;
                                                                                                             return cinq;
                Ce bout de code
                                                                                                         Ce bout de code
                   provoque une erreur à
                                                                                                           provoque une erreur à
                        ▶ la compilation :
                                                                                                                 la compilation : non
                        ► l'exécution :
                                                                                                                 ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : non
                                                                                                         La porté des identificateurs n'est pas un vain mot.
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                int deux(void){ return 2 ; }
                                                                                                         int deux(void){ return 2 ; }
                int main(void){
                                                                                                         int main(void){
Syntaxe
                                                                                        Syntaxe
                     int (*deux) = deux ;
                                                                                                             int (*deux) = deux ;
                     return deux ;
                                                                                                             return deux ;
                                                                                                         Ce bout de code
                Ce bout de code
                   ▶ provoque une erreur à
                                                                                                           provoque une erreur à
                        ▶ la compilation :
                                                                                                                 ▶ la compilation : non
                        ► l'exécution :
                                                                                                                 ▶ l'exécution : non
                   ▶ fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : non
                                                                                                         Ce code retourne une adresse.
V84 (06-12-2012)
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                         Pratique du C
Bétisier
                On souhaite retourner la somme de deux entiers saisis au
                                                                                                         On souhaite retourner la somme de deux entiers saisis au
                clavier.
                                                                                                         clavier
                int
                                                                                                         int
                main(void){
                                                                                                         main(void){
                    int a=0, b=0;
                                                                                                            int a=0, b=0;
                                                                                                             scanf("%d%d",a,b);
                     scanf("%d%d",a,b);
                     return a+b :
                                                                                                             return a+b :
                }
                                                                                                         }
                Ce bout de code
                                                                                                         Ce bout de code
                   provoque une erreur à
                                                                                                           provoque une erreur à
                        ▶ la compilation :
                                                                                                                 ▶ la compilation : non
                        l'exécution :
                                                                                                                 ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                           ▶ fait ce qu'il devrait : non
                                                                                                         Le passage de paramètre par référence permet d'exporter de
                                                                                                         l'information d'une fonction.
                                                        www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
```

```
int SommeDes10Premiers(int tab[10]) {
                                                                                                      int SommeDes10Premiers(int tab[10]) {
                int i, res =0;
                                                                                                      int i, res =0;
                for(i=1;i<11;i++)
                                                                                                      for(i=1;i<11;i++)
                   res += tab[i];
                                                                                                         res += tab[i] ;
                   return res ;
                                                                                                         return res :
                7-
                Ce bout de code
                                                                                                      Ce bout de code
                  provoque une erreur à
                                                                                                         provoque une erreur à
                        ▶ la compilation :
                                                                                                              ▶ la compilation : non
                        ▶ l'exécution :
                                                                                                              ▶ l'exécution : non
                  ► fait ce qu'il devrait :
                                                                                                         ▶ fait ce qu'il devrait : non
                                                                                                      Les indices de tableaux commencent à 0 et rien ne vous
                                                                                                      empêche de passer outre.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                             www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                       Pratique du C
Bétisier
                int main(void){
                                                                                                      int main(void){
                     int zero = 0 :
                                                                                                           int zero = 0 :
                                                                                                            int tab[5] = \{1,2,3,4,10\};
                     int tab[5] = \{1,2,3,4,10\};
                     zero = 0[tab] + *(tab+1) + *(2+tab) + tab[3] - 4[tab];
                                                                                                            zero = 0[tab] + *(tab+1) + *(2+tab) + tab[3] - 4[tab];
Syntaxe
                     return zero ;
                                                                                                            return zero ;
                                                                                                      Ce bout de code
                Ce bout de code
                  provoque une erreur à
                                                                                                         provoque une erreur à
                        ▶ la compilation :
                                                                                                              ▶ la compilation : non
                        ▶ l'exécution :
                                                                                                              ▶ l'exécution : non
                  ▶ fait ce qu'il devrait :
                                                                                                         ▶ fait ce qu'il devrait : oui
                                                                                                      La notation tab[1] est équivalente à *(tab+1) ce qui permet
                                                                                                      ce code surprenant.
V84 (06-12-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                             www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                       Pratique du C
Bétisier
                #include<stdio.h>
                                                                                                      #include<stdio.h>
                int main(void){
                                                                                                      int main(void){
                     int i = 7, a[5]; char c = 0;
                                                                                                           int i = 7, a[5]; char c = 0;
                                                                                                      /* qu'est qui est affich\'e */
                /* qu'est qui est affich\'e */
                     printf( "%d", i++ * i++ );
                                                                                                           printf( "%d", i++ * i++ );
                   comment est modifi\'e ce tableau */
                                                                                                      /* comment est modifi\'e ce tableau */
                     i = 1; a[i] = i++;
                                                                                                           i = 1; a[i] = i++;
                   ce code peut il s'arr\^eter */
                                                                                                      /* ce code peut il s'arr\^eter */
                     do { c = getchar(); } while(c != EOF) ;
                                                                                                           do { c = getchar(); } while(c != EOF) ;
                  return 0 ;
                                                                                                        return 0 ;
                Ce bout de code
                                                                                                      Ce bout de code
                  provoque une erreur à
                                                                                                         provoque une erreur à
                        ▶ la compilation :
                                                                                                              ▶ la compilation : non
                        l'exécution :
                                                                                                              ▶ l'exécution : non
                  ▶ fait ce qu'il devrait :
                                                                                                         ▶ fait ce qu'il devrait :???
                                                                                                      Si on ne lit pas la documentation du compilateur ou l'assem-
                                                                                                      bleur, il existe des questions sans réponses i.e. la norme ne
                                                                                                      spécifie pas tout.
                                                                                                                                             www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
                                                      www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
```

```
/* ce code retourne 100 */
                                                                                                              /* ce code retourne 100 */
                 int main(void){
                                                                                                              int main(void){
                    i=0 ; j=1 ;
                                                                                                                 i=0 ; j=1 ;
                     while(i<100);
                                                                                            Syntaxe
                                                                                                                 while(i<100);
                       i=i+j ;
                                                                                                                    i=i+j ;
                    return i ;
                                                                                                                 return i ;
                 }
                 Ce bout de code
                                                                                                             Ce bout de code
                    provoque une erreur à
                                                                                                                provoque une erreur à
                         ▶ la compilation :
                                                                                                                      ▶ la compilation : non
                         ► l'exécution :
                                                                                                                      ► l'exécution : non
                    ▶ fait ce qu'il devrait :
                                                                                                                ▶ fait ce qu'il devrait : non
                                                                                                              Attention aux trivialités.
                                                           www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
                                                                                             Pratique du C
Bétisier
 Pratique du C
Bétisier
                                                                                                             char foo[10] = '\0';
char bar[10] = "\0";
                 char foo[10] = '\0';
                 char bar[10] = "\0";
                 int * ptr1, * ptr2 ;
                                                                                                             int * ptr1, * ptr2 ;
Syntaxe
                                                                                            Syntaxe
                 int * pptr1, pptr2;
                                                                                                             int * pptr1, pptr2;
                 pptr1 = ptr1;
pptr2 = ptr2;
                                                                                                             pptr1 = ptr1 ;
pptr2 = ptr2 ;
                 Ce bout de code
                                                                                                              Ce bout de code
                    provoque une erreur à
                                                                                                                provoque une erreur à
                         ► la compilation :
                                                                                                                      ► la compilation :
                         ► l'exécution :
                                                                                                                      ► l'exécution :
                    ▶ fait ce qu'il devrait :
                                                                                                                ▶ fait ce qu'il devrait :
                                                                                                              De petites différences produisent de grands effets.
V84 (06-12-2012)
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                             Pratique du C
Bétisier
                 int *a[10] :
                                                                                                              int *a[10] :
                 int (*b)[10];
                                                                                                              int (*b)[10];
                 b[9]=a[0];
                                                                                                             b[9]=a[0];
                 Ce bout de code
                                                                                                              Ce bout de code
                    provoque une erreur à
                                                                                                                provoque une erreur à
                         ► la compilation :
                                                                                                                      ▶ la compilation : oui
                          ▶ l'exécution :
                                                                                                                      l'exécution :
                    ► fait ce qu'il devrait :
                                                                                                                ► fait ce qu'il devrait :
                                                                                                              La compilation signale error : incompatible types in assign-
                                                                                                              ment. En effet, le pointeur d'entier a[0] pourrait "avoir" plus
                                                                                                              de 10 cellules.
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
```

```
int main(void){
                                                                                                       int main(void){
                /* Emboitemement de /* commentaire */ */
                                                                                                       /* Emboitemement de /* commentaire */ */
                int a,b,c = 3;
                                                                                                       int a,b,c = 3;
                int *pointer = &c ;
                                                                                                       int *pointer = &c ;
                   /*divise c par lui m\^eme */
                                                                                                          /*divise c par lui m\^eme */
                   a=c/*pointer ;
                                                                                                          a=c/*pointer;
                   b=c /*met b \'a 3*/;
                                                                                                          b=c /*met b \'a 3*/;
                   return 0 ;
                                                                                                          return 0 ;
                Ce bout de code
                                                                                                       Ce bout de code
                   provoque une erreur à
                                                                                                         provoque une erreur à
                        ▶ la compilation :
                                                                                                               la compilation : non
                        ► l'exécution :
                                                                                                               ▶ l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                         ▶ fait ce qu'il devrait : non
                                                                                                       Attention aux commentaires.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                              www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                        Pratique du C
Bétisier
                #include<stdio.h>
                                                                                                       #include<stdio.h>
                char *foo(void){
                                                                                                       char *foo(void){
                    char str[31]="Pourquoi vais-je disparaitre~?" ;
                                                                                                           char str[31]="Pourquoi vais-je disparaitre~?" ;
Syntaxe
                                                                                      Syntaxe
                int main(void){
                                                                                                       int main(void){
                   printf("%d %d %s\n",1,2,foo());
                                                                                                          printf("%d %d %s\n",1,2,foo()) ;
                   return 0;
                                                                                                          return 0;
                Ce bout de code
                                                                                                       Ce bout de code
                   provoque une erreur à
                                                                                                         provoque une erreur à
                        ▶ la compilation :
                                                                                                               ▶ la compilation : non
                                                                                                               l'exécution : non
                        l'exécution :
                                                                                                         ► fait ce qu'il devrait : non
                   ▶ fait ce qu'il devrait :
                                                                                                       Le compilateur indique que warning: function returns
                                                                                                       address of local variable.
V84 (06-12-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                              www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                        Pratique du C
Bétisier
                int
                                                                                                       int
                                                                                                       main
                   char *str = "stringX" ;
                                                                                                          char *str = "stringX" ;
                   str[6]='Y';
                                                                                                          str[6]='Y';
                   return 0;
                                                                                                          return 0;
                Ce bout de code
                                                                                                       Ce bout de code
                  provoque une erreur à
                                                                                                         provoque une erreur à
                                                                                                               ▶ la compilation : non
                        la compilation :
                        ► l'exécution :
                                                                                                               l'exécution : non
                   ► fait ce qu'il devrait :
                                                                                                         ► fait ce qu'il devrait : oui
                                                                                                       La chaîne "stringX" est stockée dans un segment en lec-
                                                                                                       ture seule (.rodata) et la tentative d'accès se solde par un
                                                                                                       segmentation fault.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
                                                                                                                                             www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
```

```
#include<stdio.h>
                 struct tab_s{ int tab[2]; };
                 struct tab s foo(void){
                     struct tab_s res ;
                      res.tab[0] = 1 ; res.tab[1] = 2 ;
                      return res ;
                 int main(void){
                    struct tab_s tmp = foo() ;
                     int zero = 2*tmp.tab[0] - tmp.tab[1] ;
                    return zero;
                 }
                 Ce bout de code
                   provoque une erreur à
                         ▶ la compilation :
                         ▶ l'exécution :
                    ▶ fait ce qu'il devrait :
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                             Pratique du C
Bétisier
                 Ce code retourne la taille de la chaîne de caractères
                 int main(void){
                     char *ch1 = "Hello world";
Syntaxe
                                                                                           Syntaxe
                     return sizeof(ch1) ;
                 Ce bout de code
                    provoque une erreur à
                         ▶ la compilation :
                         ▶ l'exécution :
                    ▶ fait ce qu'il devrait :
V84 (06-12-2012)
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
 Pratique du C
Bétisier
                                                                                             Pratique du C
Bétisier
                 Une implantation de memset
                 void *memset(void *b, int c, int len)
                   int i ;
                   for(i=0;i<len;i++)
                      b[i]=c ;
                   return b ;
                 Ce bout de code
                   provoque une erreur à
                         ► la compilation :
                         ► l'exécution :
                    ▶ fait ce qu'il devrait :
V84 (06-12-2012)
                                                          www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
```

```
#include<stdio.h>
struct tab_s{ int tab[2]; };
struct tab s foo(void){
   struct tab_s res ;
   res.tab[0] = 1 ; res.tab[1] = 2 ;
   return res ;
int main(void){
  struct tab_s tmp = foo() ;
  int zero = 2*tmp.tab[0] - tmp.tab[1] ;
  return zero;
Ce bout de code
 provoque une erreur à
      ▶ la compilation : non
       l'exécution : non
 ▶ fait ce qu'il devrait : oui
On ne peut affecter un tableau à un autre, c'est possible pour
```

Ce code retourne la taille de la chaîne de caractères

```
int main(void){
  char *ch1 = "Hello world";
  return sizeof(ch1);
```

Ce bout de code

les structures.

- provoque une erreur à
 - ▶ la compilation : non
 - ▶ l'exécution : non
- ▶ fait ce qu'il devrait : non

N'utilisez sizeof que sur des types.

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
Une implantation de memset
```

```
void *memset(void *b, int c, int len)
 int i ;
 for(i=0;i<len;i++)
    b[i]=c ;
 return b ;
```

Ce bout de code

- provoque une erreur à
 - ▶ la compilation : non
 - l'exécution :
- ► fait ce qu'il devrait :

Ni arithmétique ni déréférencement de pointeur void .

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
int main(void){
                                                                                                        int main(void){
                   int *p ;
                                                                                                           int *p ;
                   p = malloc(10*sizeof (int));
                                                                                                           p = malloc(10*sizeof (int));
                   p[10] = 666;
                                                                                                           p[10] = 666;
                   return 0 ;
                                                                                                           return 0 ;
                Ce bout de code
                                                                                                        Ce bout de code
                  provoque une erreur à
                                                                                                          provoque une erreur à
                        ▶ la compilation :
                                                                                                                ▶ la compilation : non
                                                                                                                ▶ l'exécution : non
                        l'exécution :
                  ► fait ce qu'il devrait :
                                                                                                          ▶ fait ce qu'il devrait : non
                                                                                                        Affectation hors limite.
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
V84 (06-12-2012)
 Pratique du C
Bétisier
                #include<stdio.h>
                                                                                                        int main(void) {
                #include <stdlib.h>
                                                                                                        int i ;
                /* strtod, strtof, strtold - convert
                                                                                                        int tampon[5] ;
                ASCII string to floating point number
Syntaxe
                                                                                       Syntaxe
                                                                                                        for(i=0;i<666;i++)
                float strtof(const char *, char **); */
                                                                                                        tampon[i]=2 ;
                int main(int argc, char **argv){
                                                                                                        return 0;
                  printf("%f\n", strtof(argv[1], NULL));
                  return 0 ;
                                                                                                        Ce bout de code
                                                                                                          provoque une erreur à
                Si on compile avec l'option ansi, le prototype de la fonction
                                                                                                                ► la compilation :
                strtof - n'étant pas dans cette norme et étant à l'intérieur
                                                                                                                l'exécution :
                d'une directive conditionnelle — n'est pas pris en compte.
                                                                                                          ▶ fait ce qu'il devrait :
                Sans prototype, la valeur de retour de la fonction est
                supposée être un entier machine et le résultat n'est pas codé
                comme un flottant et donc faux.
V84 (06-12-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
 Pratique du C
Bétisier
                                                                                         Pratique du C
Bétisier
                int main(void) {
                                                                                                        Le code
                int i ;
                                                                                                        int main(void){
                int tampon[5] ;
                                                                                                            int tab[10] ;
                for(i=0;i<666;i++)
                                                                                                            int i = 0;
                tampon[i]=2 ;
                                                                                                            for(;i<300;i++)
                return 0;
                                                                                                                tab[i] = i ;
                }
                                                                                                            return 0 ;
                Ce bout de code
                                                                                                        donne l'exécution suivante :
                  provoque une erreur à
                                                                                                        espoir % gcc code.c
                        la compilation : non
                                                                                                        espoir % a.out
                        ▶ l'exécution : non
                                                                                                        Segmentation fault
                   ▶ fait ce qu'il devrait : non
                Cela provoque une boucle infinie!!!
V84 (06-12-2012)
                                                       www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
                                                                                                                                               www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf
```

V84 (06-12-2012)

```
Pratique du C
Bétisier
```

V84 (06-12-2012) Pratique du C Bétisier

V84 (06-12-2012)

```
int tab[10]; /* dans le fichier foo */
extern int * tab ; /* dans le fichier bar */
```

Il n'y a pas d'allocation de mémoire lors de la déclaration d'un pointeur (hormis la mémoire utilisée pour contenir l'adresse).

```
extern foo; /* ceci est valide (c'est un entier) */
```

Il existe beaucoup de règles implicites en C.

```
www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
```

```
int bar(int n) { return n+2 ;}
int
main
(void)
   int (*foo) (int) = bar ;
   foo++ ;
                                                                 Syntaxe
   return foo(1)
Quid de l'arithmétique des pointeurs de fonctions?
```

Ce bout de code

- provoque une erreur à
 - ▶ la compilation : oui
 - ► l'exécution :
- ► fait ce qu'il devrait :

Comme on ne connait pas la taille d'une fonction mais seulement son adresse et que cette taille est variable pour de fonctions de même signature, impossible de faire de l'arithmétique des pointeurs de fonctions.

```
www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf V84 (06-12-2012)
```

```
int bar(int n) { return n+2 ;}
main
(void)
   int (*foo) (int) = bar ;
   foo++ ;
   return foo(1)
```

Quid de l'arithmétique des pointeurs de fonctions? Ce bout de code

- provoque une erreur à
 - ► la compilation :
 - ▶ l'exécution :
- ▶ fait ce qu'il devrait :

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
int
bar
(int n)
   int tab[n] ;
   /* comparer avec malloc */
   return ;
```

Ce bout de code

- provoque une erreur à
 - ▶ la compilation :
 - ► l'exécution :
- ▶ fait ce qu'il devrait :

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf

```
int
bar
(int n)
   int tab[n] ;
   /* comparer avec malloc */
   return ;
Il n'y a pas que l'ANSI dans la vie.Ce bout de code
```

- provoque une erreur à
 - ▶ la compilation : oui
 - ▶ l'exécution :
- ► fait ce qu'il devrait :

Ce code est valide en C99.

www.fil.univ-lille1.fr/~sedoglav/C/Cours11.pdf