**TD1**

Exercice 1

1. LP64 correspond a 4/4/8 (int/long int/int \*) ->Linux
2. sizeof(long int) et sizeof(int \*) changent

Donc :

-Les tailles des structs qui utilisent ces types changent

-Si un long int utilise une valeur >max(int) , calcul incorrect

-Si allocation de plus de 2GB d mémoire , echec

1. permet de construire des structs ou des paquets de taille fixe

Exercice 2

1. Conversion d’un type a un autre avec une taille de support plus grande

-int 16bits -> int 32 bits

-float 32 bits-> double 64 bits

1. Oui, car le type ne change pas, seule la taille du support augmente
2. Troncatenation modulo 2^n où n est le nb de bit du type plus petit
3. R1=(100\*3)=300 modulo 256 =44

R2=(44/4)=11 modulo 256 = 11

R3=(100\*3/4)=75 modulo 256=75

(Modulo car unsigned)

1. Oui : tout entier de MSB(Most Significatif Bit) est inferieur ou égale à la taille de la mantisse
2. Tous les entiers de la question précedente
3. Troncature de -3.2=-3 et 3.2=3

floorf de -3.2=-4 et 3.2=3

ceilf de -3.2=-3 et 3.2=4

Exercice 3

1. lvalue=expression identifiable (on peut récupérer son adresse)

rvalue= non identifiable(on ne peut pas)

1. int a ->lvalue de type int

int &a -> ref à lvalue

int &&a -> ref à une rvalue

int \*a -> pointeur , type different

1. int &&a= (b+5)/value , a est de type ref à rvalue , **MAIS** à est une lvalue
2. a gauche toujours une lvalue car I/ND

I=identifiable

D= déplaçable

Nx= Non x

|  |  |  |
| --- | --- | --- |
| Code | lvalue/rvalue (droite) | Valide |
| int a= 5 | rvalue NI/D | Oui rvalue copié de lvalue |
| int &b=a | lvalue I/ND | Oui ref lvalue depuis lvalue |
| int c=b | lvalue I/ND | Oui lvalue copie ref lvalue |
| int d=(a+4)/2 | rvalue NI/D | Oui rvalue copie d’un lvalue |
| int &e =5 | rvalue NI/D | Non rvalue ref lvalue |
| int &f =a/2 | rvalue NI/D | Non rvalue ref lvalue |
| int &&g = a | lvalue I/ND | Non lvalue ref rvalue |
| int &&h=b | lvalue I/ND | Non |
| int &&i =a/4 | rvalue NI/D | Oui ref rvalue depuis rvalue |
| int &&j = 8 | rvalue NI/D | Oui ref rvalue depuis rvalue |

|  |  |  |
| --- | --- | --- |
| Code | lvalue/rvalue (droite) | Valide |
| int A = fun1() | rvalue(NI/D) | Oui lvalue depuis rvalue |
| int &B=fun2(a) | lvalue(I/ND) | Oui ref lvalue depuis lvalue |
| int &&C =fun3(5) | xvalue(I/D) | Oui ref rvalue depuis rvalue |
| int D = fun2(a) | lvalue(I/ND) | Oui lvalue depuis rfe lvalue |
| int E= fun3(7) | xvalue(I/D) | Oui lvalue depuis ref lvalue |
| int &F=fun3(a) | xvalue(I/D) | Non |
| int &G =fun1(a) | rvalue(NI/D) | Non |
| int &&H=fun2(a) | lvalue(I/ND) | Non |
| int &&I=fun1() ; | rvalue(NI/D) | Oui |

1. const int ,entier constant

const int& ,ref vers un entier constant

const int&& , ref vers un entier temporaire constant

|  |  |  |
| --- | --- | --- |
| Code | lvalue/rvalue (droite) | Valide |
| const int a= 5 | rvalue NI/D | Oui rvalue copié de lvalue |
| const int &b=a | lvalue I/ND | Oui ref lvalue depuis lvalue |
| const int c=b | lvalue I/ND | Oui lvalue copie ref lvalue |
| const int d=(a+4)/2 | rvalue NI/D | Oui rvalue copie d’un lvalue |
| const int &e =5 | rvalue NI/D | Oui |
| const int &f =a/2 | rvalue NI/D | Oui |
| const int &&g = a | lvalue I/ND | Non lvalue ref rvalue |
| const int &&h=b | lvalue I/ND | Non |
| const int &&i =a/4 | rvalue NI/D | Oui ref rvalue depuis rvalue |
| const int &&j = 8 | rvalue NI/D | Oui ref rvalue depuis rvalue |

Exercice 6

