

Rapport Projet 1

SR01

Exercice 3:

1) Programme 1:

-> Le programme 1 ne peut pas être exécuté car il contient un erreur: lvalue required as increment operand. On peut comprendre que !B ne change pas la valeur de B, c'est-à-dire B est toujours égal à 5. Donc pour réaliser l'opérande ++, il nous faut un variable avec une valeur indiquée. Alors, pour corriger le programme en maintenant l'idée de sujet, nous avons posé une idée comme ci-dessous:

```
#include <stdio.h>

int main(){
int A=20, B=5;

int D = !B;
int C=!--A/++D;
printf("A=%d B=%d c=%d \n",
A,B,C);
}
```

-> Nous avons compris que l'opérande ! a premièrement réalisé puis ++ donc nous avons posé D = !B = 0 pour que le programme fonctionne. Après l'exécution, nous avons trouvé que C = 0, A = 19 et B = 5. En effet, en remplaçant les valeurs de A, B et D dans l'opération pour calcul C, on a: $C = !(20-1)/(0+1) = !19 = 0$; $A = A-1 = 20-1 = 19$ et B toujours égal à 5 comme l'explication avant.

2) Programme 2:

-> On sait que tous les valeurs différents à 0 sont égales à 1 dans && et || logique.

-> De plus, on a: !0 = 1; C++ c'est-à-dire on applique C = -10 dans l'opération puis C=C+1; !D++ c'est-à-dire D++ en premier puis !D. Néanmoins, dans ce programme, vu que

$A \&\& B$ est égale à 1 et l'opérande après est $||$, on n'a pas besoin d'observer les opérations après, c'est-à-dire $0 \&\& C++ \&\& !D++$ est ignoré.

-> Donc, avec $A=20$, $B=5$, $C=-10$, $D=2$, on a: $A \&\& B || !0 \&\& C++ \&\& !D++ = 1$ et C, D restent en même valeur.

3) Programme 3:

-> On sait que q est l'adresse de $p[0]$ et $*q=p[0]$ initialement. $*++q = *(q+1) = p[1] = -2$ et $*q++ = *q = p[1] = -2$ puis $q = q+1$ ($*q$ maintenant égale à $p[3]$)

-> Donc l'opération $*++q * *q++ = p[1] * p[1] = -2 * -2 = 4$; et $*q = p[3] = 3$.

4) Programme 4:

-> $*q = p[0] = 1 = 001_2$. $*q++ = *q = p[0] = 001_2$ puis $q=q+1$ ($*q$ maintenant $= p[1] = -2$). $*q++ = *q = p[1] = -2 = 110_2$

-> Donc $d = 001 \& 001 || 110 = 111_2 = -1_{10}$ et $*q = p[2] = 3$.

5) Programme 5:

-> $c = ++a \&\& -b ? b -- : a++$ c'est-à-dire on observe l'opération $++a \&\& -b$, si vraie, on fait l'opération $c = b --$, si non, on fait $c = a++$. Dans ce cas-là, on a $++a \&\& -b = -7 \&\& 2 = 1 \&\& 1 = 1$ donc on fait $c = b --$ d'où $b = 1$ et $c = 2$. Alors, on trouve $a = -7$, $b = 1$, $c = 2$.

6) Programme 6:

-> $a = -8_{10} = 1000_2$; avec $b = 3_{10} = 011_2$, on a: $2^b = 010$ ou exclusif avec $011 = 001 = 1_{10}$ donc $a >>= 2^b$ c'est-à-dire a décale 1 bit vers la droite donc $a = 0100_2 = (-4)_{10}$