

1. On voit les instructions machine directement dans la mémoire, à droite. Il y a 4 adresses prises par le programme, donc cela fait une longueur de 8 octets.
2. La plus grande somme que l'on puisse obtenir est 32 767, « après on passe à $-32\,768$ ». On peut donc remarquer que la plage de valeurs autorisée est de -2^{15} à $2^{15} - 1$. En faisant l'analogie avec le complément à 2 sur 8 bits, on peut raisonnablement penser que le format utilisé est le complément à 2 sur 16 bits.
3. On change la ligne 2 en `SUB R2, R1, R0`.
4. Pas de question 4.
5. Le programme affiche 3 et seul le flag C est à 1.
6. Avec 2 et 2 il affiche 3 mais c'est le flag N qui est à 1.
Avec 3 et 3 il affiche 3 et là les flags Z et C sont à 1.
7. N passe à 1 quand un résultat est négatif, Z lorsqu'un résultat est nul et pour C c'est moins clair : il passe à 1 lorsqu'une retenue survient.
- 8.

Code assembleur

```
INP R0, 2      // Lire R0 au clavier
MOV R1, #0     // Initialiser R1 à 0
bcl: ADD R1, R1, R0 // Ajouter R0 dans R1
SUB R0, #1     // Retirer 1 à R0
BNE bcl        // Si R0 n'est pas nul, recommencer
OUT R1, 4      // Sinon afficher la somme
HLT           // Et s'arrêter
```