**Exercices supplémentaires** (La solution est à la fin du fichier)

--- Conversions

--- Représentation interne des entiers signés et non signés

1. Faire les conversions suivantes :

a)45610 = ( )2

b) 1001101012 = ( )10

c) F82116 = ( )2

d) 101001101111100112 = ( )16

1. Donner la représentation interne 8 bits (entiers signés) de
2. -129
3. 102
4. 200
5. -98
6. -128
7. -60
8. Donner la représentation interne 16 bits (entiers signés) de
9. -129
10. 1026
11. 208
12. -980
13. -128
14. -60

Donnez votre réponse en hexadécimal

1. Donner la représentation interne 32 bits (entiers signés) de
2. -1299
3. 1026
4. 2 222 222 222
5. -9800
6. -130
7. -60

Donnez votre réponse en hexadécimal

1. Refaire les numéros 2,3 et 4 mais cette fois avec des entiers non signés
2. Un entier signé a été codé en RAM sur 8 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
3. 01010101
4. 10101010
5. 10010101
6. 11111111
7. 10000001
8. 01111110
9. Un entier signé a été codé en RAM sur 16 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
10. 14FA
11. 1B38
12. 0915
13. CEAB
14. AA68
15. 09C8
16. Un entier signé a été codé en RAM sur 32 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
17. 14FA14FA
18. 10101B38
19. 64210915
20. CEAB1214
21. AA68AB12
22. 09C824CD
23. Refaire les numéros 6,7 et 8 mais cette fois avec des entiers non signés
24. Soit le programme suivant en RAM avant son exécution :

|  |  |
| --- | --- |
|  | **Déclarations** |
|  | TERME1 : entier signé, 1 octet |
|  | TERME2 : entier signé, 1 octet |
|  | PRODUIT: entier signé, 2 octets |
|  |  |
|  | **Début du programme** |
|  | Lire TERME1, TERME2 |
|  | PRODUIT=TERME1 \*TERME2 |
|  | Afficher PRODUIT |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Donnez le schéma de la réservation des variables en RAM
2. si les valeurs lues pour les variables TERME1 et TERME2 sont respectivement -1210 et -2010, donnez le contenu des variables TERME1, TERME2 et PRODUIT en RAM après l’exécution du programme
3. La réservation pour la variable PRODUIT est-elle optimale ? (C’est-à-dire, aurait-on pu prendre un seul octet pour la variable PRODUIT ? Aurait-il été mieux de prendre 4 octets au lieu de 2 ?)

11.Soit le programme suivant en RAM ainsi que le contenu des variables après l’exécution du programme :

|  |  |
| --- | --- |
|  | **Déclarations** |
|  | G : entier signé, 1 octet |
|  | H : entier signé, 1 octet |
|  | SOMME: entier signé, 2 octets |
|  |  |
|  | **Début du programme** |
|  | Lire G,H |
|  | SOMME = G + H |
|  | Afficher G,H |
|  | Afficher SOMME |
|  |  |
| G | D8 |
| H | E6 |
| SOMME | FF |
|  | BE |
|  |  |
|  |
|  |

a)Quelles seront les valeurs affichées pour les variables G et H ? (en base 10)

b)Quelle sera la somme affichée par le programme ? (en base 10)

c) La réservation pour la variable SOMME est-elle optimale ?

12.Refaire l’exercice précédent avec des variables non signées :

|  |  |
| --- | --- |
|  | **Déclarations** |
|  | G : entier **non signé**, 1 octet |
|  | H : entier **non signé,** 1 octet |
|  | SOMME: entier **non signé,** 2 oct |
|  |  |
|  | **Début du programme** |
|  | Lire G,H |
|  | SOMME = G + H |
|  | Afficher G,H |
|  | Afficher SOMME |
|  |  |
| G | D8 |
| H | E6 |
| SOMME | 01 |
|  | BE |
|  |  |

SOLUTION

1. Faire les conversions suivantes :

a)45610 = ( )2

456

256 128 64 32 16 8 4 2 1

1 1 1 0 0 1 0 0 0

456 – 256 = 200

200 – 128 = 72

72-64 = 8

8-8 =0

Réponse : 111001000

b) 1001101012 = ( )10

1 0 0 1 1 0 1 0 1

256 32 16 4 1

256+32+16+4+1 = 309

c) F82116 = ( )2

F 8 2 1

1111 1000 0010 0001

Réponse : 1111100000100001

d) 1010 0110 1111 10112 = ( )16

1010 0110 1111 1011

A 6 F B

Réponse : A6FB

1. Donner la représentation interne 8 bits (entiers signés) de
2. Impossible de représenter sur 8 bits
3. 01100110
4. Impossible de représenter sur 8 bits
5. 10011110
6. 10000000
7. 11000100
8. Donner la représentation interne 16 bits (entiers signés) de
9. FF7F
10. 0402
11. 00D0
12. FC2C
13. FF80
14. FFC4
15. Donner la représentation interne 32 bits (entiers signés) de
16. FFFFFAED
17. 00000402
18. Impossible de représenter sur 32 bits
19. FFFFD9B8
20. FFFFFF7E
21. FFFFFFC4
22. Numéros 2,3 et 4 en entiers non signés

Numéro 2 en entiers non signés :

1. Impossible de représenter un négatif en entier non signé
2. 01100110
3. 11001000
4. Impossible de représenter un négatif en entier non signé
5. Impossible de représenter un négatif en entier non signé
6. Impossible de représenter un négatif en entier non signé

Numéro 3 en entiers non signés :

1. Impossible de représenter un négatif en entier non signé
2. 0402
3. 00D0
4. Impossible de représenter un négatif en entier non signé
5. Impossible de représenter un négatif en entier non signé
6. Impossible de représenter un négatif en entier non signé

Numéro 4 en entiers non signés :

a) Impossible de représenter un négatif en entier non signé

b) 00000402

c) 84746B8E

d) Impossible de représenter un négatif en entier non signé

e) Impossible de représenter un négatif en entier non signé

f) Impossible de représenter un négatif en entier non signé

1. Un entier signé a été codé en RAM sur 8 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
2. 85
3. -86
4. -107
5. -1
6. -127
7. 126
8. Un entier signé a été codé en RAM sur 16 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
9. 5370
10. 6968
11. 2325
12. -4437
13. -21912
14. 2504
15. Un entier signé a été codé en RAM sur 32 bits. Si on demande l’affichage à l’écran de cet entier, que sera-t-il affiché ?
16. 351 933 690
17. 269 491 000
18. 1 679 886 613
19. -827 649 516
20. -1 435 981 038
21. 164 111 565
22. Refaire les numéros 6,7 et 8 mais cette fois avec des entiers non signés

Numéro 6 Entiers non signés :

1. 85
2. 170
3. 149
4. 255
5. 129
6. 126

Numéro 7 Entiers non signés :

1. 5370
2. 6968
3. 2325
4. 52907
5. 43624
6. 2504

Numéro 8 Entiers non signés :

1. 351 933 690
2. 269 491 000
3. 1 679 886 63
4. 3 467 317 780
5. 2 858 986 258
6. 164 111 565
7. a)

|  |  |
| --- | --- |
|  | **Déclarations** |
|  | TERME1 : entier signé, 1 octet |
|  | TERME2 : entier signé, 1 octet |
|  | PRODUIT: entier signé, 2 octets |
|  |  |
|  | **Début du programme** |
|  | Lire TERME1, TERME2 |
|  | PRODUIT=TERME1 \*TERME2 |
|  | Afficher PRODUIT |
|  |  |
|  |  |
| TERME1 |  |
| TERME2 |  |
| PRODUIT |  |
|  |  |
|  |  |
|  |  |
|  |  |

b) TERME1 : 11110100

TERME2 : 11101100

PRODUIT : -12 \* -20 = 240 et 240 sur 16 bits (entier signé) = 0000000011110000

c) Oui : le maximum (positif) dans un octet = 127 (entier signé).

127 \* 127 = 16 129 et 16129 entre dans 2 octets.

C’est bon aussi du côté négatif.

1. a) Affichage de G = -40

Affichage de H = -26

b) La variable SOMME contient la somme de G et de H, donc -40 + -26 , c’est-à-dire -66

On peut vérifier avec le contenu de la variable SOMME en RAM. SOMME contient FFBE.

FFBE = -66.

1. Oui : le maximum (positif) dans un octet = 127 (entier signé).

127 + 127 = 254 et 254 entre dans 2 octets.

C’est bon aussi du côté négatif.

1. a) Affichage de G = 216

Affichage de H = 230

b) La variable SOMME contient la somme de G et de H, donc 216 + 230 , c’est-à-dire 446.

On peut vérifier avec le contenu de la variable SOMME en RAM. SOMME contient 01BE.

01BE = 446.

1. Oui : le maximum (positif) dans un octet = 255 (entier non signé).

255 + 255 = 510 et 510 entre dans 2 octets.

Comme c’est non signé, on n’a pas à tester pour les négatifs.