# Semaine 1 – Langage machine – Exercices

## Exercice n°1.1 – Affectations de valeurs hexadécimales dans des registres du processeur

Téléchargez sur <http://moodle.vinci.be> le fichier **Semaine.1.Exercice.1.asm** et ouvrez-le dans l’IDE – l’environnement de développement – SASM.

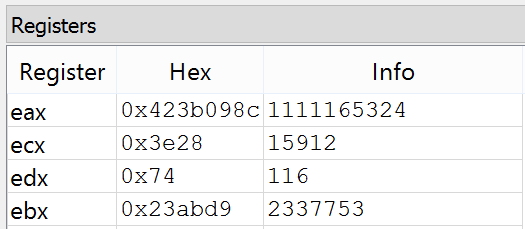
Ecrivez les instructions en langage assembleur NASM 32 bits pour initialiser les **registres** généraux eax, ebx, ecx, edx aux valeurs **hexadécimales** suivantes :

* eax = 0x423B098C
* ebx = 0x23ABD9
* ecx = 0x3E28
* edx = 0x74

Vos réponses :

|  |  |
| --- | --- |
| Affectations | Instructions en langage assembleur |
| eax = 0x423B098C | mov eax,0x423B098C |
| ebx = 0x23ABD9 | mov ebx,0x231BD9 |
| ecx = 0x3E28 | mov ecx,0x3E28 |
| edx = 0x74 | mov edx,0x74 |

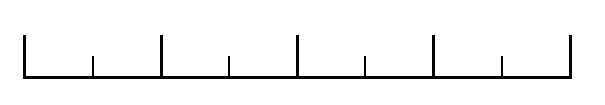
Examinez dans l’IDE SASM ligne par ligne l’exécution en mode *Debug* de votre programme. Appuyez sur la touche **F5** pour commencer l’exécution de votre programme en mode *Debug*. Dans le menu *Debug*, cochez *Show registers*. Appuyez sur la touche **F10** pour continuer ligne par ligne l’exécution de votre programme. Pour vérifier votre programme, vous pouvez observer dans l’IDE la fenêtre suivante qui montre les registres dans le processeur :



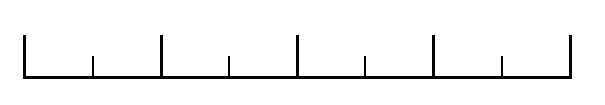
Que représentent les valeurs dans la colonne « Info » ?

Les nombres rentrés en hexadécimal dans le programme transformé en nombre décimal

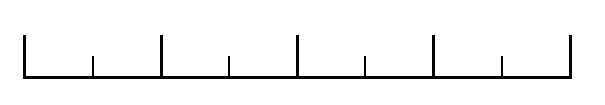
Suite aux affectations précédentes, complétez le dessin du CPU (processeur 32 bits)   
ci-dessous :



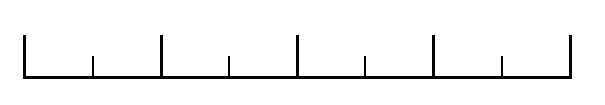
eax



ebx



ecx



edx

eax 42 3B 09 8C

ebx 23 AB D9

ecx 3E 28

edx 00 00 00 74

Les registres d’un processeur 32 bits peuvent accueillir des valeurs jusque 8 chiffres hexadécimaux. On parle de *double word*.

Le langage assembleur NASM permet les tailles de données suivantes :

* le ***b****yte* (1 octet = 8 bits = 1 byte)
* le ***w****ord* (2 octets = 16 bits = 2 bytes)
* le ***d****ouble word* (2 *word* = 2x2 octets = 4 octets = 32 bits = 4 bytes)
* le ***q****uad word* (4 *word* = 4x2 octets = 8 octets = 64 bits = 8 bytes)
* l'*ex****T****ended* (10 octets = 10 bytes = 80 bits pour les réels : *extended-precision float*)

Combien de chiffres hexadécimaux pouvez-vous écrire dans un *byte* ?

2 chiffres

Combien de chiffres hexadécimaux pouvez-vous écrire dans un *word* ?

4 chiffres hexadécimaux

Combien de chiffres hexadécimaux pouvez-vous écrire dans un *double word* ?

8 chiffres hexadécimaux

Combien de chiffres hexadécimaux pouvez-vous écrire dans un *quad word*

16 chiffres hexadécimaux

Combien de chiffres hexadécimaux pouvez-vous écrire dans un *exTended* ?

20 chiffres hexadécimaux

## Exercice n°1.2 – Affectations de valeurs décimales dans des registres du processeur

Comptez en hexadécimal à partir de zéro en remplissant le tableau suivant :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F |

Exécutez à l’aide de SASM les instructions suivantes :

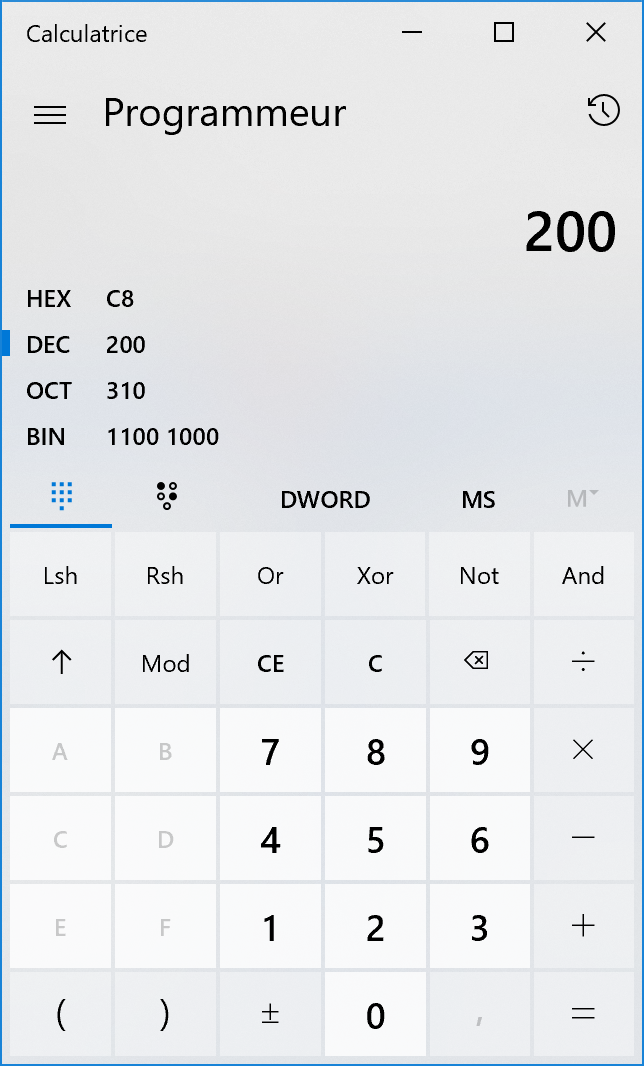
* mov eax,27
* mov ebx,200
* mov ecx,10
* mov edx,1048576

Remarquez que les valeurs sont écrites ci-dessus en décimal. Il n’y a pas de 0x devant la valeur. Observez que les registres sont toujours initialisés en hexadécimal lors de l’exécution.

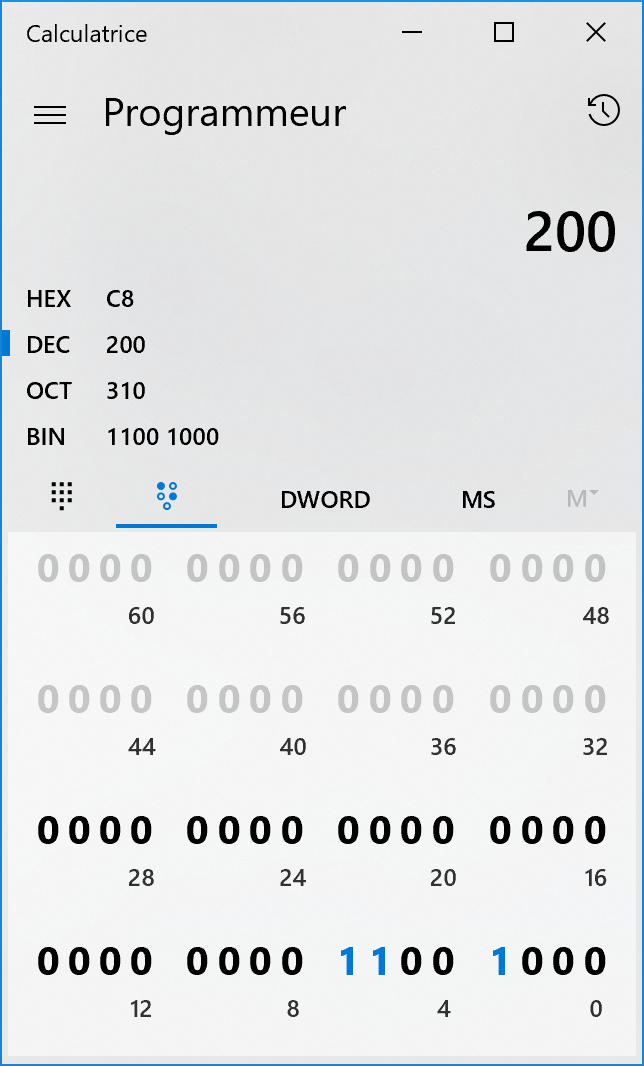
|  |  |
| --- | --- |
| Instructions en langage assembleur | Affectations (écrivez en hexadécimal 0x…) |
| mov eax,27 | eax = 0x1B |
| mov ebx,200 | ebx = 0xC8 |
| mov ecx,10 | ecx = 0xA |
| mov edx,1048576 | edx = 0x100000 |

Pour convertir un nombre d’une base à une autre, vous pouvez utiliser la **calculatrice de Windows** en mode programmeur.

Par exemple, vérifiez que 200 vaut bien 0xC8 en hexadécimal, vous entrez la valeur en décimal, sélectionnez une taille suffisante (Dword par exemple) et cliquez le bouton Hex :



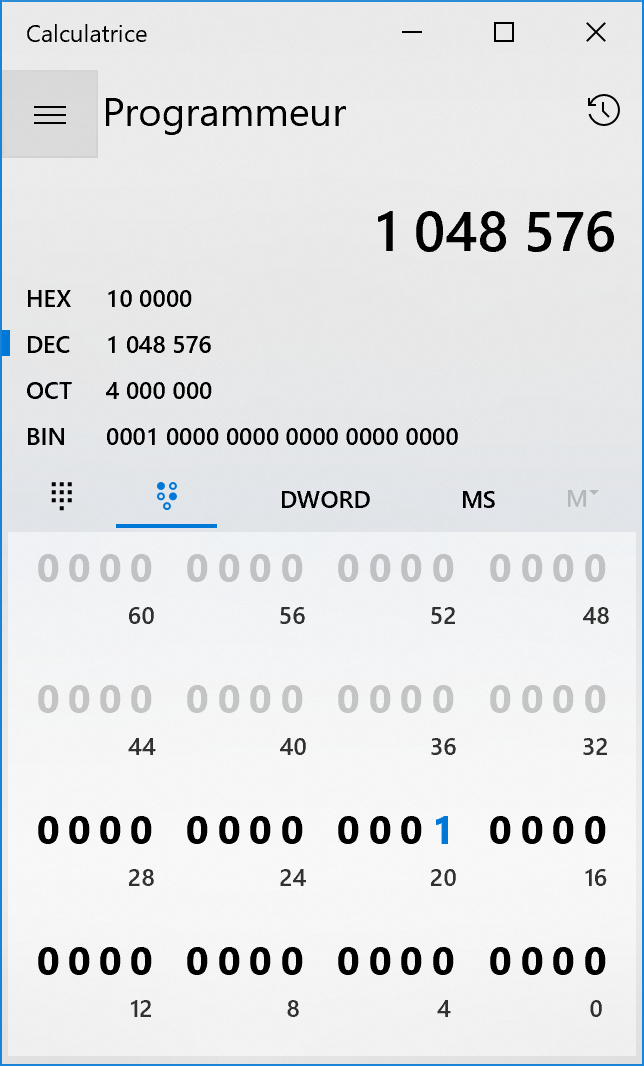
Remarquez que la valeur 0xC8 = 200 est aussi exprimée en binaire sur la taille sélectionnée : (32 bits dans l’exemple ci-dessous)



Les bits sont numérotés de droite à gauche à partir du moins significatif jusqu’au plus significatif. On parle de numérotation en ordre des bits petit boutiste (petit boutien) ou, en anglais : *little endian bit order*.

Vérifiez que 1048576 en décimal vaut 0x100000 en hexadécimal.

Remarquez que la valeur en binaire sur 32 bits de 1048576 en décimal vaut



Quelle est donc la puissance (l’exposant) de 2 qui donne 1048576 en décimal ?

20

## Exercice n°1.3 – Affectations de valeurs binaires dans des registres du processeur

Ecrivez de droite à gauche les puissances de 2 jusque 1024 :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 210 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Exécutez à l’aide de SASM les instructions suivantes :

* mov eax,11001000b
* mov ebx,11111111b
* mov ecx,10b
* mov edx,10101010101010101010101010101010b

Remarquez que les valeurs sont écrites ci-dessus en binaire. Il n’y a pas de 0x devant la valeur mais un b derrière. Observez que les registres sont toujours initialisés en hexadécimal lors de l’exécution.

|  |  |
| --- | --- |
| Instructions en langage assembleur | Affectations (écrivez en hexadécimal 0x…) |
| mov eax,11001000b | eax = 0xc8 |
| mov ebx,11111111b | ebx = 0xff |
| mov ecx,10b | ecx = 0x2 |
| mov edx,10101010101010101010101010101010b | edx = 0xaaaaaaaa |

Il est un peu long d’écrire les valeurs directement en binaire, c’est ainsi que les environnements de développement orienté assembleur affichent les valeurs en hexadécimal.

Le passage de l’un à l’autre est très facile car chaque chiffre hexadécimal a un équivalent en binaire sur 4 chiffres binaires.

Par exemple 1100 en binaire est C en hexadécimal, 1000 en binaire est 8 en hexadécimal. C8 en hexadécimal est donc 11001000 en binaire. On peut écrire que 0xC8 équivaut à 110001000b.

Comptez en binaire **sur 4 chiffres** à partir de zéro en remplissant le tableau suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| 0000 | 0001 | 0010 | 0011 |
| 4 | 5 | 6 | 7 |
| 0100 | 0101 | 0110 | 0111 |
| 8 | 9 | 10 | 11 |
| 1000 | 1001 | 1010 | 1011 |
| 12 | 13 | 14 | 15 |
| 1100 | 1101 | 1110 | 1111 |

Complétez les équivalences suivantes entre un nombre écrit en binaire et sa valeur correspondante en hexadécimal :

|  |  |  |
| --- | --- | --- |
| Hexadécimal |  | Binaire |
| 0x0 | équivaut à | 0000b |
| 0x1 | équivaut à | 0001b |
| 0x2 | équivaut à | 0010b |
| 0x5D | équivaut à | 0101 1101b |
| 0xC8 | équivaut à | 1100 1000b |
| 0x3E27 | équivaut à | 0011 1110 0010 0111b |
| 0x4AB9F560 | équivaut à | 100 1010 1011 1001 1111 0101 0110 0000‬ |

## Exercice n°1.4 – Quel est votre âge en binaire ?

Pour convertir rapidement, par écrit, un nombre décimal en binaire, il suffit d’appliquer la méthode suivante. Par exemple, j’ai 43 ans. J’écris les puissances de 2 jusque celle qui dépasse ce nombre :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  |  |  |  |  |  |  |

Je « prends » la puissance de 2 juste en-dessous de celle qui dépasse, donc ici 32. J’écris 1 en-dessous du 32 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  | 1 |  |  |  |  |  |

43 moins 32 font 11. Il reste en d’autres termes 11 à « prendre ». Je ne peux pas prendre 16, car 16 est plus grand que 11, je peux prendre 8, j’écris 1 en-dessous du 8 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  | 1 |  | 1 |  |  |  |

11 moins 8 font 3. Il reste 3 à « prendre ». Je prends 2 et 1 qui font bien trois en écrivant :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  | 1 |  | 1 |  | 1 | 1 |

Et le nombre binaire équivalent à 43 est ainsi 101011.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  | 1 | 0 | 1 | 0 | 1 | 1 |

Avec cette méthode, je peux vérifier mon calcul en faisant la somme des puissances de 2 là où il y a des 1. Cela donne 32+8+2+1=43.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 32 | 16 | 8 | 4 | 2 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |

Quel est ton âge en binaire ? Détaille et vérifie ton calcul par écrit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 32 | 16 | 8 | 4 | 2 | 1 |
|  | 1 | 0 | 0 | 1 | 0 |

Inversement, pour convertir rapidement, par écrit, un nombre binaire en décimal, il suffit d’appliquer la méthode suivante. Par exemple, j’ai 10101 ans. Quel est mon âge en décimal ? Il y a 5 chiffres, j’écris les 5 premières puissances de 2 de droite à gauche :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 16 | 8 | 4 | 2 | 1 |
| 1 | 0 | 1 | 0 | 1 |

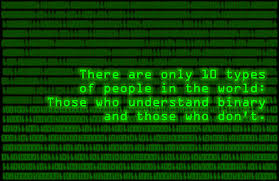
J’écris mon âge en binaire 10101 en-dessous des puissances. Ensuite, c’est simple, je compte la puissance de 2 s’il y a un 1 en-dessous.

Cela donne la somme de 16+4+1= 21 ans.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 16 | 8 | 4 | 2 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Vous pouvez maintenant comprendre quelques blagues d’informaticiens :





## Exercice n°1.5 – Trois bases pour un informaticien

Complétez le tableau suivant :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Décimal | 0 | 1 | 2 | 3 |
| Hexadécimal | 0 | 1 | 2 | 3 |
| Binaire | 0 | 1 | 10 | 11 |
| Décimal | 4 | 5 | 6 | 7 |
| Hexadécimal | 4 | 5 | 6 | 7 |
| Binaire | 0100 | 101 | 110 | 111 |
| Décimal | 8 | 9 | 10 | 11 |
| Hexadécimal | 8 | 9 | A | B |
| Binaire | 1000 | 1001 | 1010 | 1011 |
| Décimal | 12 | 13 | 14 | 15 |
| Hexadécimal | C | D | E | F |
| Binaire | 1100 | 1101 | 1110 | 1111 |

## Exercice n°1.6 – *Color picker*

Allez à la page suivante : <https://www.google.com/search?q=color+picker> pour voir un *color picker* réalisé par Google.

Un *color picker* permet de trouver les valeurs de rouge, de vert et de bleu pour afficher une couleur à l’écran.

Selon Wikipédia : « Rouge, vert, bleu, abrégé en **RVB** ou en **RGB**, de l'anglais « *Red, Green, Blue* » est, des systèmes de codage informatique des couleurs, le plus proche du matériel. » <https://fr.wikipedia.org/wiki/Rouge_vert_bleu>.

Quelle est la plage de valeurs possibles pour chacune des trois couleurs primaires ?

|  |  |  |  |
| --- | --- | --- | --- |
|  | R | V | B |
| Valeur minimale | 0 | 0 | 0 |
| Valeur maximale en hexadécimal | FF | FF | FF |
| Valeur maximale en décimal | 255 | 255 | 255 |

On peut dire ainsi que chaque couleur primaire est codée sur combien d’octets ?

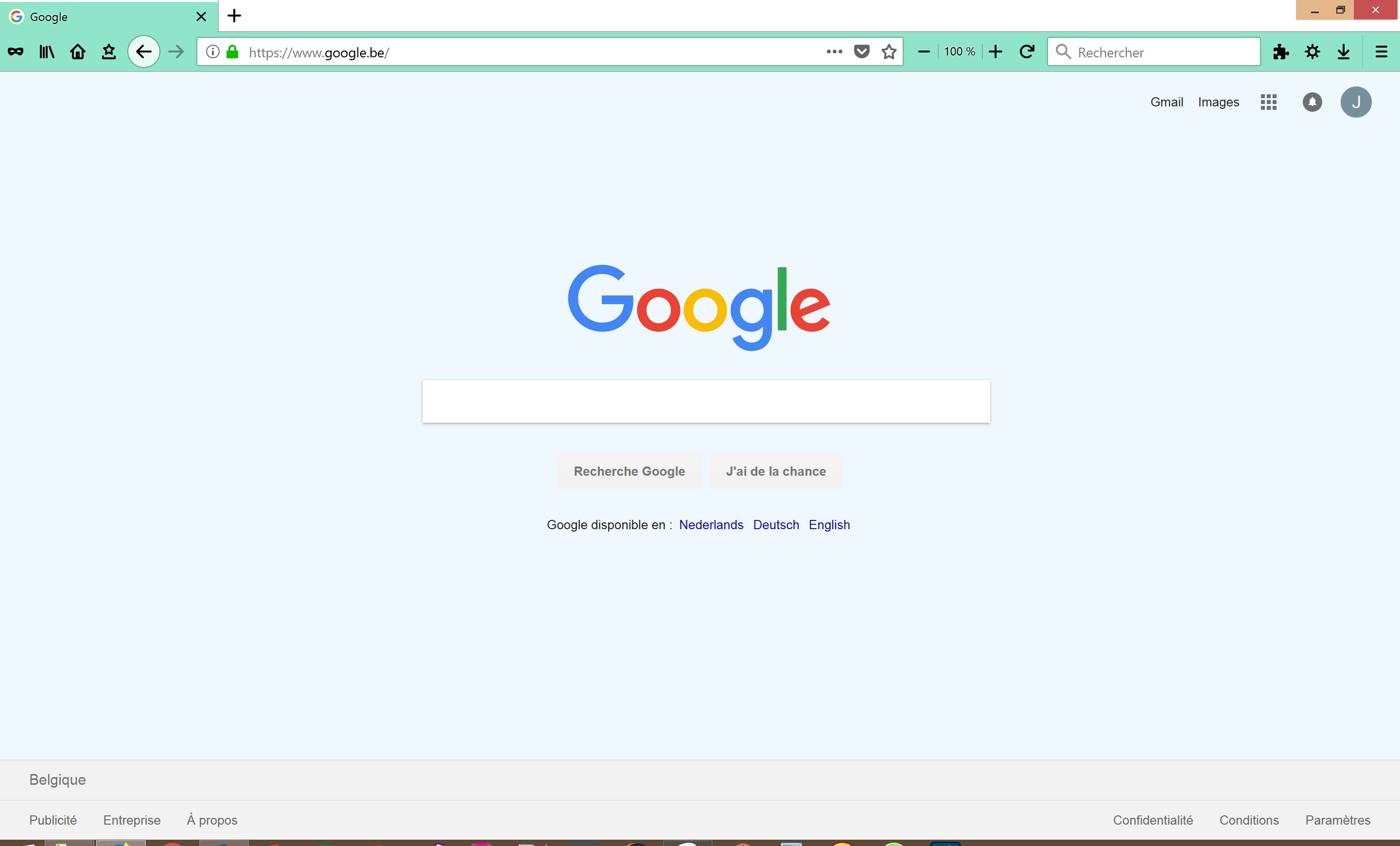
Quel est le nombre total de couleurs possibles selon ce système ?

Dans une feuille de styles .CSS, un programmeur a codé les valeurs RVB suivantes, écrivez en anglais à quelle couleur reconnue dans tout navigateur la valeur RVB correspond :

|  |  |
| --- | --- |
| Attributs : valeurs dans un .CSS | Couleurs |
| background-color: #000000 | Noir |
| background-color: #FF0000 | Rouge |
| background-color: #00FF00 | Vert |
| background-color: #0000FF | Bleu foncé |
| background-color: #FFFFFF | Blanc cassé |

Pour vous aider : <https://www.w3schools.com/cssref/css_colors.asp>.

Exercice de « hacking » CSS : allez sur le site de Google et changez *LIVE* la feuille de style dans le navigateur pour afficher un fond en « aliceblue ».



Cette modification n’est pas permanente, elle disparaît dès le rafraichissement du site.