REP Solution Interactive
Test d’aptitudes – Encodeur Base64
Révision 2

Test d'aptitudes – Encodeur Base64

1. Objectif

L'objectif de ce test est de valider les aptitudes d'analyse, de programmation et d'optimisation du candidat en réalisant une librairie d'encodage « Base64 ».

Ce test a été choisi parce que l'encodage Base64 est logiquement simple mais offre une bonne opportunité d'optimisation, ce qui permet à un programmeur de se démarquer en démontrant ses compétences générales en programmation et sa connaissance de l'environnement .NET.

2. Critères d'évaluation

2.1 Exactitude du résultat et respect des spécifications

Le critère de base est l'exactitude du résultat. Le test sera considéré comme échoué si la librairie soumise ne retourne pas le résultat escompté pour une variété de valeurs soumises en paramètre ou ne respecte pas les spécifications décrites dans ce document.

2.2 Performance

Le travail sera ensuite évalué sur la performance du résultat. Le candidat doit démontrer une expertise dans la manipulation de données et sa compréhension des éléments qui impactent la performance de ce traitement et des fonctionnalités du Framework .NET utilisées pour résoudre le problème.

La performance de la méthode d'encodage complète sera évaluée en l'appelant avec des paramètres de tailles très variées : Un jeu de données variant de 3 octets à près d'un million d'octets sera utilisé pour tester la performance de la librairie dans les différents scénarios d'utilisation.

3. Tâche à accomplir

Dans un projet .NET en C#, créer une classe statique « Base64 » exposant la méthode suivante :

```
public static string Encode(byte[] source)
```

- La méthode recevra en paramètre un tableau d'octets à encoder.
- Le tableau aura une longueur qui sera un multiple de 3 afin de simplifier le test.
- Le tableau d'octets reçus doit être encodé en suivant la procédure décrite à la section 4, et le résultat final doit être retourné dans une chaine de caractères (string).

3.1 Notes

Il va sans dire que le candidat doit effectuer l'encodage entièrement "à la main" sans faire appel à aucune librairie ou méthode d'encodage Base64 tiers ou incluse dans le Framework .NET.

Il est cependant conseillé au candidat de comparer la performance de sa librairie à celle fournie par le Framework .NET afin de juger de la performance de son travail.

4. Spécifications techniques

4.1 Description

L'encodage « Base64 » est typiquement utilisé pour représenter des données dans un media de stockage ou de transport qui ne supporte pas directement les données binaires (par exemple, pour la transmission d'un fichier attaché à un courriel). L'encodage utilise un alphabet transportable pour représenter des groupes de bits qui seront réassemblées lors du décodage.

4.2 Alphabet

Dans le cas spécifique du Base64, l'alphabet utilisé est de 64 caractères, ce qui permet de représenter 6 bits de données par caractère (2^6). L'alphabet suivant sera utilisé pour le test :

0 = A	8 = I	16 = Q	24 = Y	32 = g	40 = o	48 = w	56 = 4
1 = B	9 = J	17 = R	25 = Z	33 = h	41 = p	49 = x	57 = 5
2 = C	10 = K	18 = S	26 = a	34 = i	42 = q	50 = y	58 = 6
3 = D	11 = L	19 = T	27 = b	35 = j	43 = r	51 = z	59 = 7
4 = E	12 = M	20 = U	28 = c	36 = k	44 = s	52 = 0	60 = 8
5 = F	13 = N	21 = V	29 = d	37 = l	45 = t	53 = 1	61 = 9
6 = G	14 = O	22 = W	30 = e	38 = m	46 = u	54 = 2	62 = /
7 = H	15 = P	23 = X	31 = f	39 = n	47 = v	55 = 3	63 = +

4.3 Encodage

Le processus d'encodage Base64 transforme chaque groupe de 6 bits soumis en un caractère représentatif de la valeur de ces 6 bits, tel que défini par l'alphabet utilisé.

Le tableau d'octets soumis doit donc être vu comme une longue séquence de bits qui seront traités par groupe de 6 pour créer une chaine de caractère jusqu'à ce que tous les bits aient été traités. Le premier caractère sera donc représentatif des 6 premier bits du premier octet, alors que le second caractère sera représentatif des 2 derniers bits du premier octet combinés aux 4 premiers bits du second octet, et ainsi de suite.

4.4 Exemple

À titre d'exemple, le tableau de 3 octets suivants :

```
byte[] source = { 0x6A, 0x77, 0xC4};
```

Ces 3 octets représentent la séquence de 24 bits suivante :

```
01101010 01110111 11000100
```

Ces 24 bits nous fournissent 4 groupes de 6 bits :

```
011010 100111 011111 000100
```

La valeur numérique de chacun de ces groupes de 6 bits sera donc :

```
26 39 31 4
```

Ce qui correspond, en référant à l'alphabet fourni, aux caractères suivants :

```
anfE
```

Donc, le tableau d'octets 0x6A, 0x77, 0xC4, une fois encodé en Base64, devient anfE.

La méthode d'encodage réalisée doit donc répéter ce processus qui transforme chaque groupe de 3 octets en 4 caractères encodés jusqu'à ce que tous les octets reçu en paramètre soient encodés :