Exercice 1 :

1) Adresse logique

Les pages font 64 ko, l’offset est donc codé sur 16 bits.  
Comme le bus d’adresse fait 24 bits, l’adresse virtuelle sera codée sur 24 bits.  
Le numéro de page sera donc codé sur 24-16 = 8 bits.

Pour récapituler : offset sur 16bits puis numéro de page sur 8.

2) Pages logiques

Le numéro de page est codé sur 8 bits il y a donc 2^8 = 256 pages.

3) Descripteur de pages

Taille table pages = nb pages virt \* (taille bit de validité + taille numéro pages phys)

Nb de pages = 512/64 = 8. Le numéro de page est donc codé sur 8 bits.

Taille table pages = 256 \* ( 1 + 3 ) = 1024 bits soit 1 Kibibit.

4) Pages logiques dans la mémoire

(Je ne comprends pas bien la question, pour moi une page logique pointe juste vers une page physique et n’est pas « contenue » dans la mémoire centrale)

Les pages logiques ayant leur bit de validation à 1 sont : 1, 5, 8, 9, 13, 3E.

5) Processeur génère l’adresse logique

Dans l’adresse logique $15000 l’offset est $5000 et le numéro de page logique est $1.  
La page physique correspondant à la page logique $1 et $5.  
L’adresse physique sera donc $55000.

Dans l’adresse logique $3E0000 l’offset est $0 et le numéro de page logique est $3E.  
La page physique correspondant à la page logique $3E et $1.  
L’adresse physique sera donc $10000.

6) Simulation programme accès mémoire

Comme on simule un programme on doit faire attention à la taille des instructions ; ici elles sont codées sur 32 bits. On fera donc avancer le PC de 4octets à chaque instruction.

De $1 5000 à $1 FFFC : succès, le programme accèdera à la page physique 5.

$2 0000 : échec la page virtuelle à son bit de validité à zéro.

*(À partir de là je ne suis pas trop sûr. Je ne comprends pas bien quelle page physique on devrait mettre à la page 2. Comment le processeur ferait-il la correspondance entre les pages virtuelles et physique ? Il y a bien une continuité dans le programme, on ne peut pas continuer l’exécution sur n’importe quelle page.  
Certains ont utilisé le compteur de vieillissement comme pour la mémoire cache mais la encore ca ne fait pas beaucoup de sens pour moi. Admettons que cela soit le cas, selon la disposition des compteurs de vieillissement le programme aurait une exécution différente (on pourrait stocker la page physique 3 comme la 7 à la page virtuelle 2).  
Aussi si on ne peut avoir que chaque page physique chargée simultanément dans le mémoire, je ne vois pas trop l’intérêt d’avoir 256 pages virtuelles.  
Dans la suite de l’exercice je continuerais en utilisant le compteur de vieillissement)*

Comme la page physique $0 n’est pas chargée, on la fait correspondre avec la page logique $2.

De $2 0004 à $2 3000 : Succès

Saut à $4F E000 : Page physique $3 pas chargée, on la fait correspondre avec la page logique $4F.

De $4F E004 à $4F F500 : Succès

Accès à $7 F770 : Échec, plus vielle page physique : $4, On la fait correspondre à la page logique $7.  
Accès de $7 F774 à $8 0880 : Succès

De $4F F504à $4F FFFC : Succès

$50 000 : Échec, plus vielle page physique : $2, On la fait correspondre à la page logique $50.

$50 004 : Succès

Ret à 2 3004 : Succès

De $2 3008 à 2 FFFC : Succès

$3 0000 : Échec, plus vielle page physique : $1, On la fait correspondre à la page logique $3.

De $3 0004 à $3 6600 : Succès

Saut à $4F E000 : Succès

De $4F E004 à $4F F500 : Succès

Accès de $7 F770 à $8 0880 : Succès

De $4F F504à $50 004 : Succès

Ret à 2 3004 : Succès

