問2　仮想記憶方式（ソフトウェア）　（H25春-FE 午後問2）

【解答】
［設問1］　a－オ，b－ウ
［設問2］　エ
［設問3］　c－ウ，d－エ

【解説】
本問は，仮想記憶方式の一つであるページング方式に関する基本的な問題である。午前問題対策としても，仮想記憶方式に関する知識はあるはずである。その理解度に多少の不安があったとしても，問題文に沿って考えれば，ほぼ正解を導くことができる内容である。

仮想記憶方式に関して，問題文の繰返しであるが復習してみよう。

実行するプログラム全体を主記憶上に格納することができるのであれば，それにこしたことはない。しかし，単価の高い主記憶の容量には，ある程度の限りがある。そこで，仮想記憶方式では，主記憶よりも単価の安い大容量の補助記憶装置を，論理的な記憶領域（仮想記憶）として用いる。仮想記憶上に主記憶の容量を超えるプログラムを格納し，OS は論理的な記憶領域（仮想記憶）上のアドレスを提供する。プログラムの実行は，その時点で実行に必要な部分を主記憶上に移し実行することになるが，OS は，仮想記憶上のアドレス（仮想アドレス）と主記憶上のアドレス（物理アドレス）とを対応付けて管理する。仮想アドレスから物理アドレスに変換することを，動的アドレス変換という。

プログラムの実行では，プログラム全体，全てのメモリ領域を同時に必要とするわけではない。仮想記憶方式とは，その時点で必要とされる部分だけを主記憶に置き，その他は補助記憶装置（仮想記憶）に配置して，必要に応じて内容を入替えながらプログラムを実行できるようにする仕組みである。その入替えの単位には，プログラムの機能単位などのように意味のあるまとまりである「セグメント」と，そのようなまとまりとは無関係な固定長の「ページ」とがある。セグメントを単位とするのはセグメント方式，ページを単位とするのはページング方式と呼ぶ。本問は，ページング方式に関する問題である。

ページング方式では，仮想アドレス空間と物理アドレス空間のそれぞれをページと呼ぶ固定長の領域に分割しておき，ページ単位でアドレス空間を管理する。仮想アドレス空間及び物理アドレス空間の各ページには，先頭から順に番号を付け，それぞれを仮想ページ番号，物理ページ番号と呼び，それらの対応はページテーブルで管理する。ページテーブルの要素の個数は仮想ページの個数と同じで，各要素が仮想ページの1ページに対応している。各要素の項目は，問題文中の図で示すように，対応する仮想ページの内容が物理アドレス空間に存在しているかどうかを示す存在ビット（1：あり，0：なし）と物理ページ番号である。

［設問1］
・空欄 a：ページング方式では，プログラムの実行に必要なページを主記憶に置くわけだが，実行過程で必要なページが主記憶に存在しない場合，すなわち物理アドレス空間に存在しない場合は，ページフォールトという割込みを発生させる。必要なページがなければ，プログラムを先に進めることはできないので，必要なページを仮想記憶から主記憶に読み込む（ページイン）ことになる。したがって，（オ）が正解である。
・空欄 b：ここで，もしも主記憶に空いている領域がなければ，主記憶上のいずれかのページを仮想記憶に追い出し（ページアウト）て必要な領域を確保してから，その場所に今必要なページを読み込むことになる。主記憶上のどのページをページアウトさせるかを選択するアルゴリズム（ページングアルゴリズム）は，幾つかある。問題文にあるように，FIFO（First In First Out）アルゴリズムは，置換え対象となるページを「ページインしてから最も時間が経過しているページ」とするものである。選択肢の（ア），（ウ）は次のようなアルゴリズムである。
　・（ア）LFU（Least Frequently Used）：直前の一定時間に最も参照回数の少ないページを置換え対象とするアルゴリズム
　・（ウ）LRU（Least Recently Used）：最後に使われてから最も長い時間が経過したページを置換え対象とするアルゴリズム
　以上のとおり，空欄 b の「参照されていない時間が最も長いページを置換え対象とするアルゴリズム」は LRU である。したがって，（ウ）が正解である。
　なお，選択肢（イ）の LIFO（Last In First Out）は，「最後に入れたものを最初に出す」というアルゴリズムである。これは，仮想記憶方式のページ置換え処理アルゴリズムとしてふさわしいとはいいがたい。プログラムの実行は，概ね局所的であり，一度使用・参照したものを，時を置かずして再度使用・参照する場合が多い。このため最後にページインしたページを，すぐにページアウトしてしまう方式は効率的ではない。

［設問2］
ページ置換えの具体的な処理の正しい順番を考える問題である。
【処理の単位】を見ると，次のように，前後関係が確定できる項目があり，正解候補を限定できる。
・②でページアウトさせるページを決定しないと，①のページアウトはできない。つまり，②の後に①がなければならない。したがって，（ア）と（イ）は正しくない。
・①でページアウトさせる前に，③のページインをすることはできない。つまり，①の後に③がなければならない。したがって，（オ）と（カ）は正しくない。
更に，④から⑦はページテーブルへの値の設定であるが，「⑥　ページアウトしたページに対応するページテーブルの要素の物理ページ番号を設定する」は，明らかに正しい処理とはいえない。ページアウトしたならば物理アドレス空間にないのだから，物理ページ番号を設定する意味がないのである。したがって，⑥を含む（ア），（ウ），（オ），（カ）は正しくない。
以上のことにより，正解は（エ）で次の順番である。

②　ページ置換えアルゴリズムによって，物理アドレス空間からページアウトするページを決定する。
①　退避させるページをページアウトする。
④　ページアウトしたページに対応するページテーブルの要素の存在ビットを0にする。
③　実行に必要なページをページインする。
⑦　ページインしたページに対応するページテーブルの要素の物理ページ番号を設定する。
⑤　ページインしたページに対応するページテーブルの要素の存在ビットを1にする。

［設問3］
FIFO アルゴリズムを採用した場合のページフォールトの発生回数が問われている。ページ置換え状況をシミュレートしてみる。
まず，物理ページの個数が3の場合である。図のように、物理アドレス空間の物理ページを一つのマス目とイメージし，3段のマス目で表している。

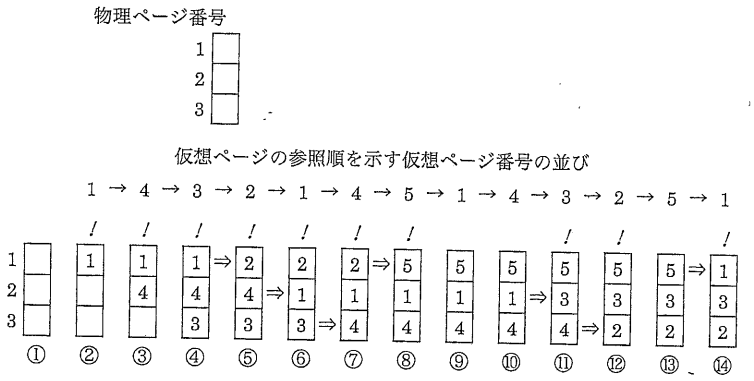

図1　物理ページ個数3の場合の状況

①　「プログラムの実行開始時点では，物理アドレス空間にはどのページも存在していない」ということから，マス目の中身は空白である。
②　最初の参照は「仮想ページ1」。物理アドレス空間には存在していない（空っぽ）ので，ページフォールトが発生。「/」で表している。物理ページ1の位置に「仮想ページ1」を格納。
③　「仮想ページ 4」の参照。物理アドレス空間には存在していないので，ページフォールトが発生。物理ページ2の位置に「仮想ページ4」を格納。
④　「仮想ページ 3」の参照。物理アドレス空間には存在していないので，ページフォールトが発生。物理ページ3の位置に「仮想ページ3」を格納。
⑤　「仮想ページ 2」の参照。物理アドレス空間には存在していないので，ページフォールトが発生。現在物理アドレス空間にある三つのうち最も早くに格納した物理ページ番号1にあるページをページアウトし，そこに「仮想ページ2」を格納。
⑥～⑧　参照しようとするページは，ことごとく物理アドレス空間にないので，次々とページフォールトが発生する。FIFO の順番に従って，ページの置換えを行う。
⑨，⑩　参照しようとするページが物理アドレス空間に存在するので，ページフォールトは発生しない。
⑪，⑫　参照しようとするページが物理アドレス空間にないので，ページフォールトが発生する。ページの置換えを行う。
⑬　参照しようとするページが物理アドレス空間に存在するので，ページフォールトは発生しない。
⑭　参照しようとするページが物理アドレス空間にないので，ページフォールトが発生する。ページの置換えを行う。
以上の結果，ページフォールト（「/」の箇所）の発生は，10 回である。したがって，空欄 c の答えは（ウ）である。
次に，物理ページの個数が4の場合。これも同様にシミュレートしてみると，次の図のようになる。ページフォールトの発生は，11 回。したがって，空欄 d の答えは（エ）である。

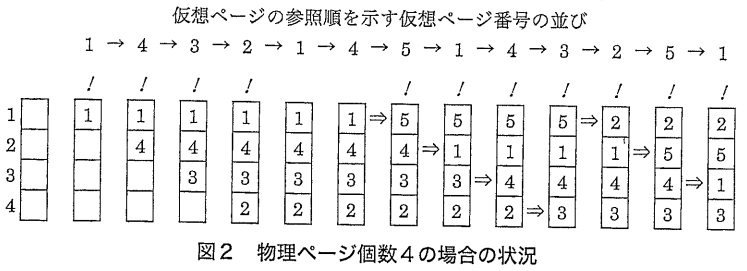

図2　物理ページ個数4の場合の状況

感覚的には，物理ページの個数が多ければ，ページフォールトの発生回数は少なくなるように思えるが，結果は逆であった。参照順序が異なれば，あるいはページ置換えアルゴリズムが異なれば別の結果になったかもしれないが，プログラムの実行状態を完全に予測することは不可能である。置換えアルゴリズムの決定には，ある程度の割切りが必要である。

ところで，本問の例では参照する仮想ページの個数が5，物理ページの個数は3か4であったが，最初に物理アドレス空間が空の状態でページフォールトが発生するのはいたしかたのないことであるが，13 回の参照で 10 回，11 回のページフォールトが発生している。一般に，必要としているメモリ容量（プログラム全体のサイズ）に対して主記憶で使えるメモリ容量が大きく不足していると，プログラムの実行に伴ってページング（ページイン・ページアウト）が多発することになる。ページングのためにも CPU などの資源は必要となる。このようにページングが多発してしまうと，他のプログラムに資源を割り当てられず，システム全体の性能を悪化させることになる。ページフォールトが多発しシステムの動きが悪くなる状況を，スラッシングという。