

問題 4 次のプログラムの説明および擬似言語の記述形式の説明を読み、各設問に答えよ。

[プログラムの説明]

J スーパーでの商品ごと月別売上高の集計表を出力するため、ファイルを集計する処理である。なお、月別売上高の集計表は商品コードの昇順に出力する。

レシートデータのレコード形式は以下のとおりであり、レシート番号の昇順に並んでいる順編成ファイルである。

レシート番号	日時				商品 1		...	商品 n	
	年	月	日	時間	商品コード	個数		商品コード	個数

商品マスタのレコード形式は以下のとおりであり、索引編成ファイルである。

商品コード	商品名	単価
-------	-----	----

売上ファイルのレコード形式は以下のとおりであり、順編成ファイルである。このファイルには、複数年分のレコードが存在する。

年	月	商品コード	商品名	個数	売上金額
---	---	-------	-----	----	------

集計ファイルのレコード形式は以下のとおりであり、順編成ファイルである。

年	月	商品コード	商品名	個数計	金額計
---	---	-------	-----	-----	-----

各処理の内容は、次のとおりである。

表 処理内容

処 理	処 理 内 容
分割・ 計算処理	レシートファイルを商品ごとのレコードに分割するとともに、商品マスタより商品名・単価を読み込み、売上金額を求め、売上ファイルを作成する。
整列処理	売上ファイルを並べ換えて、整列済み売上ファイルを作成する。売上ファイルと整列済み売上ファイルのレコード形式は同じである。
集計処理	商品ごとの月別売上個数と売上金額を集計し、集計ファイルを作成する。
編集・ 出力処理	集計ファイルを、商品ごとの月別売上高の集計表の形式に編成し出力する。

商品ごと月別売上高の集計表を出力するための流れは，次のようになる。
売上データは複数年分存在するものとする。

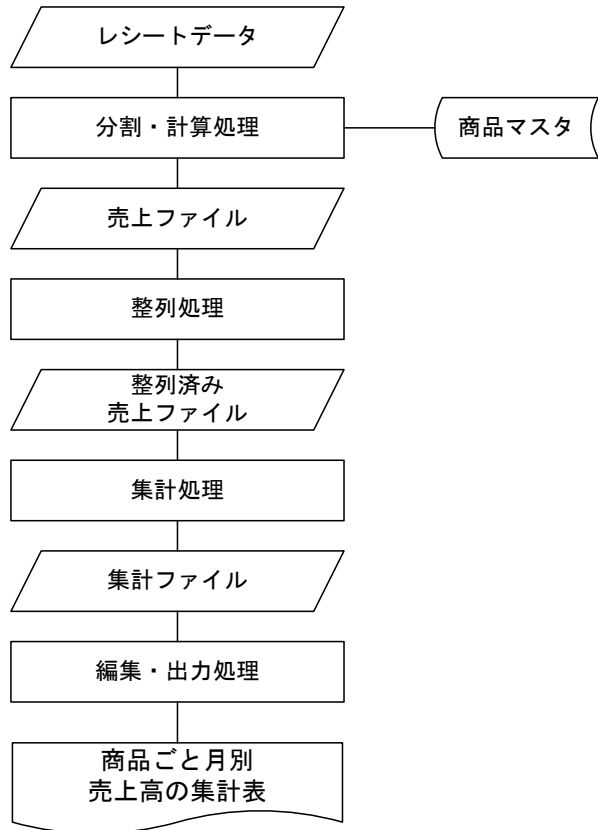


図 商品ごと月別売上高の集計表の作成処理の流れ図

＜設問 1＞ 整列処理における並べ替えのキーとして必要な最低の項目を解答群から選び，(1)に答えよ。なお，複数の項目がある場合は，左から順に並べ替えのキーの優先順位を示す。

(1) の解答群

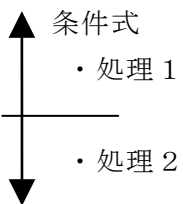
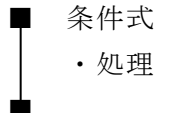
ア．商品コード

イ．商品コード，年，月

ウ．商品コード，月

エ．商品コード，年

[擬似言語の記述形式の説明]

記述形式	説明
○	手続き，変数などの名前，型などを宣言する
・変数 ← 式	変数に式の値を代入する
/* 文 */	注釈を記述する
	<p>選択処理を示す。</p> <p>条件式が真の時は処理 1 を実行し，偽の時は処理 2 を実行する。</p>
	<p>前判定繰り返し処理を示す。</p> <p>条件式が真の間，処理を実行する。</p>

<設問 2> 次の集計処理のプログラム中の に入れるべき適切な字句を解答群から選べ。なお，プログラム中の※の部分は，問題の関係上表示していない。

[プログラム]

- プログラム名: **Shukei()**
- 整数型: **kosu, urikin, kosu_ttl, urikin_ttl, nen, tuki, old_nen, old_tuki**
- 文字型: **scode, sname, old_scode, old_sname**
- ファイル: **u-file, s-file**
- 手続き: **FileRead(u-file, nen, tuki, scode, sname, kosu, urikin)**
 - /* 売上ファイル (u-file) から 1 レコードずつ読み込み */**
 - /* 年を nen, 月を tuki, 商品コードを scode, 商品名を sname, */**
 - /* 個数を kosu, 売上金額を urikin に設定する。 */**
 - /* ファイルの最後を示す識別子を入力した場合は, nen, tuki */**
 - /* scode に最高値が設定される。 */**
- 手続き: **FileWrite(s-file, ※)**
 - /* 年, 月, 商品コード, 商品名, 個数計, 金額計を設定し, */**
 - /* 集計ファイル (s-file) に 1 レコード書き出す。 */**
 - /* 引数の「※」は問題で使用するため表示しない。 */**

```

・ファイルを開く
・FileRead(u-file, nen, tuki, scode, sname, kosu, urikin)
■ scode ≠ 最高値
  ・old_scode ← scode
  ・ (2)
    ■ (3)
      ・old_nen ← nen
      ・old_tuki ← tuki
      ・kosu_ttl ← 0
      ・urikin_ttl ← 0
      ■ (3) かつ (4)
        ・kosu_ttl ← kosu_ttl + kosu
        ・urikin_ttl ← urikin_ttl + urikin
        ・FileRead(u-file, nen, tuki, scode, sname, kosu, urikin)
      ■
        ・FileWrite(s-file, (5))
    ■
  ■
  ・ファイルを閉じる

```

(2) の解答群

- | | |
|----------------------|---------------|
| ア. old_sname ← sname | イ. scode ← 0 |
| ウ. tuki ← 最高値 | エ. urikin ← 0 |

(3) , (4) の解答群

- ア. old_scode = scode
 イ. old_scode ≠ scode
 ウ. old_nen = nen
 エ. old_nen ≠ nen
 オ. old_nen = nen かつ old_tuki = tuki
 カ. old_nen ≠ nen または old_tuki ≠ tuki

(5) の解答群

- ア. nen, tuki, scode, sname, kosu, urikin
 イ. nen, tuki, scode, sname, kosu_ttl, urikin_ttl
 ウ. old_nen, old_tuki, old_scode, old_sname, kosu_ttl, urikin_ttl
 エ. old_nen, old_tuki, old_scode, sname, kosu, urikin