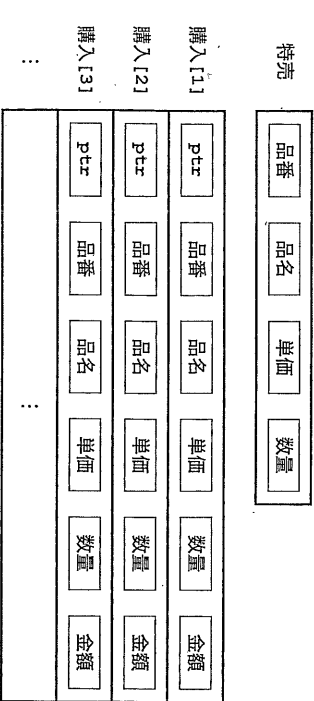


【解答】

【設問1】 aーウ, bーオ, cーカ, dーア
【設問2】 eーイ, fーフ, gーウ

【解説】

食料品店での選択型特売の値引き処理に関するアルゴリズムの問題である。プログラムでは構造型変数や構造型配列を用いたリストを扱っており、[プログラムの説明]に書かれた内容から各変数や配列の役割、使用方法が理解できたかどうか、リストの操作やマッシュング(突合せ処理)について理解できていたかどうかによって、難易度の感じ方が異なったと思われる。解答を書くためには、例として示されたデータや実行結果を利用して[プログラムの説明]に書かれた内容を理解し、プログラム中の各処理の意味を考えることが大切である。難易度としては普通からやや難しいといえる。最初に構造型変数、構造型配列について確認する。構造型とは複数の型の変数を一つにまとめて扱えるようにした型であり、構造型配列は構造型のデータを要素として格納する配列である。変数「特売」や配列「購入」は次図のような構造であり、項目(メンバ)を参照するには、特売.品番や購入[1].品番と記述する。



なお、配列「購入」はメンバptrによって品番の昇順にレコードをたどるリストである。リストの先頭は変数ptr起点に格納された値を添字とする要素であり、図1の場合、購入[2]の要素(品番111のレコード)となる。購入[2].ptrの値(1)は次に続くレコードが格納されている添字を示しており、購入[1]の要素(品番222のレコード)が2番目のレコードとなる。以降同様にptrをたどると、購入[2]→購入[1]→購入[4]→購入[5]→購入[3]とレコードは並ぶ。

ptr		ptr	品番	品名	単価	数量	金額
ptr 起点	購入[1]	4	222	B 社緑茶 500ml	140	5	700
2	購入[2]	1	111	A 社牛乳 1000ml	200	2	400
購入行数	購入[3]	0	335	C 社めんつゆ 300g	150	1	150
5	購入[4]	5	224	B 社麦茶 500ml	140	2	280
	購入[5]	3	333	C 社うどん 2 食入	180	2	360

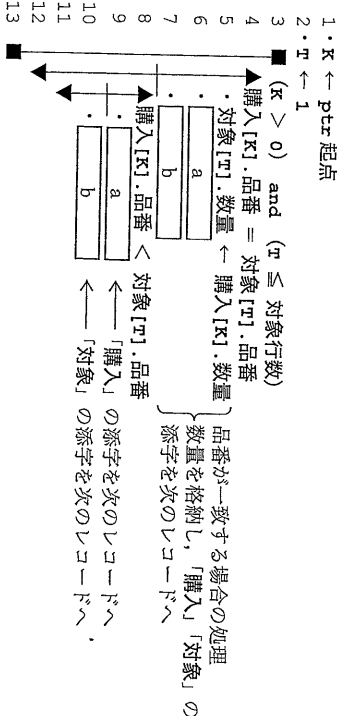
※購入[3].ptrの値の0は、最後のレコードであることを示す。

次にプログラムについて確認する。

(プログラム)

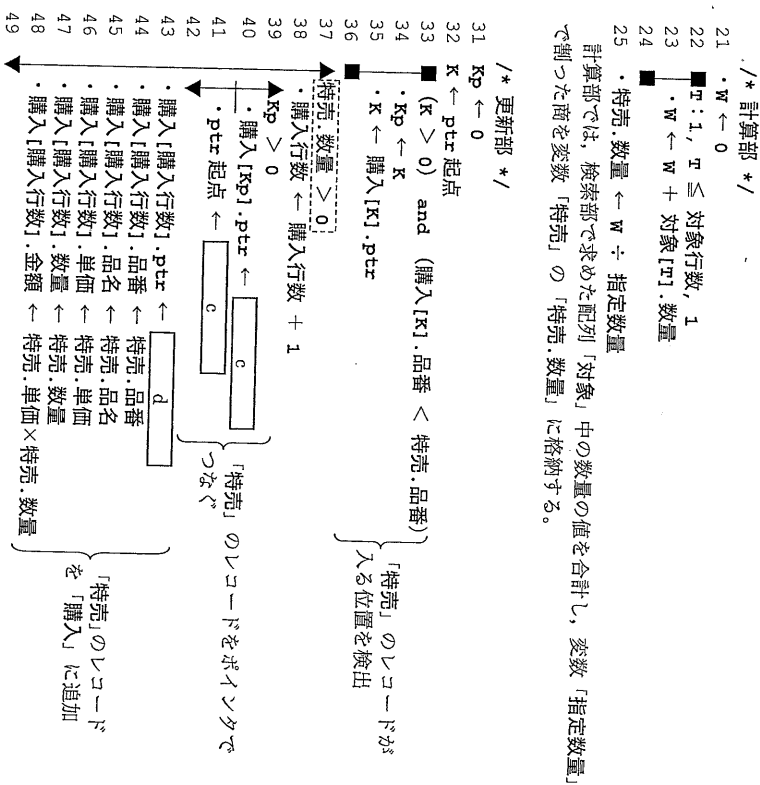
○整数型: K, Kp, T, W

/* 検索部 */



検索部では、配列「対象」中の品番と同じ品番をもつレコードが配列「購入」中にあれば、「購入」中の該当レコードの数量を、「対象」中の数量に格納する。「購入」はメンバptrによって品番の昇順にレコードをたどることができ、「対象」は配列の先頭から品番の昇順にレコードが格納されている。そこで、検索部では品番をキーにマッシュング処理を行う。マッシュングとは、あるファイルの内容を基にして、別のファイルの内容を書き換えたり、削除したりする処理である。データはあらかじめキーとなる項目の昇順又は降順に並んでおり、キー項目の値を比較しながら対応する処理を行う。本問の場合は、配列「購入」と「対象」の品番が一致(購入[K].品番＝対象[T].品番)した場合に「対象」の数量に「購入」の数量を格納する処理を行う。品番が一致しない場合の処理はないので、該当する配列の添字を更新して一致する品番を探す。このとき、レコードは品番の昇順に並んでいるので、購入[K].品番<対象[T].品番の場合には、購入[K]の品番は「購入」側だけに存在、購入[K].品番>対象[T].品番の場合には、対象[T]の品番は「対象」側だけに存在することになる。よって、前者の場合は、「購入」の添字を更新して次のレコードの格納位置へ、後者の場合は、「対象」の添字に1加算して次のレコードの格納位置を指すようにする必要がある。図1、図2のデータ例を用いてマッシュング処理を行うと、次のようになる。

購入	対象	処理内容
111	222	「購入」を次のレコードへ
222	=	「購入」の数量を「対象」の数量に格納
224	>	「購入」を次のレコードへ
224	=	「購入」の数量を「対象」の数量に格納
224	=	「購入」の数量を「対象」の数量に格納
383	なし	「対象」のレコードがなくなったため処理終了



更新部では、変数「特売」中の数量の値が1以上であれば、「特売」のレコードを配列「購入」中の適切な位置(品番の昇順)に追加する。

【設問1】

・空欄 a, b: 行番号3～13の繰返し処理の中には、配列「購入」「対象」の添字であるKとTを更新する処理がないので、空欄a, bでその処理を行う必要があることが分かる。既に確認したように、マッシュング処理において二つの配列のキー項目が一致した場合は、「購入」中の該当するレコードの数量を「対象」中の数量に格納する処理を行い、それぞれ次のレコードが処理の対象となるように二つの配列の添字を更新する。また、購入[K].品番<対象[T].品番となる場合は「購入」側が、購入[K].品番>対象[T].品番となる場合は「対象」側が次のレコードになるように添字を更新する。「購入」はリストであるため、次のレコードは現レコードである購入[K].ptrに格納された値を添字とする要素であり、「対象」の次のレコードは現在のレコードの一つ後ろ(T+1)に格納された要素である。そして、行番号8の条件式より、空欄aは「購入」の添字を更新、空欄bは「対象」の添字を更新する処理であることが分かる。したがって、空欄aは(ウ)、空欄bは(オ)が正解である。

・空欄 c, d: 行番号39の条件式Kp>0が真の場合は購入[Kp].ptrに、偽の場合はptr起点に、それぞれ空欄cの値を代入する。購入[K].ptrやptr起点に値を代入することから、ここではポインタをつなぐ処理を行うことが分かる。変数Kpは行番号31で初期値として0をセットしている。そして、Kpの値が0より大きくなるのは、行番号33～36の繰返し処理を実行し、行番号34でKpの値を更新した場合である。一方、Kpの値が初期値の0のままになるのは、行番号38～36の繰返し処理を一度も実行しない場合である。仮にKp>0の場合で考えると、行番号38～36の繰返し処理を実行することになるので、変数Kpには配列「購入」の比較対象のレコードの添字(Kの値)が、変数Kにはその次に続くレコードが格納されている添字(購入[K].ptrの値)が格納される。図1、図2のデータ例の場合、Kの初期値は2であり、1回目の繰返し処理ではKp=2, K=1となる。繰返しの条件は、K>0と購入[K].品番<特売.品番の複合条件であるが、それぞれの意味を考えると、K>0のKは配列「購入」の添字であり、Kには「購入」の購入[K].ptrの値を代入する代り、最も大きい品番をもつレコード(最後のレコード)のptrの値(0)が代入されたときに条件を満たさなくなる。つまり、Kが0になるのは、最後のレコードに到達した場合なので、条件式K>0はレコードの終わりをチェックしていることが分かる。一方、購入[K].品番<特売.品番は、配列「購入」中の品番と「特売」の品番を比べ、「特売」の品番が「購入」の品番よりも大きくなると条件を満たさなくなる。このときKpは条件を満たさなくなったレコードの前のレコードの格納位置を、Kは条件を満たさなくなったレコード自身の格納位置を指している。このことから、「特売」のレコードを「購入」のどの位置に挿入するかを探していることが分かる。これは「購入」が品番の昇順にレコードをつないだりリストであるため、追加する「特売」のレコードも品番の昇順になる位置に挿入する必要があるからである。したがって、行番号38～36の繰返し処理は、「特売」のレコードの挿入位置が見つかるまで繰り返すことになり、図1、図2のデータ例の場合、繰返し処理を抜けたときのKp, Kの値は、それぞれ4, 5となる。

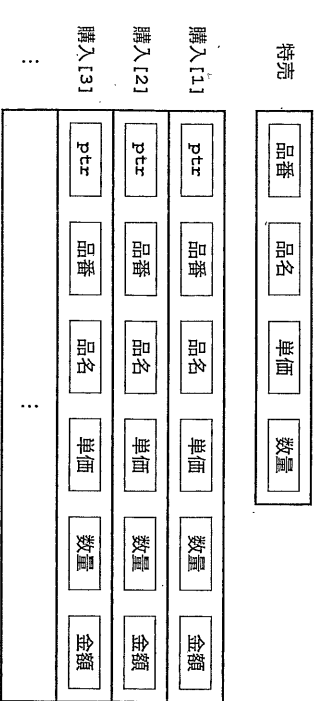
行番号48～48の処理を見ると「特売」のレコードを格納する位置(添字)は変数「購入行数」になっている。次図(「購入」を品番の順に表示)からも分かるように、「特売」のレコードは品番224と383のレコードの間に入ることになり、購入[4].ptrを「特売」のレコードが格納されている6(購入行数)に変更する必要がある。よって、空欄cには購入行数が入る。また、空欄dでは、購入[購入行数].ptrに値を代入しているが、ここでは「特売」のレコードのptrにその次にくるレコードの格納位置をセットする。そして、その格納位置は変数Kの値である。したがって、空欄cは(カ)、空欄dは(ア)が正解である。

【解答】

【設問1】 aーウ, bーオ, cーカ, dーア
【設問2】 eーイ, fーフ, gーウ

【解説】

食料品店での選択型特売の値引き処理に関するアルゴリズムの問題である。プログラムでは構造型変数や構造型配列を用いたリストを扱っており、[プログラムの説明]に書かれた内容から各変数や配列の役割、使用方法が理解できたかどうか、リストの操作やマッシュング(突合せ処理)について理解できていたかどうかによって、難易度の感じ方が異なったと思われる。解答を書くためには、例として示されたデータや実行結果を利用して[プログラムの説明]に書かれた内容を理解し、プログラム中の各処理の意味を考えることが大切である。難易度としては普通からやや難しいといえる。最初に構造型変数、構造型配列について確認する。構造型とは複数の型の変数を一つにまとめて扱えるようにした型であり、構造型配列は構造型のデータを要素として格納する配列である。変数「特売」や配列「購入」は次図のような構造であり、項目(メンバ)を参照するには、特売.品番や購入[1].品番と記述する。



なお、配列「購入」はメンバptrによって品番の昇順にレコードをたどるリストである。リストの先頭は変数ptr起点に格納された値を添字とする要素であり、図1の場合、購入[2]の要素(品番111のレコード)となる。購入[2].ptrの値(1)は次に続くレコードが格納されている添字を示しており、購入[1]の要素(品番222のレコード)が2番目のレコードとなる。以降同様にptrをたどると、購入[2]→購入[1]→購入[4]→購入[5]→購入[3]とレコードは並ぶ。

ptr		ptr	品番	品名	単価	数量	金額
ptr 起点	購入[1]	4	222	B 社緑茶 500ml	140	5	700
2	購入[2]	1	111	A 社牛乳 1000ml	200	2	400
購入行数	購入[3]	0	335	C 社めんつゆ 300g	150	1	150
5	購入[4]	5	224	B 社麦茶 500ml	140	2	280
	購入[5]	3	333	C 社うどん 2 食入	180	2	360

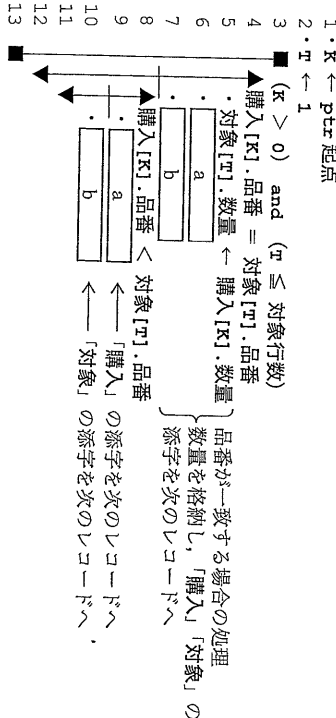
※購入[3].ptrの値の0は、最後のレコードであることを示す。

次にプログラムについて確認する。

【プログラム】

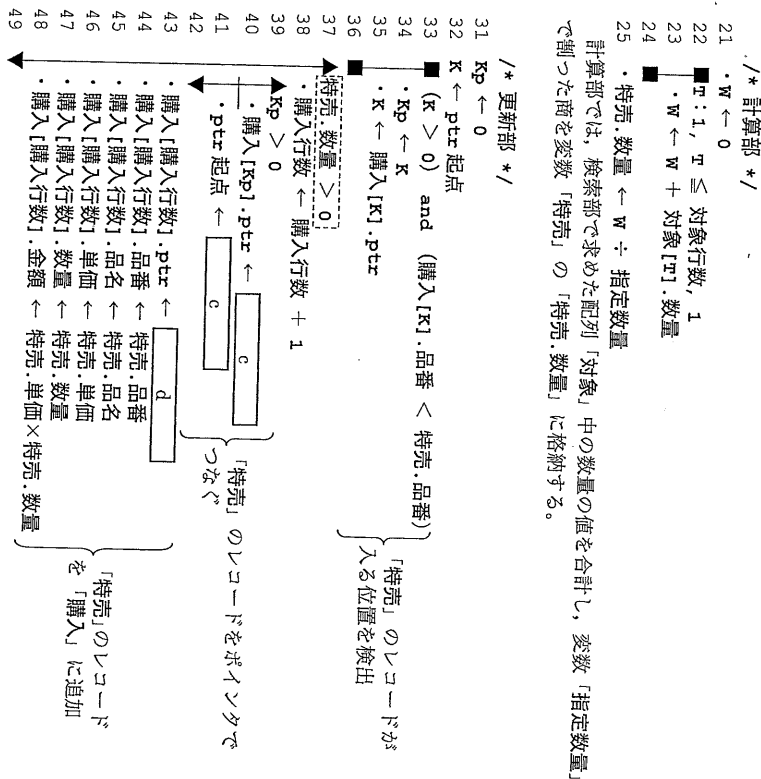
○整数型: K, Kp, T, W

/* 検索部 */



検索部では、配列「対象」中の品番と同じ品番をもつレコードが配列「購入」中にあれば、「購入」中の該当レコードの数量を、「対象」中の数量に格納する。「購入」はメンバptrによって品番の昇順にレコードをたどることができ、「対象」は配列の先頭から品番の昇順にレコードが格納されている。そこで、検索部では品番をキーにマッシュング処理を行う。マッシュングとは、あるファイルの内容を基にして、別のファイルの内容を書き換えたり、削除したりする処理である。データはあらかじめキーとなる項目の昇順又は降順に並んでおり、キー項目の値を比較しながら対応する処理を行う。本問の場合は、配列「購入」と「対象」の品番が一致(購入[K].品番＝対象[T].品番)した場合に「対象」の数量に「購入」の数量を格納する処理を行う。品番が一致しない場合の処理はないので、該当する配列の添字を更新して一致する品番を探す。このとき、レコードは品番の昇順に並んでいるので、購入[K].品番<対象[T].品番の場合には、購入[K].品番は「購入」側だけに存在、購入[K].品番>対象[T].品番の場合には、対象[T].品番は「対象」側だけに存在することになる。よって、前者の場合は、「購入」の添字を更新して次のレコードの格納位置へ、後者の場合は、「対象」の添字に1加算して次のレコードの格納位置を指すようにする必要がある。図1、図2のデータ例を用いてマッシュング処理を行うと、次のようになる。

購入	対象	処理内容
111	222	「購入」を次のレコードへ
222	=	「購入」の数量を「対象」の数量に格納
224	>	「購入」, 「対象」ともに次のレコードへ
224	=	「購入」の数量を「対象」の数量に格納
224	>	「購入」, 「対象」ともに次のレコードへ
383	なし	「対象」のレコードがなくなったため処理終了



更新部では、変数「特売」中の数量の値が1以上であれば、「特売」のレコードを配列「購入」中の適切な位置(品番の昇順)に追加する。

【設問1】

・空欄a, b: 行番号3～13の繰返し処理の中には、配列「購入」「対象」の添字であるKとTを更新する処理がないので、空欄a, bでその処理を行う必要があることが分かる。既に確認したように、マッシュング処理において二つの配列のキー項目が一致した場合は、「購入」中の該当するレコードの数量を「対象」中の数量に格納する処理を行い、それぞれ次のレコードが処理の対象となるように二つの配列の添字を更新する。また、購入[K].品番<対象[T].品番となる場合は「購入」側が、購入[K].品番>対象[T].品番となる場合は「対象」側

が次のレコードになるように添字を更新する。「購入」はリストであるため、次のレコードは現レコードである購入[K].ptrに格納された値を添字とする要素であり、「対象」の次のレコードは現在のレコードの一つ後ろ(T+1)に格納された要素である。そして、行番号8の条件式より、空欄aは「購入」の添字を更新、空欄bは「対象」の添字を更新する処理であることが分かる。したがって、空欄aは(ウ)、空欄bは(オ)が正解である。

・空欄c, d: 行番号39の条件式Kp>0が真の場合は購入[Kp].ptrに、偽の場合はptr起点に、それぞれ空欄cの値を代入する。購入[K].ptrやptr起点に値を代入することから、ここではポインタをつなぐ処理を行うことが分かる。変数Kpは行番号31で初期値として0をセットしている。そして、Kpの値が0より大きくなるのは、行番号33～36の繰返し処理を実行し、行番号34でKpの値を更新した場合である。一方、Kpの値が初期値の0のままになるのは、行番号38～36の繰返し処理を一度も実行しない場合である。仮にKp>0の場合で考えると、行番号38～36の繰返し処理を実行することになるので、変数Kpには配列「購入」の比較対象のレコードの添字(Kの値)が、変数Kにはその次に続くレコードが格納されている添字(購入[K].ptrの値)が格納される。図1、図2のデータ例の場合、Kの初期値は2であり、1回目の繰返し処理ではKp=2, K=1となる。繰返しの条件は、K>0と購入[K].品番<特売.品番の複合条件であるが、それぞれの意味を考えると、K>0のKは配列「購入」の添字であり、Kには「購入」の購入[K].ptrの値を代入する代り、最も大きい品番をもつレコード(最後のレコード)のptrの値(0)が代入されたときに条件を満たさなくなる。つまり、Kが0になるのは、最後のレコードに到達した場合なので、条件式K>0はレコードの終わりをチェックしていることが分かる。一方、購入[K].品番<特売.品番は、配列「購入」中の品番と「特売」の品番を比べ、「特売」の品番が「購入」の品番よりも大きくなると条件を満たさなくなる。このときKpは条件を満たさなくなったレコードの前のレコードの格納位置を、Kは条件を満たさなくなったレコード自身の格納位置を指している。このことから、「特売」のレコードを「購入」のどの位置に挿入するかを探していることが分かる。これは「購入」が品番の昇順にレコードをつないだりリストであるため、追加する「特売」のレコードも品番の昇順になる位置に挿入する必要があるからである。したがって、行番号38～36の繰返し処理は、「特売」のレコードの挿入位置が見つかるまで繰り返すことになり、図1、図2のデータ例の場合、繰返し処理を抜けたときのKp, Kの値は、それぞれ4, 5となる。

行番号48～48の処理を見ると「特売」のレコードを格納する位置(添字)は変数「購入行数」になっている。次図(「購入」を品番の順に表示)からも分かるように、「特売」のレコードは品番224と383のレコードの間に入ることになり、購入[4].ptrを「特売」のレコードが格納されている6(購入行数)に変更する必要がある。よって、空欄cには購入行数が入る。また、空欄dでは、購入[購入行数].ptrに値を代入しているが、ここでは「特売」のレコードのptrにその次にくるレコードの格納位置をセットする。そして、その格納位置は変数Kの値である。したがって、空欄cは(カ)、空欄dは(ア)が正解である。