

平成24年度後期 情報検定

<実施 平成25年2月10日（日）>

プログラミングスキル

（説明時間 10：00～10：10）

（試験時間 10：10～11：40）

- ・ 試験問題は試験開始の合図があるまで開かないでください。
- ・ 解答用紙（マークシート）への必要事項の記入は、試験開始の合図と同時に行いますので、それまで伏せておいてください。
- ・ 試験開始の合図の後、次のページを開いてください。＜受験上の注意＞が記載されています。必ず目を通してから解答を始めてください。
- ・ 試験問題は、すべてマークシート方式です。正解と思われるものを1つ選び、解答欄の○をHBの黒鉛筆でぬりつぶしてください。2つ以上ぬりつぶすと、不正解になります。
- ・ 辞書、参考書類の使用および筆記用具の貸し借りは一切禁止です。
- ・ 電卓の使用が認められます。ただし、下記の機種については使用が認められません。

<使用を認めない電卓>

1. 電池式（太陽電池を含む）以外の電卓
2. 文字表示領域が複数行ある電卓（計算状態表示の一行は含まない）
3. プログラムを組み込む機能がある電卓
4. 電卓が主たる機能ではないもの
 - * パソコン（電子メール専用機等を含む）、携帯電話（PHS）、電子手帳、電子メモ、電子辞書、翻訳機能付き電卓、音声応答のある電卓、電卓付腕時計等
5. その他試験監督者が不適切と認めるもの

＜受験上の注意＞

1. この試験問題は33ページあります。ページ数を確認してください。
乱丁等がある場合は、手をあげて試験監督者に合図してください。
※問題を読みやすくするために空白ページを設けている場合があります。
2. 解答用紙（マークシート）に、受験者氏名・受験番号を記入し、受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は、採点されませんので十分注意してください。
3. 試験問題についての質問には、一切答えられません。自分で判断して解答してください。
4. 試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は、手をあげて試験監督者に合図してください。
5. 試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は、もう一度、受験番号・マーク・氏名が記載されているか確認して退出してください。なお、試験終了5分前の合図以降は退出できません。試験問題は各自お持ち帰りください。
6. 試験後にお知らせする合否結果（合否通知）、および合格者に交付する「合格証・認定証」はすべて、Web ページ（PC、モバイル）での認証によるデジタル「合否通知」、デジタル「合格証・認定証」に移行しました。
 - ①団体宛にはこれまでと同様に合否結果一覧ほか、試験結果資料一式を送付します。
 - ②合否等の結果についての電話・手紙等でのお問い合わせには、一切応じられませんので、ご了承ください。

<問題の構成>

必須問題 全員解答

問題 1 ～ 問題 4	2 ページ～16 ページ
-------------------------	--------------

選択問題 次の問題から 1 問選択し解答せよ。
(選択した問題は解答用紙「選択欄」に必ずマークすること)
※選択欄にマークがなく、解答のみマークした場合は採点を行いません。

・ C 言語の問題	18 ページ～21 ページ
・ 表計算の問題	22 ページ～26 ページ
・ アセンブラの問題	28 ページ～33 ページ

必須問題

問題 1 次のスタックに関する記述を読み、各設問に答えよ。

スタックとは、後入れ先出しを行うデータ構造で、スタックを操作する命令は PUSH と POP がある。

PUSH はスタックにデータを格納する命令で、図 1 のように、すでに格納されているデータに積み重ねるように格納する。

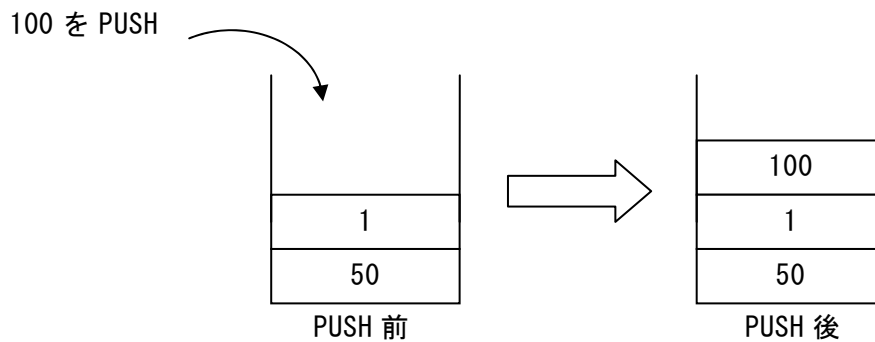


図 1 スタックにデータを PUSH する

POP はスタックからデータを取り出す命令で、図 2 のように、スタックの最上部にあるデータを取り出す。

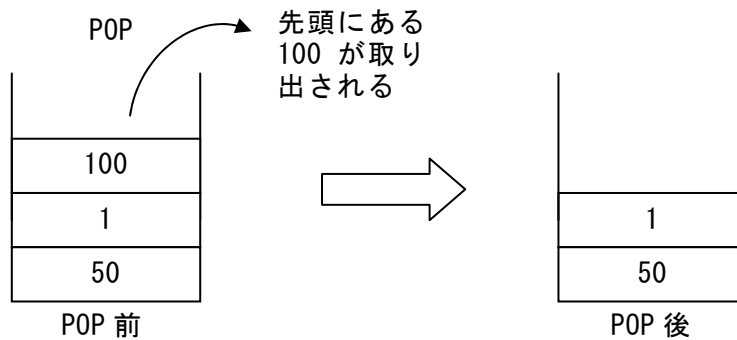


図 2 スタックからデータを POP する

なお、PUSH および POP の書式は、次のようにする。

- PUSH

書 式 : PUSH(データ)

機 能 : データをスタックに格納する。

使用例 : PUSH(100)

- POP

書 式 : POP()

機 能 : スタックから取り出したデータを返す。

使用例 : X = POP()

<設問 1 > 次の①～⑤の命令を連続で行った結果、スタックに残っている値を解答群から選べ。なお、スタックは空の状態から始めるものとする。また、X と Y はスタック操作に使用する変数である。

- ① PUSH(1)
- ② PUSH(2)
- ③ X = POP()
- ④ PUSH(X)
- ⑤ Y = POP()

(1) の解答群

- ア. 1
- イ. 1 と 2
- ウ. 2
- エ. 空

<設問 2 > スタックの利用方法として最も適切なものを解答群から選べ。

(2) の解答群

- ア. ハッシュ値を計算する
- イ. 線形リストの先頭要素のポインタを管理する
- ウ. 窓口業務の待ち行列から順番にデータを取り出す
- エ. 呼び出した副プログラムが終了した時に戻る場所を格納する

<設問 3 > 次の逆ポーランド記法に関する記述中の に入れるべき適切な字句を解答群から選べ。

一般的に使われる式は中置記法と呼ばれ、"A+B"のように記述する。これに対し、逆ポーランド記法（後置記法）は、演算子を被演算子の後に記述するものである。中置記法で"A+B"という式を逆ポーランド記法で表すと"AB+"となる。

また、逆ポーランド記法を用いれば、中置記法で演算の優先順位を変更する場合に用いる括弧が不要になる。例えば、中置記法で"(A-B)÷(C+D)"を逆ポーランド記法で記述すると"AB-CD+÷"となる。

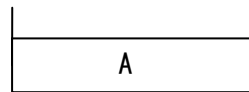
逆ポーランド記法で記述された式は、スタックを用いて次のように計算できる。

- ① 式から1つの要素を取り出す。
- ② 取り出した要素が数値ならば、スタックに追加する。また、取り出した要素が演算子ならば、スタックから値を2つ取り出し、最初に取り出した値を演算数、2つ目に取り出した値を被演算数として演算を行い、結果をスタックに格納する。

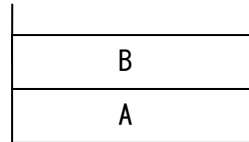
最後にスタックに残った値が計算結果となる。

(例) "AB-CD+÷"の場合

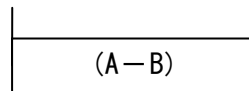
① 1文字目の"A"をスタックへ格納



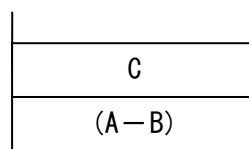
② 2文字目の"B"をスタックへ格納



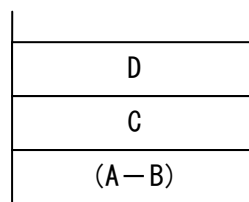
③ 3文字目は演算子の"-"なのでスタックから2つの値を取り出して引き算を行った結果をスタックへ格納



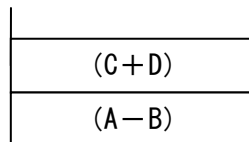
④ 4文字目の"C"をスタックへ格納



⑤ 5文字目の"D"をスタックへ格納



⑥ 7文字目は演算子の"+"なのでスタックから2つの値を取り出して足し算を行った結果をスタックへ格納



⑦ 8文字目は演算子の"÷"なのでスタックから2つの値を取り出して割り算を行った結果をスタックへ格納

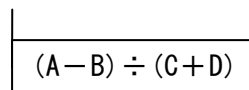


図3 逆ポーランド記法の式を計算する

ここで、逆ポーランド記法による式が"ABCD+-÷"であり、A=20、B=10、C=2、D=3である場合、最後にスタックに残る値（計算結果）は

(3)

 になる。

(3) の解答群

ア. 4

イ. 5

ウ. 7

エ. 10

<設問 4> 次の逆ポーランド記法の計算手順に関する記述中の に入れるべき適切な字句を解答群から選べ。

スタックを操作する PUSH および POP 命令で逆ポーランド記法による式 "ABC-÷" を計算する場合、次のような順番で実行する。なお、X と Y はスタック操作に使用する変数である。

- ① PUSH(A)
- ② PUSH(B)
- ③ PUSH(C)
- ④ X = POP()
- ⑤ Y = POP()
- ⑥
- ⑦ X = POP()
- ⑧ Y = POP()
- ⑨

(4) , (5) の解答群

- | | |
|--------------|--------------|
| ア. PUSH(X) | イ. PUSH(X-Y) |
| ウ. PUSH(X÷Y) | エ. PUSH(Y) |
| オ. PUSH(Y-X) | カ. PUSH(Y÷X) |

問題2 次のファイル集計の説明を読み、各設問に答えよ。

[ファイル集計の説明]

Jファミリーレストランでの月ごとの時間区分売上一覧表を表示するため、ファイルを集計する処理である。

時間区分売上一覧表は、月ごとに時間区分（1：5:00～10:59，2：11:00～16:59，3：17:00～24:00）別に売上金額の多いメニュー順に出力する。

時間区分 売上一覧表								
9月								
5:00～10:59			11:00～16:59			17:00～24:00		
順位	メニュー名	金額	順位	メニュー名	金額	順位	メニュー名	金額
1位	鮭定食	361,200	1位	日替わり定食	1,185,750	1位	ステーキセット	578,600
2位	納豆定食	345,800	2位	焼き肉定食	1,123,500	2位	おライス	567,200
3位	タコサトセット	346,200	3位	焼き魚定食	1,118,300	3位	カルパナラ	479,000
⋮			⋮			⋮		
⋮			⋮			⋮		
⋮			⋮			⋮		
合計		3,159,600	合計		6,855,100	合計		6,523,800
						月合計		16,538,500

図1 時間区分 売上一覧表

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	日付			時間区分	商品1		⋮	商品n	
	年	月	日		メニューコード	個数		メニューコード	個数

メニューファイル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。

メニューコード	メニュー名	単価

売上データ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。

日付			時間区分	メニューコード	メニュー名	金額
年	月	日				
U_NEN	U_TUKI	U_HI	U_JIKAN	U_MENU	U_MENUMEI	U_KINGAKU

集計データ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	時間区分	メニューコード	メニュー名	メニュー計

時間区分売上一覧表を表示するための流れは次のようになる。

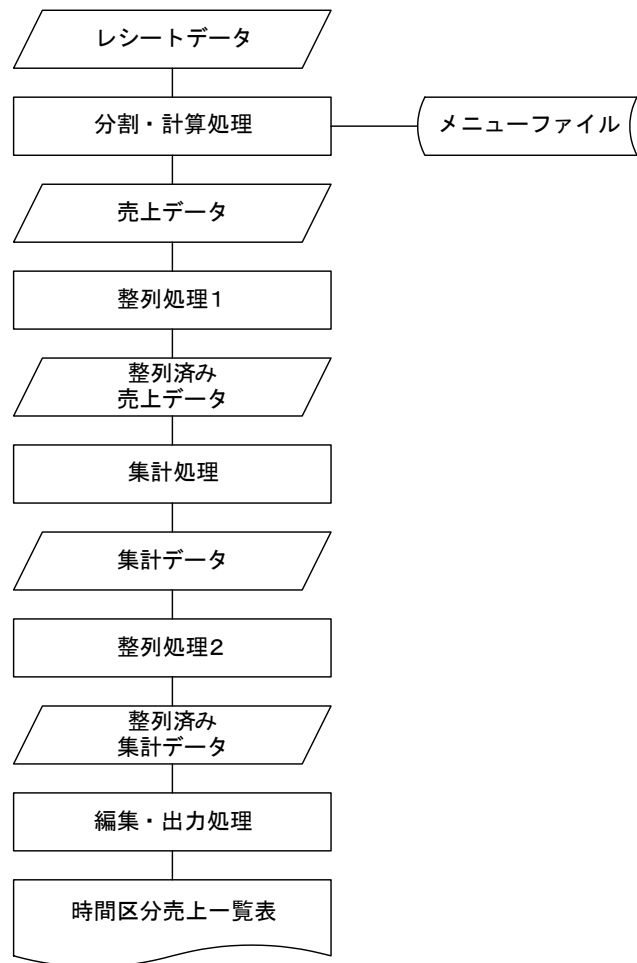


図2 時間区分売上一覧表を表示する流れ

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	レシートデータをメニューごとのレコードに分割するとともに、メニューファイルよりメニュー名・単価を読み込み、売上金額を求め、売上データを作成する。
整列処理1	売上データを並べ換えて、整列済み売上データを作成する。売上データと整列済み売上データのレコード形式は同じである。
集計処理	月ごとの時間区分別にメニューごとの売上金額を集計し、集計データを作成する。
整列処理2	集計データを並べ換えて、整列済み集計データを作成する。集計データと整列済み集計データのレコード形式は同じである。
編集・ 出力処理	整列済み集計データを、時間区分ごとの売上一覧表の形式に編成し出力する。

<設問 1 > 次の集計処理の流れ图中的 [] に入れるべき適切な字句を解答群から選べ。

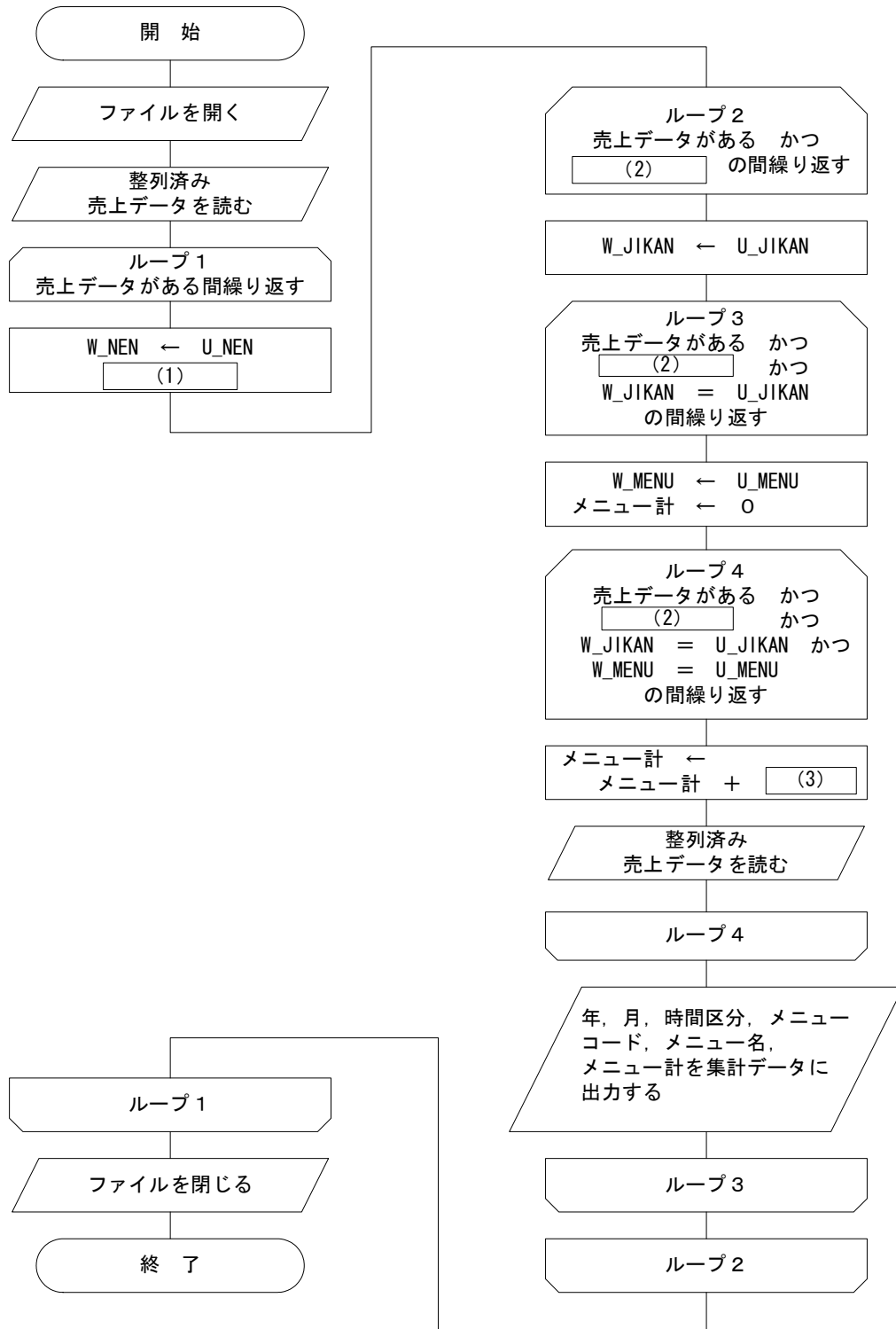


図 3 集計処理の流れ図

問題3 次のスポーツクラブに関する記述を読み、各設問に答えよ。

ある会員制スポーツクラブでは、利用者の利用形態に合わせて様々な会員種別を用意しており、利用者の希望を聞いて最適な会員種別を提案している。

なお、営業時間は月曜日から土曜日は10:00～22:00、日曜日と祝日は10:00～19:00である。

表1 会員種別

会員種別	利用可能時間帯	月会員料
正会員	すべての営業日・時間帯で利用可能。	10,000円
デイ会員	平日(月曜日～金曜日)の昼(10:00～18:00)の時間帯だけ利用可能。ただし祝日は不可。	8,000円
ホリデイ会員	土曜日、日曜日、祝日のすべての営業時間で利用可能。	7,000円
ナイト会員	平日(月曜日～金曜日)の夜(18:00～22:00)の時間帯だけ利用可能。ただし祝日は不可。	6,000円
月4会員	月に4回まで、すべての営業日・時間帯で利用可能。	6,000円

利用者は1度だけ無料体験を利用できるが、その後はいずれかの会員にならないと利用できない。また会員の利用可能時間帯以外での利用は、1回につき2,000円が別途必要になる。

利用者の利用形態に最適な会員種別を選べるように決定表を作成した。

なお、複数の選択肢が考えられる場合は利用者の利益になるような設定を勧める。例えば、平日の昼と夜に共に複数回利用する場合は、デイ会員またはナイト会員にして別途費用を徴収するよりも正会員を勧めるようにする。

平日と休日、および昼と夜は、両方ともN(N0)となる場合は全く利用しないことなので、少なくとも一方はY(YES)になる。また表中の記号“-”はYでもNでも処理に関係しないことを表している。

<設問 1 > 次の決定表中の に入れるべき適切な字句を解答群から選べ。

表 2 決定表

月 5 回以上利用	(1)	Y	Y	(2)	(3)	N
平日に利用		Y	Y			—
休日に利用		N	N			—
昼に利用		Y	Y			—
夜に利用		Y	N			—
正会員になる	X	(4)	(5)			
デイ会員になる						
ホリデイ会員になる					X	
ナイト会員になる				X		
月 4 会員になる						X

(1) ~ (3) の解答群

ア.	<input type="text" value="N"/> <input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="N"/> <input type="text" value="Y"/>	イ.	<input type="text" value="N"/> <input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="N"/>	ウ.	<input type="text" value="N"/> <input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="Y"/>	エ.	<input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="N"/>
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

オ.	<input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="N"/> <input type="text" value="—"/> <input type="text" value="—"/>	カ.	<input type="text" value="Y"/> <input type="text" value="N"/> <input type="text" value="Y"/> <input type="text" value="—"/> <input type="text" value="—"/>	キ.	<input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="N"/> <input type="text" value="N"/> <input type="text" value="Y"/>	ク.	<input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="Y"/> <input type="text" value="Y"/>
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(4) , (5) の解答群

ア.	<input type="text" value="X"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	イ.	<input type="text"/> <input type="text" value="X"/> <input type="text"/> <input type="text"/> <input type="text"/>	ウ.	<input type="text"/> <input type="text"/> <input type="text" value="X"/> <input type="text"/> <input type="text"/>	エ.	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text" value="X"/> <input type="text"/>	オ.	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text" value="X"/>
----	--------------------------------------------------------------------------------------------------------------------------------	----	--------------------------------------------------------------------------------------------------------------------------------	----	--------------------------------------------------------------------------------------------------------------------------------	----	--------------------------------------------------------------------------------------------------------------------------------	----	--------------------------------------------------------------------------------------------------------------------------------

<設問 2 > 次の流れ図の説明を読み、に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

決定表をもとに流れ図を作成した。利用者の利用形態は次の変数に 1 または 0 で入力される。それ以外の値が入力されることはない。

KAI … 1 は月 5 回以上利用する。0 は月 4 回以下の利用となる。

HEI … 1 は平日に利用する。0 は平日に利用しない。

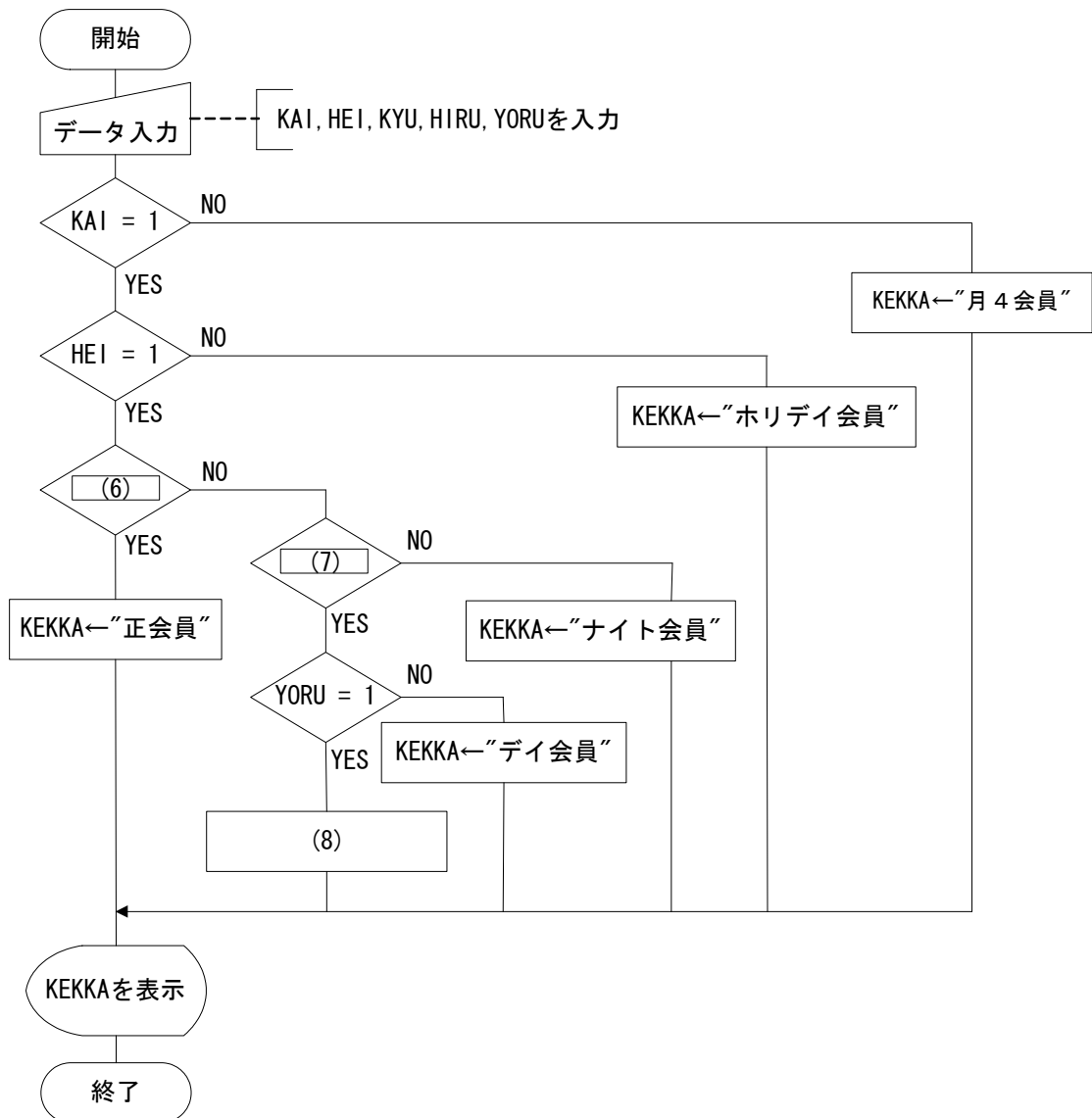
KYU … 1 は休日に利用する。0 は休日に利用しない。

HIRU … 1 は昼に利用する。0 は昼に利用しない。

YORU … 1 は夜に利用する。0 は夜に利用しない。

判定結果は、変数 KEKKA に文字データとして会員種別を求める。

[流れ図]



(6) , (7) の解答群

ア. KAI = 0

イ. KYU = 0

ウ. KYU = 1

エ. HEI = 0

オ. HIRU = 0

カ. HIRU = 1

(8) の解答群

ア. KEKKA ← “月4会員”

イ. KEKKA ← “正会員”

ウ. KEKKA ← “デイ会員”

エ. KEKKA ← “ナイト会員”

オ. KEKKA ← “ホリデイ会員”

問題4 次のプログラムの説明を読み、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。

[プログラムの説明]

商品マスタファイルの内容を変更ファイルの内容で更新し、新商品マスタファイルを作成する処理である。更新の種別は、次の3種類である。

種別 (shubetu) : 1 追加処理

新商品マスタファイルに変更ファイルのレコードを出力する。もし、商品マスタファイルに同じ商品コードのレコードが存在する場合は、エラー表示をする。

種別 (shubetu) : 2 更新処理

変更ファイルの商品コードと同じ商品コードを持つ商品マスタファイルのレコードに対して、変更レコードで置き換えたレコードを出力する。もし、商品マスタファイルに同じ商品コードが存在しない場合は、エラー表示をする。

種別 (shubetu) : 3 削除処理

変更ファイルの商品コードと同じ商品コードを持つ商品マスタファイルのレコードを新商品マスタファイルに出力しない。もし、商品マスタファイルと同じ商品コードが存在しない場合は、エラー表示をする。

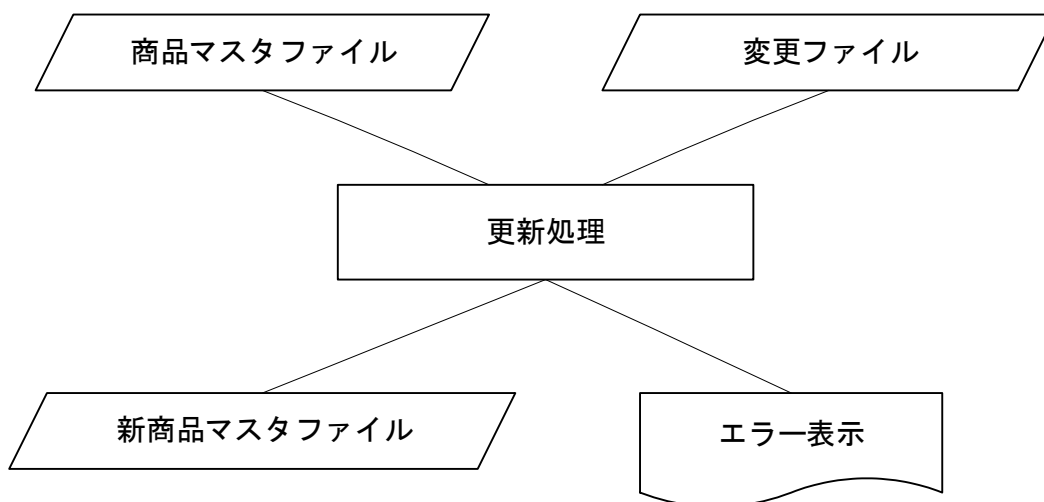


図1 商品マスタファイルを更新する処理

各ファイルは順編成ファイルであり、形式は次のようになっている。

商品マスタファイル (M_file)

商品コード	商品名	単価
-------	-----	----

変更ファイル (H_file)

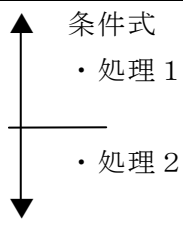
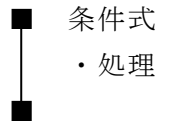
商品コード	商品名	単価	種別
-------	-----	----	----

新商品マスタファイル (NM_file)

商品コード	商品名	単価
-------	-----	----

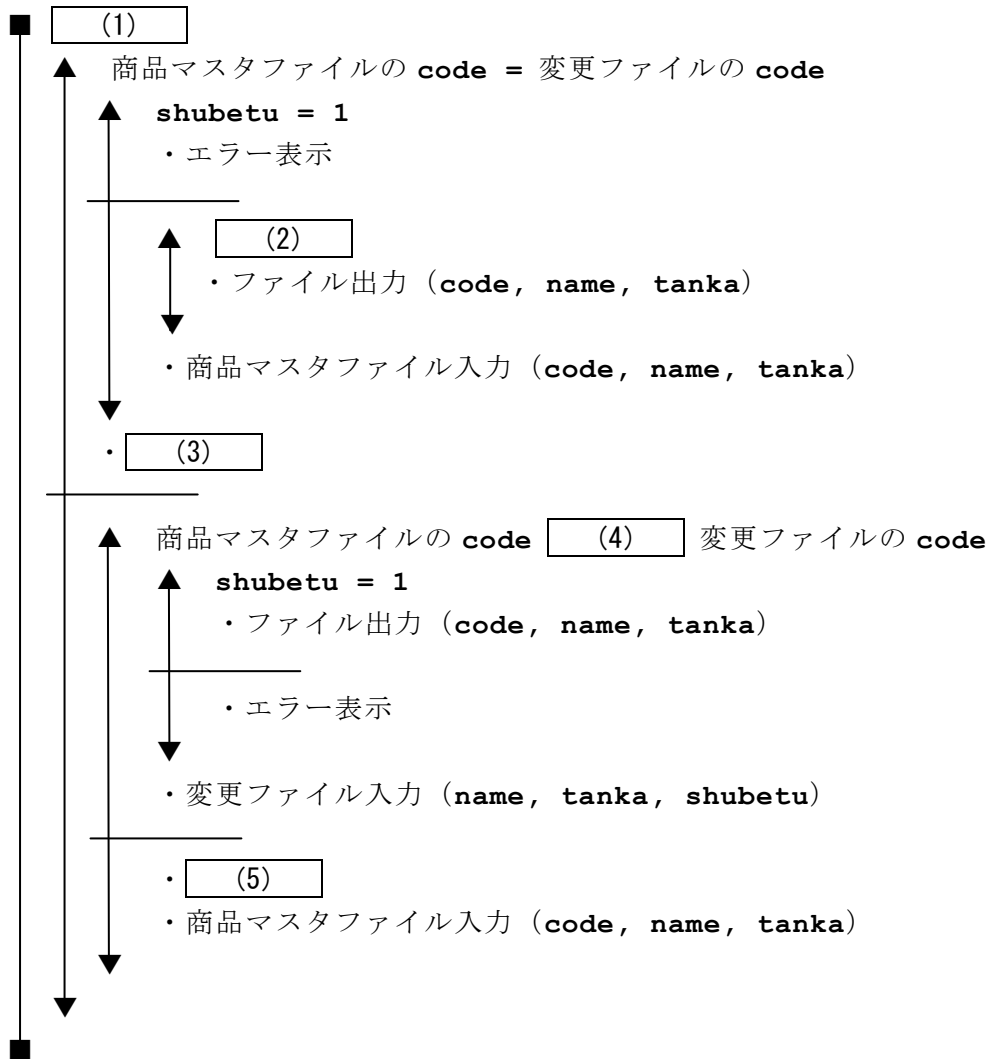
商品マスタファイル、変更ファイルとも商品コードの昇順に整列されており、各ファイルに同じ商品コードを持つレコードは存在しない。なお、商品マスタファイルと変更ファイルの最後には商品コードが‘9999’のレコードがある。これはファイルの最後を表し、新商品マスタファイルには出力しない。

[疑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
	選択処理を示す。 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。
	前判定繰り返し処理を示す。 条件式が真の間、処理を実行する。

[プログラム]

- プログラム名：商品マスタの更新
- 整数型：**code, tanka, shubetu**
- 文字型配列：**name**
- 手続き：商品マスタファイル入力 (**code, name, tanka**)
- 手続き：変更ファイル入力 (**code, name, tanka, shubetu**)
- 手続き：ファイル出力 (**code, name, tanka**)
 - ・商品マスタファイル入力 (**code, name, tanka**)
 - ・変更ファイル入力 (**code, name, tanka, shubetu**)



(1) の解答群

- ア. 商品マスタファイルの `code = 9999` かつ 変更ファイルの `code = 9999`
- イ. 商品マスタファイルの `code = 9999` または 変更ファイルの `code = 9999`
- ウ. 商品マスタファイルの `code ≠ 9999` または 変更ファイルの `code ≠ 9999`

(2) の解答群

- ア. `shubetu = 1` イ. `shubetu = 2` ウ. `shubetu = 3`

(3) , (5) の解答群

- ア. エラー表示
- イ. 商品マスタファイル入力 (`code, name, tanka`)
- ウ. ファイル出力 (`code, name, tanka`)
- エ. 変更ファイル入力 (`code, name, tanka, shubetu`)

(4) の解答群

- ア. `>` イ. `=` ウ. `<`

< 選 択 問 題 >

選択問題は問題から1つ選択し解答せよ。

選択した問題は必ず、解答用紙「選択欄」にマークすること。

※選択欄にマークがなく、解答のみの場合は採点を行いません。

各構成は以下のとおり。

選択問題

- | | |
|------------|---------------|
| ・ C言語の問題 | 18 ページ～21 ページ |
| ・ 表計算の問題 | 22 ページ～26 ページ |
| ・ アセンブラの問題 | 28 ページ～33 ページ |

選択問題 C言語の問題

次のC言語プログラムの説明を読み、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。

[プログラムの説明]

テキストファイルに含まれる単語と、その出現数を出力するプログラムcountWordである。

- ・テキストファイルはShift-JISで保存されており、含まれる文字はすべて半角文字の英数字と記号で構成されている。
- ・単語の区切りとして使用する文字は、図の13種類とする。

,	.	()	!	?	<	>	/	"	空白	改行	タブ
---	---	---	---	---	---	---	---	---	---	----	----	----

図 単語の区切り文字

- ・テキストファイルに含まれる単語は、最大でも1000種類とする。

[関数の説明]

countWord関数

引 数：ファイルポインタ。

機 能：引数のファイルポインタで示されるテキストファイルを入力し、単語ごとの出現数を集計して出力する。

戻り値：なし。

addWord関数

引 数：文字配列のポインタ。

機 能：引数で受け取ったポインタが示す文字列が、新しく出現した単語であれば構造体配列に格納し、そうでなければ出現数に1を加える。

戻り値：なし。

findWord関数

引 数：文字配列のポインタ。

機 能：引数で受け取ったポインタが示す文字列が、構造体配列に存在するか調べる。

戻り値：構造体配列に存在すれば要素位置、存在しなければ-1。

isDelSym関数

引数：文字。

機能：引数で受け取った文字が、単語の区切り文字か判定する。

戻り値：区切り文字の場合は1，そうでない場合は0。

[プログラム]

```

#define SYM_N      13          /* 単語区切り文字の数 */
#define SIZE      256        /* 単語の大きさ */
#define MX_WORDS  1000       /* 単語の最大数 */
#define NOTFOUND (-1)

typedef struct {
    char  word[SIZE];
    int   cnt;
} WORD_REC;

WORD_REC tango[MX_WORDS]; /* 出現する単語を格納する構造体配列 */
int word_max;             /* 構造体配列の添え字 */
void countWord(FILE *fp) {
    char  buff[SIZE], temp[SIZE];
    int   i, j;
    word_max = 0;
    while(fgets(buff, SIZE, fp) != NULL) {
        i = 0;
        while( i < strlen(buff) ) {
            (1);
            /* 単語をtempへ取り出す */
            while( isDelSym(buff[i]) == 0 ) {
                temp[j] = buff[i];
                i++;
                j++;
            }
            (2);
            if (j > 0) addWord(temp);
            (3);
        }
    }
    /* 結果の出力 */
    for(i = 0; i < word_max; i++) {
        printf(" %d\t%s\n", tango[i].cnt, tango[i].word);
    }
}

```

```

void addWord(char *buff) {
    int    k;
    k = findWord(buff);
    if (k == NOTFOUND) {
        /* 新しい単語であれば構造体配列へ追加 */
        strcpy(tango[word_max].word, buff);
        tango[word_max].cnt = 1;
        (4);
    } else {
        (5);
    }
}

int findWord(char *buff) {
    int    i, ret_value;
    ret_value = NOTFOUND;
    i = 0;
    while(i < word_max && ret_value < 0) {
        if (strcmp(buff, tango[i].word) == 0) {
            (6);
        } else {
            i++;
        }
    }
    return ret_value;
}

int isDelSym(char ch) {
    static char sym[] = { ',', '.', '(', ')', '!', '?', '<', '>',
                        '/', '"', ' ', '\n', '\t' };

    int i, ret_value;
    (7);
    i = 0;
    while(i < SYM_N && ret_value == 0) {
        if (sym[i] == ch) {
            ret_value = 1;
        } else {
            i++;
        }
    }
    return ret_value;
}

```

(1) , (3) の解答群

ア. `i++`

ウ. `i = word_max`

オ. `j = 0`

イ. `i = 0`

エ. `j++`

カ. `j = word_max`

(2) の解答群

ア. `temp[i] = '¥0'`

ウ. `temp[j] = '¥0'`

イ. `temp[i+1] = '¥0'`

エ. `temp[j+1] = '¥0'`

(4) , (5) の解答群

ア. `tango[k].cnt++`

ウ. `tango[word_max].cnt++`

オ. `word_max++`

イ. `tango[k].cnt = 0`

エ. `tango[word_max].cnt = 0`

カ. `word_max = 0`

(6) , (7) の解答群

ア. `ret_value = 0`

ウ. `ret_value = i - 1`

オ. `ret_value++`

イ. `ret_value = 1`

エ. `ret_value = i`

カ. `ret_value = NOTFOUND`

次の表計算ソフトの記述を読み、記述中の に入れるべき適切な式を解答群から選べ。

この問題で使用する表計算ソフトの仕様は下記のとおりである。

IF 関数

条件が真の時に「真の場合」、偽の時に「偽の場合」の計算結果や値を返す。

IF 関数の中に IF 関数を入れることができる。

書式：IF (条件式, 真の場合, 偽の場合)

LEFT 関数

文字列の左端から文字数で指定した位置までの文字列を返す。

書式：LEFT (文字列, 文字数)

RIGHT 関数

文字列の右端から文字数で指定した位置までの文字列を返す。

書式：RIGHT (文字列, 文字数)

MAX 関数

範囲の中で一番大きい値を返す。

書式：MAX (範囲)

MIN 関数

範囲の中で一番小さい値を返す。

書式：MIN (範囲)

RANK 関数

範囲内の数値を並べたときに何番目になるか (順位) を返す。順序は、降順の場合は 0, 昇順の場合は 1 を設定する。なお、範囲内の検査値に同じものがあれば同じ順位を返し、以降の順位に欠番が生じる。

書式：RANK (検査値, 範囲, 順序)

SUM 関数

指定した範囲の合計値を求める。

書式：SUM (開始セル番地 : 終了セル番地)

VLOOKUP 関数

検索範囲から、検索値を探し、位置で指定したセルの値を返す。位置は 1 から始まる相対的な値であり、検索範囲中に見つけた行の中で、左から何番目の列かを示す。検索方法は 0 または 1 を指定し、0 の場合は完全に一致する値を、1 の場合は検索値以下の最大値を探す。

書式：VLOOKUP (検索値, 検索範囲, 位置, 検索方法)

式

=に続けて計算式や関数などを入力する。

&

文字列の結合をする演算子。

セル番地の参照

セル番地に\$を付けることで、絶対番地（絶対参照）を表す。

他のワークシートの参照

「ワークシート名!セル番地」とすることで他のワークシートのセルを参照することができる。また、ワークシート名を"sheet1:sheet5"のように記述すると、ワークシート範囲を指定することができる。

J町内会では、町内会イベントの目玉としてダンスコンテストを開催した。

参加資格は、2人以上であること以外は自由である。なお、加点のため次のようなカテゴリを設けるが、順位付けは全体で行う。

表 参加カテゴリ

カテゴリ	条件
ジュニア	小学生以下のみのチーム
ファミリー	親子での参加であり、子供は小学生以下であるチーム
一般	上記2つ以外のチーム

- 評価は 5 人の審査員が行う。各審査員は、技術力、表現力、一体感の 3 要素をそれぞれ 10 点満点で評価する。3 要素を合計して審査員の点数とする。5 人の審査員の中で、一番高い点数と一番低い点数を除いた 3 人の審査員の点数を合わせたものが、チームの得点となる。
- ジュニア部門には 10 点、ファミリー部門には 5 点の加点がある。
- チームの人数が 10 人以上の場合は大編成とし、10 人未満は小編成とした。大編成の場合は 5 点の加点がある。
- チームの順位は(a)の得点と(b)、(c)の加点を合わせた合計得点の高い順に決まる。

- (e) 順位の1位が優勝，2位が準優勝となり，3位以下で合計得点が80点以上であれば敢闘賞が与えられる。
- (f) 1位のチームが複数あった場合は，それらのチーム全て優勝とし，準優勝は与えない。
- (g) 2位のチームが複数あった場合は，それらのチーム全て準優勝とする。

町内会の役員であるAさんは，表計算ソフトを用いてダンスコンテストの集計を行うことにした。

なお，このコンテストには10組のチームが参加を申し込んだ。

Aさんは，コンテストに申し込んだチームとその区分を入力したワークシートを作成し，“エントリーシート”と名前を付けた（図1）。C列の区分コードは，カテゴリの区分と小編成か大編成の区分を記号化して合わせたものである。1文字目がカテゴリの区分で，2文字目が編成の区分になっている。

	A	B	C	D	E	F	G
1	エントリーNo	チーム名	区分コード		区分	カテゴリ名	加点
2	1	ファンタジスタ	JS		J	ジュニア	10
3	2	Kファミリー	FS		F	ファミリー	5
4	3	定丸舞踏団	GS		G	一般	0
5	4	ASB35	GL				
6	5	RBスクール	GS				
7	6	Teamモモ	FS		区分	編成名	加点
8	7	PAC隊	JL		S	的小编成	0
9	8	Pickey	FS		L	大編成	5
10	9	AXILE	GL				
11	10	チームDSK	GS				

図1 エントリーシート

また，審査員が評価を入力するためのワークシートを作成した（図2）。A列の“エントリーNo”とB列の“チーム名”は，図1のエントリーシートから複写し，F列の点数は，セルF2に式“=SUM(C2:E2)”を入力してセルF3～F11に複写した。

5人の審査員がいるので，このワークシートを4つ複写し，それぞれ“採点シート1”，“採点シート2”，…，“採点シート5”という名前を付けて連続して配置した。

	A	B	C	D	E	F
1	エントリーNo	チーム名	技術力	表現力	一体感	点数
2	1	ファンタジスタ				0
3	2	Kファミリー				0
4	3	定丸舞踏団				0
5	4	ASB35				0
6	5	RBスクール				0
7	6	Teamモモ				0
8	7	PAC隊				0
9	8	Pickey				0
10	9	AXILE				0
11	10	チームDSK				0

図2 採点シート

Aさんは、これらの採点シートを集計するための集計シートを作成した（図3）。

	A	B	C	D	E	F	G	H
1	エントリーNo	チーム名	区分コード	区分名	加点	合計得点	順位	表彰
2	1	ファンタジスタ	JS	ジュニア小編成	10	73	9	
3	2	Kファミリー	FS	ファミリー小編成	5	76	5	
4	3	定丸舞踏団	GS	一般小編成	0	83	2	準優勝
5	4	ASB35	GL	一般大編成	5	76	5	
6	5	RBスクール	GS	一般小編成	0	88	1	優勝
7	6	Teamモモ	FS	ファミリー小編成	5	76	5	
8	7	PAC隊	JL	ジュニア大編成	15	81	3	敢闘賞
9	8	Pickey	FS	ファミリー小編成	5	77	4	
10	9	AXILE	GL	一般大編成	5	74	8	
11	10	チームDSK	GS	一般小編成	0	69	10	

図3 集計シート

A列からC列は、エントリーシートからコピーした。

D列は、C列の区分コードからカテゴリ名と編成名を合わせたものを区分名として表示するため、セルD2に次の式を入力し、セルD3～D11まで複写した。

$$= \text{ (1) } \& \text{ (2) }$$

E列は、カテゴリと編成により加点される値を表示するため、セルE2に次の式を入力し、セルE3～E11まで複写した。

$$= \text{ (3) } + \text{ (4) }$$

(1) , (3) の解答群

- ア. VLOOKUP (LEFT (C2, 1), エントリーシート!E2:G4, 2, 0)
- イ. VLOOKUP (LEFT (C2, 1), エントリーシート!E2:G4, 3, 0)
- ウ. VLOOKUP (LEFT (C2, 1), エントリーシート!E\$2:G\$4, 2, 0)
- エ. VLOOKUP (LEFT (C2, 1), エントリーシート!E\$2:G\$4, 3, 0)
- オ. VLOOKUP (LEFT (C\$2, 1), エントリーシート!E\$2:G\$4, 2, 0)
- カ. VLOOKUP (LEFT (C\$2, 1), エントリーシート!E\$2:G\$4, 3, 0)

(2) , (4) の解答群

- ア. VLOOKUP (RIGHT (C2, 1), エントリーシート!E8:G9, 2, 0)
- イ. VLOOKUP (RIGHT (C2, 1), エントリーシート!E8:G9, 3, 0)
- ウ. VLOOKUP (RIGHT (C2, 1), エントリーシート!E\$8:G\$9, 2, 0)
- エ. VLOOKUP (RIGHT (C2, 1), エントリーシート!E\$8:G\$9, 3, 0)
- オ. VLOOKUP (RIGHT (C\$2, 1), エントリーシート!E\$8:G\$9, 2, 0)
- カ. VLOOKUP (RIGHT (C\$2, 1), エントリーシート!E\$8:G\$9, 3, 0)

F 列の合計得点は，“採点シート 1”から“採点シート 5”の F 列に集計された点数から一番高い得点と一番低い得点を除いた合計値に E 列（加点）を加えて求める。セル F2 に次の式を入力し，セル F3～F11 に複写した。

$$= \text{SUM}(\text{採点シート 1}:\text{採点シート 5}!F2) - (\boxed{(5)}) + \boxed{(6)}$$

(5) , (6) の解答群

- ア. E2
- イ. MAX(採点シート 1:採点シート 5!F2)
- ウ. MAX(採点シート 1:採点シート 5!F2) + MIN(採点シート 1:採点シート 5!F2)
- エ. MAX(採点シート 1:採点シート 5!F2) - MIN(採点シート 1:採点シート 5!F2)
- オ. MIN(採点シート 1:採点シート 5!F2)
- カ. SUM(E\$2:E2)

G 列の順位は，合計得点の高いチームが 1 位になるように，次の式をセル G2 に入力し，セル G3～G11 に複写した。

$$= \boxed{(7)}$$

(7) の解答群

- ア. RANK(F2, F\$2:F\$11, 0)
- イ. RANK(F2, F\$2:F\$11, 1)
- ウ. RANK(F\$2, F2:F11, 0)
- エ. RANK(F\$2, F2:F11, 1)

H 列の表彰は，順位の結果と合計得点の結果により，優勝，準優勝，敢闘賞と表示するため，次の式をセル H2 に入力し，セル H3～H11 に複写した。

$$= \boxed{(8)}$$

(8) の解答群

- ア. IF(G2=1, "優勝", IF(G2=2, "準優勝", IF(F2>=80, "敢闘賞", "")))
- イ. IF(G2<3, IF(G2=1, "優勝", "準優勝"), IF(G2>=80, "敢闘賞", ""))
- ウ. IF(G2>=3, IF(G2=1, "優勝", "準優勝"), IF(F2>=80, "敢闘賞", ""))
- エ. IF(G2>=80, "敢闘賞", IF(G2=1, "優勝", IF(G2=2, "準優勝", "")))

問題を読みやすくするために、
このページは空白にしてあります。

選択問題 アセンブラの問題

次のアセンブラ言語CASL II プログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

IN 命令で 10 進数形式の文字列を DEC 番地以降に入力して 2 進数に変換し、BIN 番地に求める副プログラムである。

IN 命令で入力されるデータはすべて文字データのため、10 進数 1 桁の値 0~9 も文字コードの '0' ~ '9' で入力される。なお、入力した文字が '0' ~ '9' 以外は 'NOT NUMERIC' というエラーメッセージを表示する。

10 進数、文字コードおよび文字コードの 16 進表示の関係を表 1 に示す。

表 1 10 進数と文字コードの対応表

10 進数	文字コード	16 進表示
0	'0'	#0030
1	'1'	#0031
2	'2'	#0032
3	'3'	#0033
4	'4'	#0034
5	'5'	#0035
6	'6'	#0036
7	'7'	#0037
8	'8'	#0038
9	'9'	#0039

また、プログラムの実行例を図 1 に示す。

DEC 番地+0	0000 0000 0011 0010	'2'	BIN 番地	0000 0001 0000 0000
+1	0000 0000 0011 0101	'5'		
+2	0000 0000 0011 0110	'6'		
	:			

図 1 入力データ '256' のプログラムの実行例

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	DBCHN	START		
110		RPUSH		
120		LD	GR0,=0	
130		LD	GR1,=0	
140		IN	DEC,DLEN	
150	LOOP1	ST	GR0,BIN	
160		LD	GR2,DEC,GR1	
170		CPA	GR2,='0'	
180			(1)	
190		CPA	GR2,='9'	
200			(2)	
210		AND	GR2,#000F	;文字を数値に変換
220		SLA	GR0,2	
230		ADDA	GR0,BIN	
240			(3)	
250		ADDA	GR0,GR2	
260		ADDA	GR1,=1	
270		CPA	GR1,DLEN	
280		JMI	LOOP1	
290			(4)	
300		JUMP	OWARI	
310	ERR	ST	GR2,EMSG	
320		OUT	EMSG,ELEN	
330	OWARI	RPOP		
340		RET		
350	BIN	DS	1	
360	EMSG	DS	1	
370		DC	':NOT NUMERIC'	
380	ELEN	DC	13	
390	DLEN	DS	1	
400	DEC	DS	256	
410		END		

<設問 1 > 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。

(1) , (2) の解答群

ア. **JMI ERR**

ウ. **JNZ ERR**

オ. **JPL ERR**

キ. **JZE ERR**

イ. **JMI OWARI**

エ. **JNZ OWARI**

カ. **JPL OWARI**

ク. **JZE OWARI**

(3) の解答群

ア. **ADDA GR0,BIN**

ウ. **SLA GR0,1**

イ. **ADDA GR2,BIN**

エ. **SLA GR0,2**

(4) の解答群

ア. **LD GR0,GR2**

ウ. **ST GR0,BIN**

イ. **LD GR2,GR0**

エ. **ST GR2,BIN**

<設問 2 > 行番号 210 と置き換えることのできる命令を解答群から選べ。

(5) の解答群

ア. **SLA GR0,12**

ウ. **SUBA GR0,=30**

イ. **SRA GR0,4**

エ. **SUBA GR0,='0'**

<設問 3> 次のプログラムの説明を読み、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。

[プログラムの説明]

本問のプログラムを応用して、16進数形式の文字列を HEX 番地以降に入力して 2 進数に変換し、BIN 番地に求めるプログラムを作成した。

入力されるデータは 16 進数 4 桁で、4 桁以外が入力されたとき、または、'0' ~ '9' 及び 'A' ~ 'F' 以外の文字が入力されたときはエラー表示する。

16 進数の A~F、文字コードおよび文字コードの 16 進表示の関係を表 2 に示す。

表 2 16 進数と文字コードの対応表

16 進数	文字コード	16 進表示
A	'A'	#0041
B	'B'	#0042
C	'C'	#0043
D	'D'	#0044
E	'E'	#0045
F	'F'	#0046

また、プログラムの実行例を図 2 に示す。

HEX 番地+0	<input type="text" value="0000 0000 0011 0010"/>	'2'	BIN 番地	<input type="text" value="0010 1010 0011 1011"/>
+1	<input type="text" value="0000 0000 0100 0001"/>	'A'		
+2	<input type="text" value="0000 0000 0011 0011"/>	'3'		
+3	<input type="text" value="0000 0000 0100 0010"/>	'B'		
	:			

図 2 入力データ '2A3B' のプログラムの実行例

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	HBCHN	START		
110		RPUSH		
120		LD	GR0,=0	
130		LD	GR1,=0	
140		LD	GR3,=4	
150		IN	HEX,HLEN	
160		CPA	GR3,HLEN	
170		JNZ	ERR2	
180	LOOP1	LD	GR2,HEX,GR1	
190		CPA	GR2,'0'	
200			(1)	
210		CPA	GR2,'9'	
220			(6)	
230		AND	GR2,=#000F	;文字を数値に変換
240		JUMP	OK	
250	NEXT	CPA	GR2,'A'	
260			(1)	
270		CPA	GR2,'F'	
280			(2)	
290		SUBA	GR2,=55	; 'A' ~ 'F' を 10 ~ 15 に変換
300	OK		(7)	
310		ADDL	GR0,GR2	
320		ADDA	GR1,=1	
330		CPA	GR1,=4	
340		JMI	LOOP1	
350			(4)	
360		JUMP	OWARI	
370	ERR	ST	GR2,EMSG	
380		OUT	EMSG,ELEN	
390		JUMP	OWARI	
400	ERR2	OUT	EMSG2,ELEN2	
410	OWARI	RPOP		
420		RET		
430	BIN	DS	1	
440	EMSG	DS	1	
450		DC	':NOT NUMERIC'	
460	ELEN	DC	13	

行番号	ラベル	命令	オペランド	コメント
470	EMSG2	DC	'KETA ERROR'	
480	ELEN2	DC	10	
490	DLEN	DS	1	
500	DEC	DS	256	
510		END		

(6) の解答群

ア. JMI ERR

イ. JMI NEXT

ウ. JPL ERR

エ. JPL NEXT

(7) の解答群

ア. SLL GR0,1

イ. SLL GR0,4

ウ. SLL GR2,1

エ. SLL GR2,4

