

平成 31 年度 春期
基本情報技術者試験
午後 問題

試験時間 13:00 ~ 15:30 (2 時間 30 分)

注意事項

1. 試験開始及び終了は、監督員の時計が基準です。監督員の指示に従ってください。
2. 試験開始の合図があるまで、問題冊子を開いて中を見てはいけません。
3. 答案用紙への受験番号などの記入は、試験開始の合図があってから始めてください。
4. 問題は、次の表に従って解答してください。

問題番号	問 1	問 2 ~ 問 7	問 8	問 9 ~ 問 13
選択方法	必須	4 問選択	必須	1 問選択

5. 答案用紙の記入に当たっては、次の指示に従ってください。
 - (1) 答案用紙は光学式読取り装置で読み取った上で採点しますので、B 又は HB の黒鉛筆で答案用紙のマークの記入方法のとおりマークしてください。マークの濃度がうすいなど、マークの記入方法のとおり正しくマークされていない場合は読み取れないことがあります。特にシャープペンシルを使用する際には、マークの濃度に十分注意してください。訂正の場合は、あとが残らないように消しゴムできれいに消し、消しくずを残さないでください。
 - (2) 受験番号欄に受験番号を、生年月日欄に受験票の生年月日を記入及びマークしてください。答案用紙のマークの記入方法のとおり記入及びマークされていない場合は、採点されないことがあります。生年月日欄については、受験票の生年月日を訂正した場合でも、訂正前の生年月日を記入及びマークしてください。

[問 3, 問 4, 問 6, 問 7, 問 9 を
選択した場合の例]

選択欄							
問 1	<input type="radio"/>	問 2	<input checked="" type="radio"/>	問 8	<input type="radio"/>	問 9	<input type="radio"/>
		問 3	<input checked="" type="radio"/>			問 10	<input checked="" type="radio"/>
		問 4	<input checked="" type="radio"/>			問 11	<input checked="" type="radio"/>
		問 5	<input checked="" type="radio"/>			問 12	<input checked="" type="radio"/>
		問 6	<input checked="" type="radio"/>			問 13	<input checked="" type="radio"/>
		問 7	<input checked="" type="radio"/>				

- (3) 選択した問題については、次の例に従って、選択欄の問題番号の (選) をマークしてください。答案用紙のマークの記入方法のとおりマークされていない場合は、採点されません。問 2~問 7 について 5 問以上マークした場合は、はじめの 4 問を採点します。問 9~問 13 について 2 問以上マークした場合は、はじめの 1 問を採点します。

- (4) 解答は、次の例題にならって、解答欄にマークしてください。答案用紙のマークの記入方法のとおりマークされていない場合は、採点されません。

【例題】 次の に入れる正しい答えを、解答群の中から選べ。

春の情報処理技術者試験は、 a 月に実施される。

解答群 ア 2 イ 3 ウ 4 エ 5

正しい答えは“ウ 4”ですから、次のようにマークしてください。

例題	a	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
----	---	-----------------------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

裏表紙の注意事項も、
必ず読んでください。

【問題一覧】

●問1（必須問題）

問題番号	出題分野	テーマ
問1	情報セキュリティ	クラウドサービスの利用者認証

●問2～問7（6問中4問選択）

問題番号	出題分野	テーマ
問2	ソフトウェア	仮想記憶方式
問3	データベース	定期健康診断のデータが登録されているデータベース
問4	ネットワーク	eラーニングシステムの構成変更
問5	ソフトウェア設計	農産物の検査管理システム
問6	プロジェクトマネジメント	社内システムの仕様変更の扱い
問7	システム戦略	製造業における情報システムの統合

●問8（必須問題）

問題番号	出題分野	テーマ
問8	データ構造及びアルゴリズム	ハフマン符号化を用いた文字列圧縮

●問9～問13（5問中1問選択）

問題番号	出題分野	テーマ
問9	ソフトウェア開発（C）	入力ファイル中の文字の出現回数の印字
問10	ソフトウェア開発（COBOL）	学習塾のテスト結果の印字
問11	ソフトウェア開発（Java）	迷路と迷路上を移動する駒
問12	ソフトウェア開発（アセンブラ）	ビット列に対する操作
問13	ソフトウェア開発（表計算）	販売促進活動における販売データの利用

共通に使用される擬似言語の記述形式

擬似言語を使用した問題では、各問題文中に注記がない限り、次の記述形式が適用されているものとする。

[宣言、注釈及び処理]

記述形式	説明
○	手続、変数などの名前、型などを宣言する。
/* 文 */	文に注釈を記述する。
・変数 ← 式	変数に式の値を代入する。
・手続(引数, …)	手続を呼び出し、引数を受け渡す。
↑ 条件式 ↓ 処理	単枝選択処理を示す。 条件式が真のときは処理を実行する。
↑ 条件式 ↓ 処理 1 ↓ 処理 2	双枝選択処理を示す。 条件式が真のときは処理 1 を実行し、偽のときは処理 2 を実行する。
■ 条件式 ■ 処理	前判定繰返し処理を示す。 条件式が真の間、処理を繰り返し実行する。
■ 処理 ■ 条件式	後判定繰返し処理を示す。 処理を実行し、条件式が真の間、処理を繰り返し実行する。
■ 変数: 初期値, 条件式, 増分 ■ 処理	繰返し処理を示す。 開始時点で変数に初期値 (式で与えられる) が格納され、条件式が真の間、処理を繰り返す。また、繰り返すごとに、変数に増分 (式で与えられる) を加える。

〔演算子と優先順位〕

演算の種類	演算子	優先順位
単項演算	+, -, not	高 ↑ ↓ 低
乗除演算	×, ÷, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

注記 整数同士の除算では、整数の商を結果として返す。%演算子は、剰余算を表す。

〔論理型の定数〕

true, false

次の問 1 は必須問題です。必ず解答してください。

問 1 クラウドサービスの利用者認証に関する次の記述を読んで、設問 1, 2 に答えよ。

A 社では現在、Web ベースの業務システムが複数稼働しており、それぞれが稼働するサーバ（以下、業務システムサーバという）を社内 LAN に設置している。A 社のネットワーク構成を、図 1 に示す。

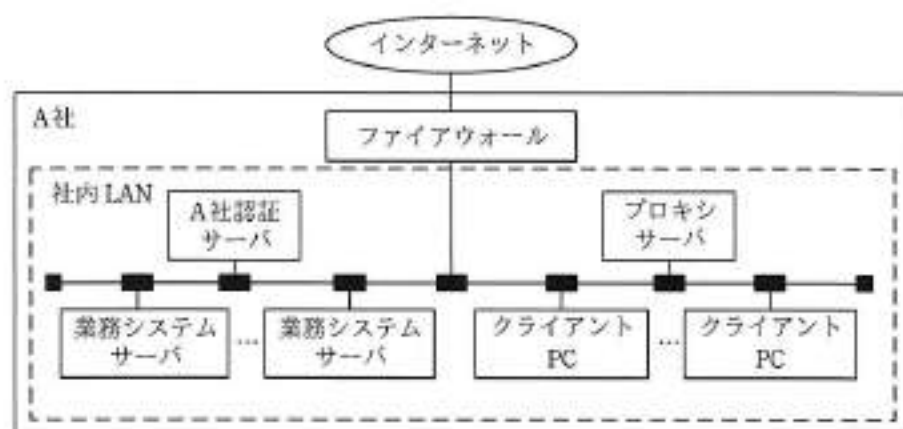


図 1 A 社のネットワーク構成

利用者は、業務システムを、社内 LAN に設置されたクライアント PC の Web ブラウザから利用する。社外から社内 LAN へのリモートアクセスは禁止されている。業務システムの利用者認証は、A 社認証サーバでの利用者 ID とパスワード（以下、この二つを併せて利用者認証情報という）の検証によって行っており、シングルサインオンを実現している。

社内 LAN からインターネットを介した社外への通信は、クライアント PC からプロキシサーバを経由した、HTTP over TLS（以下、HTTPS という）による通信だけが、ファイアウォールによって許可されている。社外からインターネットを介した社内 LAN への通信は、全てファイアウォールによって禁止されている。ファイアウォールの設定は、A 社のセキュリティポリシーに基づき変更しないものとする。

【クラウドサービスの利用者認証】

このたび A 社は、業務システムの一つである販売管理システムを、B 社がインターネットを介して提供する販売管理サービス（以下、B 社クラウドサービスという）に移行することにした。利用者認証に関しては、A 社認証サーバと B 社クラウドサービスを連携し、次の(1)～(3)を実現することにした。

- (1) B 社クラウドサービスをシングルサインオンの対象とする。
- (2) A 社の利用者認証は、B 社クラウドサービスについても、A 社認証サーバで行う。
- (3) 利用者が本人であることを確認するために A 社認証サーバで用いる は、B 社クラウドサービスには送信しない。

(1)～(3)を実現するために、A 社は、利用者認証を仲介する ID プロバイダ（以下、IdP という）を社内 LAN に設置することにした。IdP は、認証結果、認証有効期限及び利用者 ID（以下、これら三つを併せて認証済情報という）にデジタル署名を付加してから、Web ブラウザを介して、B 社クラウドサービスに送信する。B 社クラウドサービスは、付加されているデジタル署名を使って、受信した認証済情報に がないことを検証する。このために、IdP の を B 社クラウドサービスに登録しておく。

Web ブラウザと B 社クラウドサービスとの間、及び Web ブラウザと IdP との間の通信には、HTTPS を用いる。IdP と A 社認証サーバとの間の通信には LDAP を用いる。

【B 社クラウドサービスが利用可能になるまでの処理の手順】

A 社の利用者が、利用者認証されていない状態で、B 社クラウドサービスを利用しようとした場合に、利用可能になるまでの処理の手順を次の①～⑩に示す。

- ① 利用者は、Web ブラウザから B 社クラウドサービスにアクセスの要求を送信する。
- ② B 社クラウドサービスは、アクセスの要求を IdP に転送する指示（以下、転送指示という）を、Web ブラウザに返信する。

- ③ Web ブラウザは、②の転送指示に従い、IdP にアクセスの要求を送信する。
- ④ IdP は、利用者認証情報の入力画面を Web ブラウザに返信する。
- ⑤ 利用者は、Web ブラウザで利用者認証情報を入力する。Web ブラウザは、入力された利用者認証情報を IdP に送信する。
- ⑥ IdP は、利用者認証情報を A 社認証サーバに送信する。
- ⑦ A 社認証サーバは、利用者認証情報を検証し、認証結果を IdP に返信する。
- ⑧ IdP は、認証結果が成功の場合に、認証済情報を発行し、当該情報の B 社クラウドサービスへの転送指示とともに、Web ブラウザに返信する。
- ⑨ Web ブラウザは、⑧の転送指示に従い、認証済情報を B 社クラウドサービスに送信する。
- ⑩ B 社クラウドサービスは、認証済情報に基づいて、B 社クラウドサービスの利用を許可し、操作画面を Web ブラウザに返信する。

B 社クラウドサービスが利用可能になるまでの処理の流れを、図 2 に示す。図 2 中の①～⑩は、処理の手順の①～⑩と対応している。

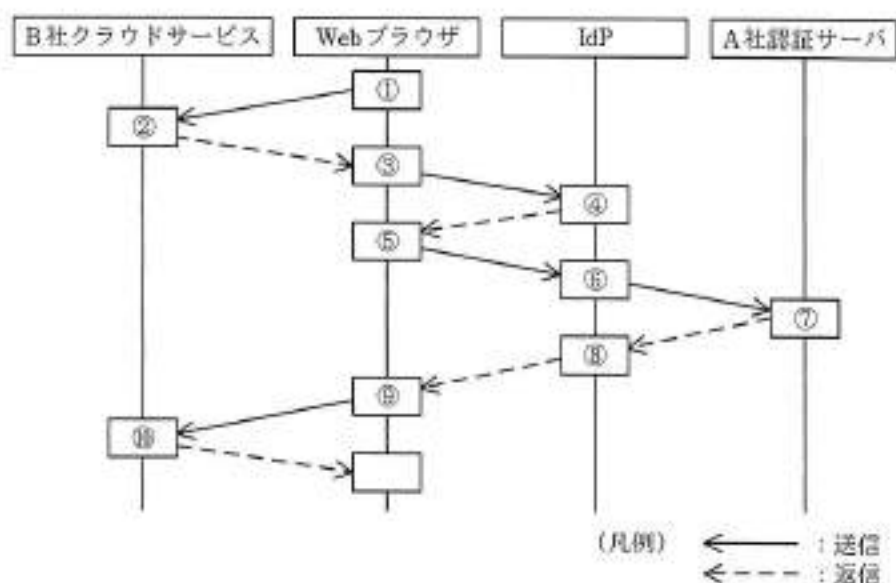


図 2 B 社クラウドサービスが利用可能になるまでの処理の流れ

設問1 本文中の に入れる適切な答えを、解答群の中から選べ。

a～cに関する解答群

- | | | |
|----------|----------|---------|
| ア PKI | イ 改ざん | ウ 公開鍵 |
| エ サービス妨害 | オ 生体情報 | カ パスワード |
| キ 秘密鍵 | ク 利用者 ID | ケ 漏えい |

設問2 次の記述中の に入れる適切な答えを、解答群の中から選べ。

B社クラウドサービスでは、接続元の IP アドレスを A 社のものに限定する機能は提供されていない。しかし、他の業務システムと同様に、B社クラウドサービスを、社内 LAN からの利用に限定できる。

この理由は、 d が必要であるが、IdP を社内 LAN に設置するので、社外から B社クラウドサービスを利用しようとしても、図 2 中の e の送信で失敗し、利用者認証されないからである。

dに関する解答群

- ア B社クラウドサービスが、IdP と直接通信する
- イ B社クラウドサービスが、利用者認証情報を検証し、Web ブラウザに返信する
- ウ IdP が、利用者に代わって、利用者認証情報を B社クラウドサービスに送信する
- エ Web ブラウザが、IdP と通信する

eに関する解答群

- | | | | | |
|-----|-----|-----|-----|-----|
| ア ① | イ ③ | ウ ⑤ | エ ⑥ | オ ⑧ |
|-----|-----|-----|-----|-----|

次の問2から問7までの6問については、この中から4問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、5問以上マークした場合には、はじめの4問について採点します。

問2 仮想記憶方式に関する次の記述を読んで、設問1～3に答えよ。

仮想記憶方式は、OSが提供する論理的な記憶領域（以下、仮想記憶という）上のアドレスと物理的な主記憶上のアドレスを対応付けて管理する方式である。仮想記憶方式では、補助記憶装置を仮想記憶の実装媒体として用いることによって、プログラムが主記憶の容量を超える大きさであっても、これを仮想記憶上のデータとして格納し、実行することができる。仮想記憶上のアドレス空間を仮想アドレス空間、主記憶上のアドレス空間を物理アドレス空間と呼び、それぞれの空間における記憶場所は仮想アドレス、物理アドレスで指定する。

仮想記憶方式の実現方法の一つにページング方式がある。この方式では、仮想アドレス空間と物理アドレス空間をそれぞれ仮想ページ、物理ページと呼ぶ固定長の領域に分割し、管理する。ページング方式では、プログラムの実行過程で、実行に必要な仮想ページのデータが物理アドレス空間に存在していないときは、そのデータが格納されている仮想ページからデータを物理ページに読み込んで利用する。

ページング方式のページ管理方法の例を次の(1)～(3)に示す。

- (1) 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、1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ぶ。
- (2) 仮想ページと物理ページの対応は、ページテーブルで管理する。ページテーブルの各要素は仮想ページと1対1に対応付けられており、要素の個数は仮想ページの個数と同じである。
- (3) ページテーブルには、仮想ページのデータが物理アドレス空間に存在しているかどうかを示すビット（以下、存在ビットという）と、存在している場合に対応する物理ページ番号を登録する領域がある。存在ビットは、当該仮想ページのデータが物理アドレス空間に存在している場合は1、存在していない場合は0である。

仮想アドレス空間の仮想ページと物理アドレス空間の物理ページとの対応例を、

図 1 に示す。ここで、図 1 中の A～H は、仮想ページ及び物理ページに格納されているデータを示す。

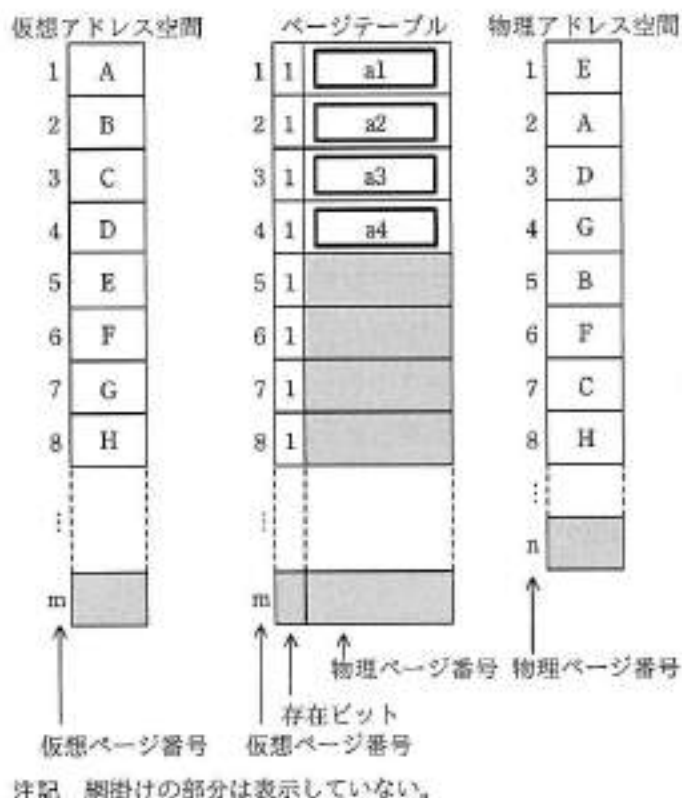


図 1 仮想アドレス空間の仮想ページと物理アドレス空間の物理ページとの対応例

設問 1 図 1 中の に入れる正しい答えを、解答群の中から選べ。ここで、a1～a4 に入れる答えは、a に関する解答群の中から組合せとして正しいものを選ぶものとする。

a に関する解答群

	a1	a2	a3	a4
ア	1	2	3	4
イ	2	3	5	7
ウ	2	5	7	3
エ	5	1	4	7
オ	5	2	7	3

設問2 次の記述中の に入れる適切な答えを、解答群の中から選べ。ここで、b1 と b2, c1 と c2 に入れる答えは、それぞれ b, c に関する解答群の中から組合せとして適切なものを選ぶものとする。

ページング方式では、プログラムの実行過程で、実行に必要なデータが物理アドレス空間に存在していないときには、ページフォールトという割込みが発生する。ページフォールトが発生すると、仮想ページに格納されているデータを物理ページに読み込む処理（以下、ページフォールト割込み処理という）を行う。

〔ページフォールト割込み処理〕

- (1) 物理ページのうち、仮想ページと対応付けられていない物理ページ（以下、空きページという）を一つ探す。
- (2) 空きページがなかった場合、空きページにする b1 ページを一つ選び、その選んだページに格納されているデータを b2 。その後、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。これによって空きページを確保する。
- (3) 空きページがあった場合、又はなかった場合では(2)の処理後に、その空きページにプログラムの実行に必要な c1 ページに格納されているデータを c2 。その後、対応するページテーブルの要素に物理ページ番号を登録して、その存在ビットを1にする。

bに関する解答群

	b1	b2
ア	仮想	補助記憶装置から読み込む
イ	仮想	補助記憶装置に書き出す
ウ	物理	補助記憶装置から読み込む
エ	物理	補助記憶装置に書き出す

cに関する解答群

	c1	c2
ア	仮想	補助記憶装置から読み込む
イ	仮想	補助記憶装置に書き出す
ウ	物理	補助記憶装置から読み込む
エ	物理	補助記憶装置に書き出す

設問3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

ページング方式において、ページフォールトの発生回数を少なくするためには、どの物理ページを空きページにするかが重要になる。一般的に知られているアルゴリズムである FIFO と LRU について、あるプログラムの実行過程で仮想ページが次の順で参照される場合を考える。ここで、プログラムの実行開始時点では、このプログラムの実行のために割り当てられた物理ページは全て空きページである。

[仮想ページの参照順]

4 → 2 → 1 → 5 → 4 → 2 → 3 → 4 → 2 → 1 → 5 → 3 → 5

FIFO を用いると、割り当てられた物理ページの個数が 3 のときは、ページフォールトの発生回数は 9 回である。物理ページの個数が 4 ならば、物理ページの個数が 3 のときと比べて、ページフォールトの発生回数は d

一方、LRU を用いると、物理ページの個数が 3 のときは、ページフォールトの発生回数は 10 回である。物理ページの個数が 4 ならば、物理ページの個数が 3 のときと比べて、ページフォールトの発生回数は e

d, eに関する解答群

- | | | |
|---------|---------|---------|
| ア 1回増える | イ 1回減る | ウ 2回増える |
| エ 2回減る | オ 3回増える | カ 3回減る |
| キ 4回増える | ク 4回減る | ケ 変わらない |

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問3 定期健康診断のデータが登録されているデータベースに関する次の記述を読んで、設問1～3に答えよ。

D 中学校では、年に1回、定期健康診断を実施し、結果をデータベースに登録している。

身長と体重の測定結果は、身長の単位は cm、体重の単位は kg として、いずれも 0.1 刻みで健診結果表に登録している。定期健康診断を受ける生徒の生年月日、性別などの属性情報は、受診者情報表に登録している。

健診結果表を作成する SQL 文の一部と、受診者情報表を作成する SQL 文を次に示す。

```
CREATE TABLE 健診結果表 (
  年度 CHAR(4),
  受診者ID CHAR(10),
  学年 INT,
  身長 DECIMAL(4, 1),
  体重 DECIMAL(4, 1),
  :
  肺活量 INT,
  PRIMARY KEY(年度, 受診者ID),
  FOREIGN KEY(受診者ID)
  REFERENCES 受診者情報表(受診者ID)
)
```

```
CREATE TABLE 受診者情報表 (
  受診者ID CHAR(10),
  氏名 VARCHAR(50),
  ふりがな VARCHAR(50),
  生年月日 DATE,
  性別 SMALLINT,
  特記事項 VARCHAR(512),
  PRIMARY KEY(受診者ID)
)
```

ここで、DECIMAL(x , y) は固定小数点数を扱うデータ型であり、 x は精度であって、表示する数字の桁数を示し、 y は位取りであって、小数点以下の数字の桁数を示す。例えば、データベースに 175.5 を登録するには、 x に 4 を、 y に 1 を指定する。

健診結果表及び受診者情報表のデータ（以下、定期健康診断データという）から、身長と体重を統計データとして参照するために、測定結果ビューを使用する。測定結果ビューを作成する SQL 文を次に示す。

```
CREATE VIEW 測定結果ビュー(年度, 学年, 性別, 生年月日, 身長, 体重) AS
SELECT 健診結果表.年度, 健診結果表.学年,
CASE 受診者情報表.性別
WHEN 0 THEN '男'
WHEN 1 THEN '女'
ELSE '-'
END,
受診者情報表.生年月日, 健診結果表.身長, 健診結果表.体重
FROM 健診結果表, 受診者情報表
WHERE 健診結果表.受診者ID = 受診者情報表.受診者ID
```

設問1 2010年度から2019年度までの定期健康診断データを基に、男子生徒の体格の推移を見る。次の SQL 文の実行結果を用いて、図1に示す体格推移表を作成した。平均身長と平均体重は小数第2位を四捨五入した値である。SQL 文の に入れる正しい答えを、解答群の中から選べ。ここで、c1 と c2 に入れる答えは、c に関する解答群の中から組合せとして正しいものを選ぶものとする。

```
SELECT 年度, 学年, 性別,
AVG(身長) AS 平均身長, MAX(身長) AS 最大身長, MIN(身長) AS 最小身長,
AVG(体重) AS 平均体重, MAX(体重) AS 最大体重, MIN(体重) AS 最小体重
FROM 測定結果ビュー WHERE 年度  a AND 性別 =  b
GROUP BY 年度, 学年, 性別
ORDER BY 年度  c1, 学年  c2
```

年度	学年	性別	平均身長	最大身長	最小身長	平均体重	最大体重	最小体重
2010	1	男	153.2	169.5	140.2	47.3	71.3	42.1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2019	1	男	152.8	168.2	140.4	48.1	73.1	43.6
2019	2	男	153.2	179.9	142.8	53.2	89.6	45.6
2019	3	男	154.1	180.1	139.3	50.4	93.2	41.1

注記 行の並び順は SQL 文の出力順と同じである。

図1 体格推移表

aに関する解答群

- ア LIKE '201_' イ LIKE '2010' ウ LIKE '2019'
 エ = '201%' オ = '201_' カ = '201*'

bに関する解答群

- ア 0 イ 1 ウ '男'
 エ '女' オ '-'

cに関する解答群

	c1	c2
ア	ASC	ASC
イ	ASC	DESC
ウ	DESC	ASC
エ	DESC	DESC

設問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

D 中学校では、転入生を受け入れる場合、転出元の中学校から受領した定期健康診断の結果の情報をデータベースに登録する。健診結果表には、受診者 ID と年度に d 制約を設定し、受診者 ID に受診者情報表の受診者 ID への e 制約を設定している。この制約に従うと、転入生の定期健康診断の結果の情報の健診結果表への登録は、受診者情報表に転入生の情報を登録した後に行う必要がある。

d, eに関する解答群

- ア UNIQUE イ 検査 ウ 参照
 エ 主キー オ 非 NULL

設問3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

図2に示す身長階級表を作成し、これに対応させて、2019年度の男子生徒の定期健康診断データを基に、図3に示す身長度数分布表を作成したところ、“人数が相違している”との指摘があった。

身長階級	階級下限	階級上限
140cm未満	0.0	139.9
140cm以上150cm未満	140.0	149.9
150cm以上160cm未満	150.0	159.9
160cm以上170cm未満	160.0	169.9
170cm以上180cm未満	170.0	179.9
180cm以上190cm未満	180.0	189.9
190cm以上	190.0	999.9

図2 身長階級表

身長階級	人数
140cm未満	3
140cm以上150cm未満	104
150cm以上160cm未満	302
160cm以上170cm未満	182
170cm以上180cm未満	32
180cm以上190cm未満	1
190cm以上	1

図3 身長度数分布表

図3に示す身長度数分布表の作成は、次の(1)～(3)の手順で行った。

- (1) 図2に示す身長階級表をデータベースに作成した。
- (2) 新たなSQL文を作成した。
- (3) (2)で作成したSQL文の実行結果を用いて、図3に示す身長度数分布表を作成した。

調査の結果、(2)で作成したSQL文に誤りがあることが分かった。(2)で作成したSQL文は、次のとおりである。このSQL文は、対象とした全生徒を必ずいずれかの身長区分に分類して集計するために、身長階級表からは全レコードを、測定結果ビューからはON句に指定した結合条件に一致するレコードを抽出している。

```
SELECT 身長階級, 階級下限, COUNT(+) AS 人数
FROM 測定結果ビュー RIGHT OUTER JOIN 身長階級表 ON
    身長 BETWEEN 階級下限 AND 階級上限
    AND 年度 = '2019'
    AND 性別 = '男'
GROUP BY 身長階級, 階級下限, 階級上限
ORDER BY 階級下限 ASC
```


図 1 を見ると、2019 年度は男子全学年での最大身長が 180.1cm であり、190cm 以上の生徒は存在しないが、図 3 の身長度数分布表では、190cm 以上の人数が 1 となっている。この原因は、(2)で作成した SQL 文では f とすべき箇所を COUNT(*)としていることにある。その結果、測定結果ビューと身長階級表を結合した結果において、NULL となっているレコードもカウントしているからである。

fに関する解答群

- | | | |
|-------------|---------------|-----------|
| ア COUNT(身長) | イ COUNT(身長階級) | ウ MAX(身長) |
| エ MAX(身長階級) | オ 身長 | カ 身長階級 |

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問4 eラーニングシステムの構成変更に関する次の記述を読んで、設問1～3に答えよ。

G社は、全国に設置した様々な規模の教室から教育コンテンツにアクセスできるWebベースのeラーニングシステムを構築し、このシステムを使った教育事業を展開している。

eラーニングシステムは、1台のコンテンツサーバと1台のアプリケーションサーバで構成されている。コンテンツサーバは、教材や試験問題などの教育コンテンツを保持し、アプリケーションサーバを経由して、クライアントである教室のPCに教育コンテンツを送信する。

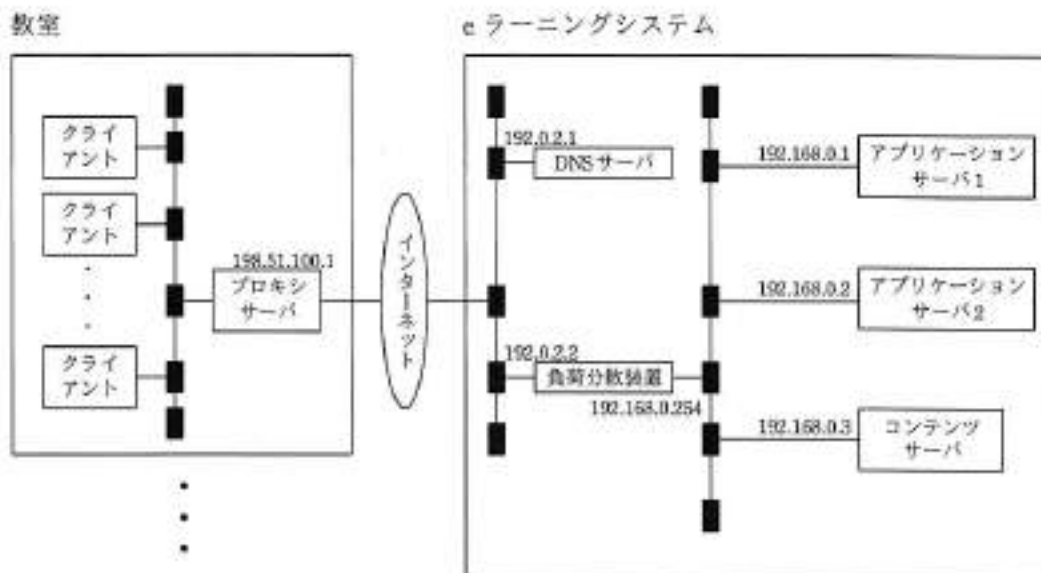
アプリケーションサーバは、受講者の認証を行い、ログインしている受講者を管理する。

受講者は、クライアントを利用して、教室ごとに設置されているプロキシサーバとインターネットを経由して、eラーニングシステムにアクセスして学習する。

最近、一部の受講者から、システム利用に関して、“応答に時間が掛かる”などの苦情が寄せられている。G社では、応答時間を短縮するために、アプリケーションサーバ1台の追加と負荷分散装置の導入を伴う新しいネットワーク構成を検討した。負荷分散装置は、クライアントからの要求を、同じ機能をもつ複数のサーバのうちのいずれかに振り分ける装置である。

[検討したネットワーク構成]

検討したネットワーク構成を、図1に示す。



注記 xx.xxxは、IPアドレスを表す。

図1 検討したネットワーク構成

クライアントからプロキシサーバを経由して、eラーニングシステムにアクセスするために、DNSサーバ及び負荷分散装置には次の設定を行う。

- ・ DNSサーバに、eラーニングシステムのドメイン名とこれに対応するIPアドレスとして とを登録する。
- ・ 負荷分散装置に、振り分け先IPアドレスとして とを登録する。

[負荷分散装置を用いたアクセスの振り分け]

アプリケーションサーバは、ログインしている受講者を管理し、その受講者がどの教育コンテンツを閲覧中かなどの状況を保持する。したがって、負荷分散装置を用いてアプリケーションサーバの負荷分散を行う場合には、受講者がeラーニングシステムにログインしてからログアウトするまでは、その受講者が利用する1台のクライアントからアプリケーションサーバへの要求を、常に同一のアプリケーションサーバへ振り分ける必要がある。負荷分散装置には、送信元のIPアドレスの情報を基に要求を振り分けるタイプ（以下、装置タイプAという）と、OSI基本参照モデルのレイヤ4以上の情報を基に要求を振り分けるタイプ（以下、装置タイプBという）とがある。二つのタイプそれぞれの装置の動作について、概要を次に示す。

(1) 装置タイプ A

- (a) 振り分け先が決まっていない送信元 IP アドレスからの要求は、ラウンドロビン方式で決定したアプリケーションサーバに振り分けるとともに、送信元 IP アドレスと振り分け先のアプリケーションサーバの IP アドレスとを記録する。
- (b) 振り分け先が決まっている送信元 IP アドレスからの要求は、そのアプリケーションサーバに振り分ける。

装置タイプ A を用いると、①多くのクライアントのある大規模な教室からのアクセスが、1 台のアプリケーションサーバに集中して、アプリケーションサーバの負荷に偏りが生じることが予想される。

(2) 装置タイプ B

- (a) クライアントから送信された要求中の HTTP ヘッダ内に c (以下、識別情報という) がない場合は、ラウンドロビン方式で決定したアプリケーションサーバに振り分ける。
- (b) アプリケーションサーバから送信された応答に含まれる HTTP ヘッダ内の識別情報と、当該アプリケーションサーバの IP アドレスとを記録する。
- (c) クライアントから送信された要求中の HTTP ヘッダ内の識別情報に対応するアプリケーションサーバの IP アドレスが (b) の処理によって記録されている場合は、そのアプリケーションサーバに振り分ける。

装置タイプ B を用いると、多くのクライアントのある大規模な教室からプロキシサーバを経由してアクセスがあっても、振り分け先の決定をクライアント単位で行える。

検討の結果、アプリケーションサーバの負荷に偏りが少なくなることから、装置タイプ B を導入することにした。

設問1 本文中の に入れる適切な答えを、解答群の中から選べ。

aに関する解答群

- | | | |
|---------------|---------------|----------------|
| ア 192.0.2.1 | イ 192.0.2.2 | ウ 192.168.0.1 |
| エ 192.168.0.2 | オ 192.168.0.3 | カ 198.51.100.1 |

bに関する解答群

- | | |
|-----------------------------|-------------------------------|
| ア 192.168.0.1 と 192.168.0.2 | イ 192.168.0.1 と 192.168.0.3 |
| ウ 192.168.0.2 と 192.168.0.3 | エ 192.168.0.2 と 192.168.0.254 |

cに関する解答群

- ア DNS サーバの IP アドレス
- イ アプリケーションサーバとコンテンツサーバのポート番号
- ウ セッション ID を示す cookie
- エ 負荷分散装置の IP アドレス

設問2 本文中の下線部①で、装置タイプ A を用いたときに、アプリケーションサーバの負荷に偏りが生じる要因となり得るものはどれか。適切な答えを、解答群の中から選べ。

解答群

- ア 同じ教室のどのクライアントからの要求も送信元 IP アドレスが全て同じになること。
- イ クライアントから送られてきた IP パケット内の送信先アドレスが変換されること。
- ウ 負荷分散装置が送信元 IP アドレスの情報を用いるだけの単純な機能なので、高速に動作すること。
- エ プロキシサーバを経由しても、HTTP ヘッダの情報が変更されないこと。

設問3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

e ラーニングシステムの応答時間は、1 件の要求に対するネットワーク上のデータ転送時間、アプリケーションサーバでの処理待ち時間、及びアプリケーションサーバの処理時間（処理を始めてから応答を返すまでの時間）の合計である。検討したネットワーク構成での応答時間の短縮効果を評価するために、アプリケーションサーバでの平均処理待ち時間を計算する。

アプリケーションサーバの要求 1 件の平均処理時間は 0.40 秒であり、要求の平均到着率は 2.30 件/秒である。処理の待ち行列はアプリケーションサーバだけで発生するものとして、現行のネットワーク構成における平均処理待ち時間を、次の M/M/1 の待ち行列モデルの式で計算すると、4.60 秒となる。

$$\text{平均処理待ち時間} = \frac{\rho}{1-\rho} \times \text{平均処理時間}$$

ρ : アプリケーションサーバの利用率 = λ / μ

μ : アプリケーションサーバの処理率 (=平均処理時間の逆数)

λ : アプリケーションサーバへの平均到着率

アプリケーションサーバを 1 台追加し、装置タイプ B を導入したとき、1 台のアプリケーションサーバでの平均処理待ち時間を M/M/1 の待ち行列モデルによって計算すると、 秒となる。ここで、要求は 2 台のアプリケーションサーバに交互に振り分けられると仮定する。また、平均処理待ち時間は小数第 3 位を四捨五入し、ネットワーク上のデータ転送時間とアプリケーションサーバの処理時間は変わらないものとする。

解答群

ア 0.03

イ 0.34

ウ 1.34

エ 3.41

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問5 農産物の検査管理システムに関する次の記述を読んで、設問1～3に答えよ。

K社は、農産物を検査し、結果を報告する検査業務を実施している。検査の申込者の情報、検査を依頼された農産物（以下、検査農産物という）の情報、及び検査項目ごとの検査結果の情報は、検査管理システムで管理する。

[検査農産物の検査業務の流れ]

検査農産物の検査業務の流れは、次のとおりである。

- (1) 担当者は、申込者から検査農産物と検査項目を受け付ける。一つの検査農産物に対して五つまでの検査項目を受け付ける。
- (2) 担当者は、申込者が新規の場合、申込者の情報を検査管理システムに登録する。
- (3) 担当者は、検査農産物の情報を検査管理システムに登録する。ただし、検査項目は登録しない。
- (4) 担当者は、受け付けた検査項目の検査を実施する。受け付けた全ての検査項目の検査が終了した後、検査結果の情報を、検査項目ごとに検査管理システムに一括して登録する。
- (5) 担当者は、検査管理システムに登録された検査結果を確認して、申込者に検査結果を報告する。

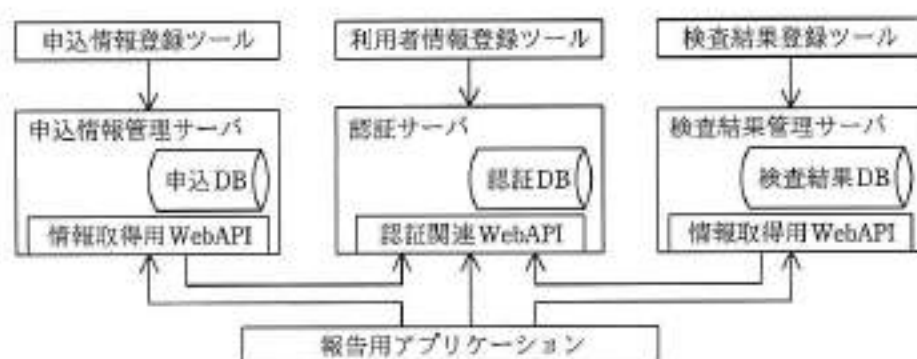
[検査管理システムの説明]

検査管理システムの構成を、図1に示す。検査管理システムは3台のサーバから成り、各サーバは個別にデータベース（以下、DBという）をもつ。各DBへの情報の登録には、各サーバの申込情報登録ツール、利用者情報登録ツール及び検査結果登録ツールを利用する。

申込者に報告する検査結果を確認するために、担当者は報告用アプリケーションを利用する。報告用アプリケーションは、各サーバが提供しているWebインタフェース

の API（以下、WebAPI という）である情報取得用 WebAPI と認証関連 WebAPI だけを利用して、各 DB にアクセスする。

担当者は、報告用アプリケーションに自身の利用者 ID とパスワードを入力して、検査管理システムにログインする。報告用アプリケーションは、ログイン処理に認証関連 WebAPI を利用し、ログインに成功したら認証サーバからアクセス用の ID（以下、アクセス ID という）を取得する。アクセス ID は、認証情報として認証 DB に格納され、ログアウトするまで管理される。報告用アプリケーションは、アクセス ID を情報取得用 WebAPI の引数に指定して、各 DB から必要な情報を取得する。



凡例

→: アクセスを示す。記載がないアクセスは発生しない。

図1 検査管理システムの構成

〔検査管理システムが管理する情報〕

検査管理システムが管理する情報と項目を、図2に示す。各情報は、個別の ID（利用者 ID、検査 ID など）を付与される。各情報は、いずれかの DB だけに格納されている。各サーバがもつ DB は、そのサーバが提供している情報取得用 WebAPI（表1参照）又は認証関連 WebAPI と、登録ツールだけからアクセスできる。よって、申込 DB に格納されている情報は、図2に示す情報のうち、**a** である。

<table border="1"> <tr><th>認証情報</th></tr> <tr><td>アクセス ID</td></tr> <tr><td>利用者 ID</td></tr> <tr><td>ログイン日時</td></tr> <tr><td>最終アクセス日時</td></tr> </table>	認証情報	アクセス ID	利用者 ID	ログイン日時	最終アクセス日時	<table border="1"> <tr><th>利用者情報</th></tr> <tr><td>利用者 ID</td></tr> <tr><td>パスワード</td></tr> </table>	利用者情報	利用者 ID	パスワード	<table border="1"> <tr><th>申込者情報</th></tr> <tr><td>申込者 ID</td></tr> <tr><td>団体名</td></tr> <tr><td>住所</td></tr> <tr><td>電話番号</td></tr> </table>	申込者情報	申込者 ID	団体名	住所	電話番号	<table border="1"> <tr><th>検査農産物情報</th></tr> <tr><td>検査農産物 ID</td></tr> <tr><td>b</td></tr> <tr><td>検査農産物名</td></tr> </table>	検査農産物情報	検査農産物 ID	b	検査農産物名	<table border="1"> <tr><th>検査結果情報</th></tr> <tr><td>検査 ID</td></tr> <tr><td>c</td></tr> <tr><td>検査項目名</td></tr> <tr><td>検査結果</td></tr> <tr><td>検査日</td></tr> </table>	検査結果情報	検査 ID	c	検査項目名	検査結果	検査日
認証情報																											
アクセス ID																											
利用者 ID																											
ログイン日時																											
最終アクセス日時																											
利用者情報																											
利用者 ID																											
パスワード																											
申込者情報																											
申込者 ID																											
団体名																											
住所																											
電話番号																											
検査農産物情報																											
検査農産物 ID																											
b																											
検査農産物名																											
検査結果情報																											
検査 ID																											
c																											
検査項目名																											
検査結果																											
検査日																											

注記 項目名の下線は主キーを示す。

図 2 検査管理システムが管理する情報と項目

〔情報取得用 WebAPI の説明〕

申込情報管理サーバと検査結果管理サーバが提供している情報取得用 WebAPI の説明を、表 1 に示す。

情報取得用 WebAPI は、指定された引数に基づいて処理を実行し、取得した情報を戻り値とする。

引数の指定方法には、情報取得用 WebAPI によって、個別の ID を指定する方法、ID のリストを指定する方法、開始番号を指定する方法の 3 通りがある。情報取得用 WebAPI の引数が開始番号の場合、格納されている情報を整列し、整列後の並びの先頭の情報を 1 件目とし、開始番号の位置から最多で 10 件を取得して戻り値とする。例えば、検査結果情報が 25 件ある場合、開始番号が 1 のときは整列後の並びの 1～10 件目を、開始番号が 11 のときは 11～20 件目を、開始番号が 21 のときは 21～25 件目を戻り値とする。

情報取得用 WebAPI は、処理開始時に、認証関連 WebAPI を利用して、引数に指定されたアクセス ID が認証情報として管理されているかどうかを確認する。管理されていない場合は、処理を受け付けない。ここで、表 1 中の WebAPI 仕様の処理の説明では、アクセス ID の確認処理の説明は省略している。

表1 情報取得用 WebAPI の説明

サーバ	WebAPI名	WebAPI仕様	
申込情報管理 サーバ	getOffererInfo	引数	アクセス ID, 申込者 ID のリスト (10 件まで指定可)
		戻り値	申込者情報のリスト
		処理	指定された申込者 ID の申込者情報を申込 DB から取得する。
	listOffererInfo	引数	アクセス ID, 開始番号
		戻り値	申込者情報のリスト
		処理	申込 DB に格納されている申込者情報を申込者 ID の昇順に整理し, 開始番号から最多で 10 件の申込者情報を取得する。
	getProductInfo	引数	アクセス ID, 検査農産物 ID のリスト (10 件まで指定可)
		戻り値	検査農産物情報のリスト
		処理	指定された検査農産物 ID の検査農産物情報を申込 DB から取得する。
listProductInfo	引数	アクセス ID, 開始番号	
	戻り値	検査農産物情報のリスト	
	処理	申込 DB に格納されている検査農産物情報を検査農産物 ID の昇順に整理し, 開始番号から最多で 10 件の検査農産物情報を取得する。	
検査結果管理 サーバ	getResultInfo	引数	アクセス ID, 検査 ID のリスト (10 件まで指定可)
		戻り値	検査結果情報のリスト
		処理	指定された検査 ID の検査結果情報を検査結果 DB から取得する。
	getResultInfo ForProduct	引数	アクセス ID, 検査農産物 ID
		戻り値	検査結果情報のリスト
		処理	指定された検査農産物 ID に対応する検査農産物の全ての検査結果情報を検査結果 DB から取得する。
	listResultInfo	引数	アクセス ID, 開始番号
		戻り値	検査結果情報のリスト
		処理	検査結果 DB に格納されている検査結果情報を検査日, 検査 ID の昇順に整理し, 開始番号から最多で 10 件の検査結果情報を取得する。

注記 WebAPI の処理では, 引数に指定した申込者 ID, 検査農産物 ID 又は検査 ID に対応する情報のうち, 各 DB に存在するものだけを戻り値とする。

設問1 本文中及び図2中の に入れる正しい答えを, 解答群の中から選べ。

a に関する解答群

- ア 認証情報及び申込者情報
- イ 認証情報及び利用者情報
- ウ 認証情報, 申込者情報及び検査農産物情報
- エ 検査農産物情報
- オ 申込者情報
- カ 申込者情報及び検査農産物情報

b, cに関する解答群

- ア アクセス ID イ 検査 ID ウ 検査項目名
エ 検査農産物 ID オ 申込者 ID カ 利用者 ID

設問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

報告用アプリケーションには農産物検査結果表示機能があり、この機能は、1件の検査農産物 ID を入力して“表示”ボタンをクリックすると、画面に情報を表示する。農産物検査結果表示機能が表示する画面の例を、図 3 に示す。検査結果の情報が登録済みの検査農産物の検査農産物 ID を入力した場合、“表示”ボタンをクリックしてから画面の表示が完了するまでには、表 1 に示す情報取得用 WebAPI の実行が最低 d 回必要である。ここで、実行時にエラーは発生しないものとする。

検査農産物ID： 20190301-001		検査農産物名： 水稻うるち玄米		
申込者情報		団体名： ○○団体		
		住所： ○○県○○市○○町1-1		
		電話番号： ○○○-○○○○-○○○○		
検査結果情報				
	検査ID	検査項目名	検査結果	検査日
1	0000984501	形質	一等標準品	2019-03-05
2	0000984502	整粒 (%)	70	2019-03-05
3	0000984503	水分 (%)	15.0	2019-03-06
4				
5				

図 3 農産物検査結果表示機能が表示する画面の例

dに関する解答群

- ア 2 イ 3 ウ 4 エ 5
オ 6 カ 7 キ 8

設問3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

担当者から、検査農産物 ID の検査結果を個別に閲覧するのではなく、一覧形式で閲覧したいという要望が挙がった。要望を受けて、報告用アプリケーションに検査結果リスト表示機能を追加する。

検査結果リスト表示機能では、表 1 に示す情報取得用 WebAPI を利用して、登録されている検査結果の情報（以下、農産物検査結果情報という）を全て一覧形式で表示する。1 行の表示内容は、各農産物検査結果情報の一覧全体中での位置を示す番号と、農産物検査結果情報の内訳として、団体名、検査農産物 ID、検査 ID、検査項目名、検査結果及び検査日である。一覧にすると、最近実施した検査の検査結果が先に表示されるように、検査日の降順に整列する。検査結果リスト表示機能が表示する画面の例を、図 4 に示す。ここで、実行時にエラーは発生しないものとする。

番号	団体名	検査農産物ID	検査ID	検査項目名	検査結果	検査日
1	□□□団体	20190401-001	0000989587	形質	一等標準品	2019-04-02
2	⋮	⋮	⋮	⋮	⋮	⋮
3	⋮	⋮	⋮	⋮	⋮	⋮
4	⋮	⋮	⋮	⋮	⋮	⋮
5	⋮	⋮	⋮	⋮	⋮	⋮
6	⋮	⋮	⋮	⋮	⋮	⋮
7	⋮	⋮	⋮	⋮	⋮	⋮
8	⋮	⋮	⋮	⋮	⋮	⋮
9	⋮	⋮	⋮	⋮	⋮	⋮
10	⋮	⋮	⋮	⋮	⋮	⋮

注記 番号の列には、開始番号から始まる連番を表示する。

図 4 検査結果リスト表示機能が表示する画面の例

検査結果リスト表示機能は、1 画面に最多で 10 件の農産物検査結果情報を表示する。画面に表示されていない農産物検査結果情報を表示するときは、ボタン“前へ”又は“次へ”をクリックして表示の対象を切り替える。

この機能では、初期表示時の開始番号を 1 として画面を表示する。ボタン

“前へ”又は“次へ”をクリックすると、開始番号に -10 又は 10 を加算して、画面を再表示する。画面表示のたびに、開始番号及び情報取得用 WebAPI を利用して表示する情報を取得する場合、情報取得用 WebAPI を e という順番で 1 回ずつ実行する。

eに関する解答群

- ア listOffererInfo, listProductInfo, getResultInfoForProduct
- イ listProductInfo, getOffererInfo, getResultInfoForProduct
- ウ listProductInfo, listOffererInfo, getResultInfoForProduct
- エ listResultInfo, getOffererInfo, getProductInfo
- オ listResultInfo, getProductInfo, getOffererInfo

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問6 社内システムの仕様変更の扱いに関する次の記述を読んで、設問1, 2に答えよ。

Q社では、ある社内システムを刷新する開発プロジェクト（以下、刷新プロジェクトという）を実施している。刷新プロジェクトは、設計、プログラム開発、結合試験及び総合試験の4工程で推進する。刷新プロジェクトは、利用部門、プロジェクト管理チーム、プロジェクトマネージャ（以下、担当PMという）及び複数の開発チームで構成されている。Q社では、過去の社内システムの構築において、開発中に発生した仕様変更に伴い、プロジェクト計画の大幅な見直しが必要になったことがあり、利用部門から仕様変更の依頼が発生した場合の取扱手順を定めている。

〔仕様変更の依頼が発生した場合の取扱手順〕

プロジェクト管理チーム、担当PM、開発チームの役割は、次のとおりである。

・プロジェクト管理チーム

仕様変更依頼票の受渡しなど利用部門との書類や情報のやり取り、担当PMや開発チームとの書類のやり取り、取りまとめを行う。また、変更管理会議の運営を行う。

・担当PM

仕様変更依頼票の確認、仕様変更の採否の判断、プロジェクト計画の更新を行う。

・開発チーム

仕様変更依頼票の内容に基づく影響調査を実施する。

- (1) 利用部門は、仕様変更の目的及び内容と、変更によって得られる効果を記述した仕様変更依頼票を起票し、に提出する。
- (2) は、仕様変更依頼票を受け付けて、受付番号を記入して、管理簿への記録を行う。
- (3) は、仕様変更依頼票の記述内容を確認し、内容が妥当であれば、受領し、変更内容に対する影響調査を該当するに対して依頼するように

□ a1 □ に指示する。ここで、プログラム開発以降の工程では、システム全体への影響を迅速に把握する必要があるので、Q社のルールでは、全ての□ a3 □ に対して影響調査を依頼する。受領しない場合、受領しない理由を仕様変更依頼票に付して利用部門に返却するよう □ a1 □ に指示する。

- (4) 依頼された □ a3 □ は、仕様変更依頼票に記述してある仕様変更の内容に基づき、自チームで開発している機能に関する影響を調査する。追加、変更などが必要な設計書のページ数、対象プログラム及び影響を受ける開発規模を調査し、影響調査結果を回答する。
- (5) □ a1 □ は、影響調査結果を取りまとめた、利用部門、□ a2 □ 及び影響のある □ a3 □ のメンバーを出席者とする変更管理会議を開催する。
- (6) 変更管理会議において、影響調査結果を参考にして、担当 PM は、仕様変更の依頼の重要性及び緊急性、納期遅延の可能性、工数増加の有無、変更によるプログラム品質への影響などを総合的に評価して、変更を採用するかどうかを判断する。
- (7) プロジェクト管理チームは、利用部門に仕様変更依頼の採否の結果を回答する。
- (8) 変更管理会議において、仕様変更依頼が採用された場合、担当 PM は □ b □ する。

設問1 [仕様変更の依頼が発生した場合の取扱手順] の記述中の □ □ □ に入れる適切な答えを、解答群の中から選べ。ここで、a1～a3 に入れる答えは、a に関する解答群の中から組合せとして適切なものを選ぶものとする。

aに関する解答群

	a1	a2	a3
ア	開発チーム	担当 PM	プロジェクト管理チーム
イ	担当 PM	開発チーム	プロジェクト管理チーム
ウ	担当 PM	プロジェクト管理チーム	開発チーム
エ	プロジェクト管理チーム	担当 PM	開発チーム

bに関する解答群

- ア 仕様変更依頼の内容に対する影響調査を指示
- イ 仕様変更依頼の内容をプロジェクト計画に反映させて更新
- ウ 仕様変更依頼票の起票に至る経緯をプロジェクト管理チームに確認
- エ 仕様変更依頼票の起票に至る経緯を利用部門に確認

設問2 次の記述中の に入れる適切な答えを、解答群の中から選べ。

刷新プロジェクトのスケジュールを、図1に示す。ここで、1か月は4週とする。

期間(月)	1	2	3	4	5	6	7	8	9	10
工程	設計		プログラム開発				結合試験		総合試験	

図1 刷新プロジェクトのスケジュール

スケジュールのとおりに進んでいったところ、刷新プロジェクトの開始後13週目において、機能Lに対する仕様変更依頼票が起票され、すぐに担当PMは受領した。刷新プロジェクトでは、機能Lを開発チーム1が、機能Mを開発チーム2が、機能Nを開発チーム3が担当している。現在の工程では、変更内容に対する影響調査の依頼先は、Q社のルールにのっとり、となる。

〔影響調査結果の概要〕

影響調査の結果、開発チーム1が担当している機能L（当初計画の開発規模は、400kステップ）だけに影響があることが分かった。機能Lに関する生産性及び当初計画の工数を、表1に示す。

表 1 機能 L に関する生産性及び当初計画の工数

工程	生産性 (k ステップ/人月)	工数 (人月)
設計	10	40
プログラム開発	4	100
結合試験	8	50
総合試験	10	40

仕様変更を行っても、開発を進めている機能 L のプログラムを変更する必要はなく、機能 L の当初計画の開発規模の 10% の追加開発が必要と分かった。また、仕様変更分における工程ごとの生産性は、当初計画と同じである。仕様変更に関わる、設計工程から結合試験工程までは、当初計画分と切り離して実施可能であり、この期間に必要な追加の工数は d 人月となる。

開発チーム 1 では、仕様変更分の総合試験の実施に関して、次の二つの計画を検討した。

・計画 1

仕様変更分の総合試験は、当初計画分の総合試験の開始時点から実施する。

・計画 2

仕様変更分の総合試験は、当初計画分の総合試験の 5 週目開始時点から実施する。

仕様変更に関わる、設計工程から結合試験工程までの期間は、刷新プロジェクトの開始後 17 週目開始から、計画 1 又は計画 2 の総合試験工程の開始前までとする。総合試験工程は、図 1 のスケジュールで示した当初計画どおりに完了させる。ここで、仕様変更分の開発に必要な追加の要員数は、設計工程から結合試験工程までの期間の各週で同一とする。また、追加の要員は、全ての工程を担当できるスキルを備えているものとする。

計画 1 の場合は刷新プロジェクトの開始後 e 週目終了までに、計画 2 の場合は刷新プロジェクトの開始後 36 週目終了までに、結合試験工程を完了させる必要がある。仕様変更に関わる、設計工程から結合試験工程までの期間の各週に必要な追加の要員は、計画 1 の場合は少なくとも f 人、

計画 2 の場合は少なくとも g 人である。追加可能な要員が各週とも最多 4 人である場合、h。

前記の影響調査結果も踏まえて変更管理会議を開催し、仕様変更依頼を採用することになった。

cに関する解答群

- ア 開発チーム 1
- イ 開発チーム 2
- ウ 開発チーム 3
- エ 開発チーム 1 及び開発チーム 2
- オ 開発チーム 1 及び開発チーム 3
- カ 開発チーム 2 及び開発チーム 3
- キ 全ての開発チーム

dに関する解答群

- | | | | |
|------|------|-------|-------|
| ア 19 | イ 23 | ウ 190 | エ 230 |
|------|------|-------|-------|

eに関する解答群

- | | | | |
|------|------|------|------|
| ア 17 | イ 31 | ウ 32 | エ 36 |
|------|------|------|------|

f, gに関する解答群

- | | | | |
|-----|-----|-----|-----|
| ア 1 | イ 2 | ウ 3 | エ 4 |
| オ 5 | カ 6 | キ 7 | ク 8 |

hに関する解答群

- ア 計画 1 だけが実現可能である
- イ 計画 2 だけが実現可能である
- ウ 両計画とも実現可能である
- エ 両計画とも実現不可能である

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問7 製造業における情報システムの統合に関する次の記述を読んで、設問1～3に答えよ。

T社は、大手家電製品メーカーであり、製品の設計及び需要予測は事業部ごとに行っている。各製造工場は、一つ以上の事業部の製品を製造している。

2018年度に各製造工場が製造した製品の事業部別取扱高を、表1に示す。

表1 2018年度に各製造工場が製造した製品の事業部別取扱高

単位 億円

製造工場	事業部			
	P事業部	Q事業部	R事業部	S事業部
H製造工場	—	200	—	50
I製造工場	500	—	—	—
J製造工場	—	—	—	150
K製造工場	—	—	300	—
L製造工場	—	200	—	—
M製造工場	—	—	—	100
合計	500	400	300	300

注記 “—”は、当該製造工場が当該事業部の製品を製造していないことを示す。

T社は、事業部と製造工場ごとに情報システムを整備してきた経緯があり、設計システムには三つ、需要予測システムには三つ、調達システムには四つのシステムがある。各事業部が利用している設計システム及び需要予測システムを表2に、各製造工場が利用している調達システムを表3に示す。システムの名称が同一の場合は、事業部又は製造工場が同じシステムを共用していることを意味している。例えば、P事業

部とS事業部は、設計システムとしてAシステムを共用している。

表2 各事業部が利用している設計システム及び需要予測システム

事業部	設計システム	需要予測システム
P事業部	Aシステム	Eシステム
Q事業部	Bシステム	Fシステム
R事業部	Cシステム	Fシステム
S事業部	Aシステム	Gシステム

表3 各製造工場が利用している調達システム

製造工場	調達システム
H製造工場	Vシステム
I製造工場	Wシステム
J製造工場	Xシステム
K製造工場	Yシステム
L製造工場	Xシステム
M製造工場	Xシステム

T社では情報システムの運用保守コストの削減が求められており、本社の情報システム部門の主導によってプロジェクトを立ち上げ、設計システム、需要予測システム及び調達システムのそれぞれについて、複数のシステムを統合することとした。

設問1 データの受渡し（以下、データ連携という）に関する次の記述中の

に入れる正しい答えを、解答群の中から選べ。

調達システムは、製品の製造に必要な部品の在庫管理と発注処理を行っている。各製造工場が利用している調達システムは、その工場が製造する製品を担当している事業部が利用している設計システム及び需要予測システムとデータ連携している。設計システムと需要予測システムとの間のデータ連携はない。

表1～3によれば、最も多くの調達システムとデータ連携している設計システムは システムであり、そのデータ連携先の調達システム数は である。また、調達システムの中で、データ連携している設計システム数と需要予測システム数の合計が最も多いのは、 システムと システムである。

aに関する解答群

ア A

イ B

ウ C

bに関する解答群

ア 2

イ 3

ウ 4

エ 5

オ 6

c, dに関する解答群

ア V

イ W

ウ X

エ Y

設問2 情報システムの運用保守コストに関する次の記述中の に入れる正しい答えを、解答群の中から選べ。

本社の情報システム部門による調査の結果、情報システムの運用保守コストとして、設計システムのそれぞれには年間2億円、需要予測システムのそれぞれには年間3億円、調達システムのそれぞれには年間1億円が掛かっていることが分かった。

それらに加えて、調達システムのそれぞれには、各製造工場が製造する製品の年間の取扱高の0.1%分の運用保守コストが掛かっていることが分かった。

また、システム間のデータ連携のための経路（以下、データ連携経路という）の運用保守コストとして、データ連携経路一つ当たり年間0.1億円が掛かっていることが分かった。今後においても、今までデータ連携していないシステム間に新たなデータ連携経路を一つ作成した場合、データ連携経路の運用保守コストは年間0.1億円増加する。

2018年度のT社の設計システム、需要予測システム、調達システム及びデータ連携経路の年間の運用保守コストの合計は 億円であった。設計システム、需要予測システム及び調達システムを全社でそれぞれ一つに統合できれば、これらのシステム及びデータ連携経路の年間の運用保守コストの合計は 億円となる。ここで、取扱高などの前提条件は変わらないものとする。また、一つのシステムを複数の事業部や製造工場で共用する場合でも、共用するシステムの運用保守コストは一つの事業部や製造工場が単独で利用する場合と変わらないものとする。

eに関する解答群

ア 19.0 イ 20.2 ウ 21.7 エ 25.2

fに関する解答群

ア 6.0 イ 6.2 ウ 7.7 エ 9.7

設問3 Gシステムの廃棄に関する次の記述中の に入れる適切な答えを、解答群の中から選べ。

本社の情報システム部門は、Gシステムの保守サポート期間の終了が迫っているため、Gシステムを廃棄し、他の需要予測システムによって代替できるかどうかを検討した。

新たなデータ連携経路を作成することなくGシステムとの代替が可能な需要予測システムは 。

gに関する解答群

ア Eシステムだけである イ Fシステムだけである
ウ Eシステム及びFシステムである エ ない

次の問8は必須問題です。必ず解答してください。

問8 ハフマン符号化を用いた文字列圧縮に関する次の記述を読んで、設問1～3に答えよ。

“A”～“D”の4種類の文字から成る文字列をハフマン符号化によって圧縮する。ハフマン符号化では、出現回数の多い文字には短いビット列を、出現回数の少ない文字には長いビット列を割り当てる。ハフマン符号化による文字列の圧縮手順は、次の(1)～(4)のとおりである。

- (1) 文字列中の文字の出現回数を求め、出現回数表を作成する。例えば、文字列“AAAABBCDCDDACCAAAAA”（以下、文字列 α という）中の文字の出現回数表は、表1のとおりになる。

表1 文字列 α 中の文字の出現回数表

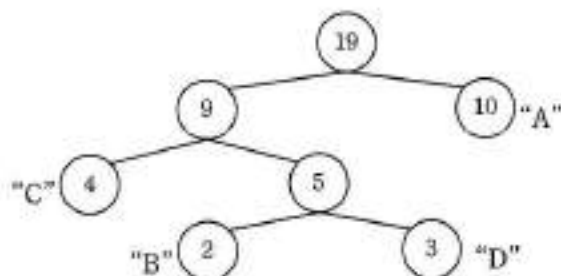
文字	A	B	C	D
出現回数	10	2	4	3

- (2) 文字の出現回数表に基づいてハフマン木を作成する。

ハフマン木の定義は、次のとおりである。

- ・ 節と枝で構成する二分木である。
- ・ 親である節は、子である節を常に二つもち、子の節の値の和を値としてもつ。
- ・ 子をもたない節（以下、葉という）は文字に対応し、出現回数を値としてもつ。
- ・ 親をもたない節（以下、根という）は、文字列の文字数を値としてもつ。

文字列 α に対応するハフマン木の例を、図1に示す。



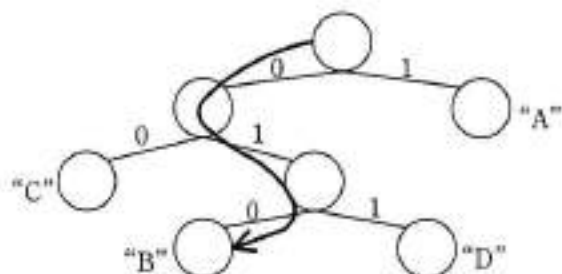
注記 丸の中の数値は各節がもつ値を表す。“A”～“D”は葉に対応する文字を表す。

図1 文字列 α に対応するハフマン木の例

ハフマン木は、次の手順で配列によって実現する。

- ① 節の値を格納する1次元配列を用意する。
 - ② 文字の出現回数表に基づいて、各文字に対応する葉の値を、配列の先頭の要素から順に格納する。
 - ③ 親が作成されていない節を二つ選択し、選択した順に左側の子、右側の子とする親の節を一つ作成する。この節の値を、配列中で値が格納されている最後の要素の次の要素に格納する。節の選択は節の値の小さい順に行い、同じ値をもつ節が二つ以上ある場合は、配列の先頭に近い要素に値が格納されている節を選択する。
 - ④ 親が作成されていない節が一つになるまで③を繰り返す。
- (3) ハフマン木から文字のビット列（以下、ビット表現という）を次の手順で作成する。
- ① 親と左側の子をつなぐ枝に0、右側の子をつなぐ枝に1の値をもつビットを割り当てる。
 - ② 文字ごとに根から対応する葉までたどったとき、枝のビット値を順に左から並べたものを各文字のビット表現とする。

図2に示すとおり、根から矢印のようにたどると、文字列 α の文字“B”のビット表現は010となる。



注記 線分は枝を表し、枝の上の数値は各枝のビット値を表す。

図2 文字列αにおける文字“B”のビット表現の作成例

- (4) 文字列の全ての文字を(3)で得られたビット表現に置き換えて、ビット列を作成する。

設問1 次の記述中の に入れる正しい答えを、解答群の中から選べ。

文字列“ABBBBBBBCCDD”を、ハフマン符号化を用いて表現する。各文字とビット表現を示した表は a である。ハフマン符号化によって圧縮すると、文字“A”～“D”をそれぞれ2ビットの固定長で表現したときの当該文字列の総ビット長に対する圧縮率は b となる。ここで、圧縮率は次式で計算した値の小数第3位を四捨五入して求める。

$$\text{圧縮率} = \frac{\text{ハフマン符号化によって圧縮したときの総ビット長}}{\text{2ビットの固定長で表現したときの総ビット長}}$$

aに関する解答群

ア

文字	A	B	C	D
ビット表現	010	1	00	011

イ

文字	A	B	C	D
ビット表現	010	0	01	111

ウ

文字	A	B	C	D
ビット表現	100	0	101	11

エ

文字	A	B	C	D
ビット表現	100	1	00	01

bに関する解答群

ア 0.77

イ 0.85

ウ 0.88

エ 0.92

設問2 ハフマン木を作成するプログラム 1 の説明及びプログラム 1 を読んで、プログラム 1 中の に入れる正しい答えを、解答群の中から選べ。

[プログラム 1 の説明]

- (1) 四つの 1 次元配列 `parent`, `left`, `right` 及び `freq` の同じ要素番号に対応する要素の組み（以下、要素組という）によって、一つの節を表す。要素番号は 0 から始まる。四つの配列の大きさはいずれも十分に大きく、全ての要素は -1 で初期化されている。
- (2) 図 3 に、図 1 に示したハフマン木を表現した場合の各配列の要素がもつ値を示す。配列 `parent` には親，配列 `left` には左側の子，配列 `right` には右側の子を表す要素組の要素番号がそれぞれ格納され，配列 `freq` には節の値が格納される。節が葉のとき，配列 `left` と配列 `right` の要素の値は，いずれも -1 である。図 3 では，要素番号 0 ~ 3 の要素組が，順に文字 "A" ~ "D" の葉に対応している。節が根のとき，配列 `parent` の要素の値は -1 である。

配列名	要素番号						
	0	1	2	3	4	5	6
<code>parent</code>	6	4	5	4	5	6	-1
<code>left</code>	-1	-1	-1	-1	1	2	5
<code>right</code>	-1	-1	-1	-1	3	4	0
<code>freq</code>	10	2	4	3	5	9	19

注記 矢印 $\bullet \rightarrow$ は，始点，終点の二つの要素組に対応する節が子と親の関係にあることを示す。

図 3 図 1 に示したハフマン木を表現する四つの配列

- (3) 副プログラム `Huffman` は，次の ① ~ ⑤ を受け取り，ハフマン木を表現する配列を作成する。
- ① 葉である節の個数 `size`
 - ② 初期化された配列 `parent`

- ③ 初期化された配列 left
 - ④ 初期化された配列 right
 - ⑤ 初期化された後、文字の出現回数が要素番号 0 から順に格納された配列 freq
- (4) 副プログラム SortNode は、親が作成されていない節を抽出し、節の値の昇順に整列し、節を表す要素組の要素番号を順に配列 node に格納し、その個数を変数 nsize に格納する。行番号 19~24 で親が作成されていない節を表す要素組の要素番号を抽出し、行番号 25 で節の値の昇順に整列する。
- (5) 副プログラム Sort (プログラムは省略) は、節を表す要素組の要素番号の配列 node を受け取り、要素番号に対応する要素組が表す節の値が昇順となるように整列する。節の値が同じときの順序は並べ替える直前の順序に従う。
- (6) 副プログラム Huffman, SortNode 及び Sort の引数の仕様を、表 2~4 に示す。

表 2 副プログラム Huffman の引数の仕様

引数	データ型	入出力	説明
size	整数型	入力/出力	節の個数
parent[]	整数型	入力/出力	節の親を表す要素組の要素番号を格納した配列
left[]	整数型	入力/出力	節の左側の子を表す要素組の要素番号を格納した配列
right[]	整数型	入力/出力	節の右側の子を表す要素組の要素番号を格納した配列
freq[]	整数型	入力/出力	節の値を格納した配列

表 3 副プログラム SortNode の引数の仕様

引数	データ型	入出力	説明
size	整数型	入力	節の個数
parent[]	整数型	入力	節の親を表す要素組の要素番号を格納した配列
freq[]	整数型	入力	節の値を格納した配列
nsize	整数型	出力	配列 node 中の、整列対象とした節の個数
node[]	整数型	出力	節の値の昇順に整列した、親が作成されていない節を表す要素組の要素番号を格納した配列

表 4 副プログラム Sort の引数の仕様

引数	データ型	入出力	説明
freq[]	整数型	入力	節の値を格納した配列
nsiz	整数型	入力	配列 node 中の、整列対象の節の個数
node[]	整数型	入力/出力	節を表す要素組の要素番号を格納した配列

[プログラム 1]

(行番号)

```

1  ○副プログラム: Huffman(整数型: size, 整数型: parent[], 整数型: left[],
    整数型: right[], 整数型: freq[])
2  ○整数型: i, j, nsiz
3  ○整数型: node[]
4  ・SortNode(size, parent, freq, nsiz, node)
5  ■ c
6  ・i ← node[0] /* 最も小さい値をもつ要素組の要素番号 */
7  ・j ← node[1] /* 2番目に小さい値をもつ要素組の要素番号 */
8  ・left[size] ← i
9  ・right[size] ← j
10 ・freq[size] ← freq[i] + freq[j] /* 子の値の合計 */
11 ・parent[i] ← size /* 子に親の節の要素番号を格納 */
12 ・parent[j] ← size /* 子に親の節の要素番号を格納 */
13 ・size ← size + 1
14 ・SortNode(size, parent, freq, nsiz, node)
15 ■

16 ○副プログラム: SortNode(整数型: size, 整数型: parent[], 整数型: freq[],
    整数型: nsiz, 整数型: node[])
17 ○整数型: i
18 ・nsiz ← 0
19 ■ i: 0, i < size, 1
20 d
21   ↑
21   ・node[nsiz] ← i
22   ↓
22   ・nsiz ← nsiz + 1
23   ↓
24 ■
25 ・Sort(freq, nsiz, node)

```

c, dに関する解答群

- ア $nsize \geq 0$ イ $nsize \geq 1$ ウ $nsize \geq 2$
エ $parent[i] < 0$ オ $parent[i] > 0$ カ $size \leq nsize$
キ $size \geq nsize$

設問3 ハフマン木から文字のビット表現を作成して表示するプログラム2の説明及びプログラム2を読んで、プログラム2中の に入れる正しい答えを、解答群の中から選べ。

[プログラム2の説明]

- (1) ビット表現を求めたい文字に対応する葉を表す要素組の要素番号を、副プログラム Encode の引数 k に与えて呼び出すと、ハフマン木から文字のビット表現を作成して表示する。
- (2) 副プログラム Encode の引数の仕様を、表5に示す。

表5 副プログラム Encode の引数の仕様

引数	データ型	入出力	説明
k	整数型	入力	節を表す要素組の要素番号
parent[]	整数型	入力	節の親を表す要素組の要素番号を格納した配列
left[]	整数型	入力	節の左側の子を表す要素組の要素番号を格納した配列

- (3) 副プログラム Encode は、行番号2の条件が成り立つとき、副プログラム Encode を再帰的に呼び出す。これによって、ハフマン木を葉から根までたどっていく。
- (4) 根にたどり着くと次は葉に向かってたどっていく。現在の節が親の左側の子のときは0を、右側の子のときは1を表示する。
- (5) 関数 print は、引数で与えられた文字列を表示する。

(プログラム 2)

(行番号)

```
1  ○副プログラム: Encode(整数型: k, 整数型: parent[], 整数型: left[])
2  ▲ e
3  · Encode(parent[k], parent, left)
4  ▲ f
5  · print("0") /* 0を表示する */
6  · print("1") /* 1を表示する */
7  ▼
8  ▼
```

eに関する解答群

- ア $k \geq 0$ イ $left[k] = -1$ ウ $left[k] \geq 0$
エ $parent[k] = -1$ オ $parent[k] \geq 0$

fに関する解答群

- ア $left[k] = k$ イ $left[parent[k]] = k$
ウ $parent[k] = k$ エ $parent[left[k]] = k$

次の問9から問13までの5問については、この中から1問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、2問以上マークした場合には、はじめの1問について採点します。

問9 次のCプログラムの説明及びプログラムを読んで、設問1, 2に答えよ。

入力ファイルを読み込んで、文字コードごとの出現回数を印字するプログラムである。

[プログラムの説明]

- (1) 入力ファイルは、バイナリファイルとして読み込む。入力ファイル中の各バイトの内容(ビット構成)に制約はない。入力ファイル名は、#defineで指定する。
- (2) 入力ファイル中の各バイトについて、文字コード(16進数00~FFで表示する)ごとの出現回数を求めて印字する。印字例を、図1に示す。
- (3) 印字様式を次に示す(記号①, ②, ③は、図1中の記号を指している)。
 - ・ 1行目に、処理したバイト数を①の形式で印字する。
 - ・ 3行目以降に、出現回数とその文字コードを②の形式で印字する。ただし、文字コードが20~7Eの場合は、文字コードの後にそれが表す文字(文字は、この冊子の末尾にあるアセンブラ言語の仕様の1.3で規定するもの)を③の形式で印字する。文字コードは、64行×4列の範囲に、上から下、左から右に文字コードの昇順となるように並べる。
- (4) プログラム中で使用している関数fgetc(s)は、ストリームsから1文字を読み込んで返す。ストリームが入力ファイルの終わりに達しているときはEOFを返す。
- (5) 入力ファイルのサイズは、long型(32ビットとする)で表現できる数値の範囲を超えないものとする。

①

----- 10000 bytes processed -----			
0 00	0 40 'e'	0 80	0 C8
0 01 ②	15 41 'A' ③	0 81	0 C1
0 02	0 42 'B'	0 82	0 C2
0 03	20 43 'C'	0 83	0 C3
⋮	⋮	⋮	⋮ (中略)
3620 20 ' '	0 60 '++'	0 A0	0 E0
8 21 '!'	103 61 'a'	0 A1	0 E1
76 22 '""'	10 62 'b'	0 A2	0 E2
⋮	⋮	⋮	⋮ (中略)
18 3E '>'	0 7E '++'	0 BE	0 FE
0 3F '?'	0 7F	0 BF	0 FF

図 1 印字例 (文字コード順)

〔プログラム〕

```

#include <stdio.h>

#define InName "sample.c" /* 入力ファイル名 */

int main() {

    FILE *infile;
    ①→ int chr, i;
        long cnt;
        long freq[256]; /* freq[i]: 文字コード i の出現回数 */

    for (chr = 0; chr <= 255; chr++)
        freq[chr] = 0;

    infile = fopen(InName, "rb");
    cnt = 0;
    while ((chr = fgetc(infile)) != EOF) {
        cnt++;
        freq[chr]++;
    }
    fclose(infile);
}

```



```

printf(" %10ld bytes processed\n\n", );
for (i = 0; i < 64; i++) {
    for (chr = i; chr <= ; chr +=  ) {
        if ((0x20 <= chr) && (chr <= 0x7E))
            printf(" %10ld %02X '%c'", freq[chr], chr, chr);
        else
            printf(" %10ld %02X   ", freq[chr], chr);
    }
    printf("\n");
}

```

②→

設問1 プログラム中の に入れる正しい答えを、解答群の中から選べ。

aに関する解答群

ア cnt - 1 イ cnt ウ cnt + 1

b, cに関する解答群

ア 4 イ 64 ウ 256
 エ i + 4 オ i + 64 カ i + 192

設問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。ここで、記述中の と には、設問1の正しい答えが入っているものとする。

文字コードを出現回数の降順に並べて印字する処理を追加する。追加した処理による印字例を、図2に示す。印字の様式は、文字コードの並び順を除いて、図1の3行目以降の様式と同じである。同じ出現回数の文字コードは、それらを文字コードの昇順に並べる。

3628 20 ' '	12 48 'K'	0 80	0 C0	
404 69 'i'	12 4C 'L'	0 81	0 C1	
329 8A	12 57 'W'	0 82	0 C2	
329 8D	11 4D 'M'	0 83	0 C3	
299 65 'e'	10 58 'X'	0 84	0 C4	
295 72 'r'	10 62 'b'	0 85	0 C5	
⋮	⋮	⋮	⋮	(中略)
14 36 '6'	0 7E '-'	0 8E	0 FE	
12 34 '4'	0 7F	0 8F	0 FF	

図 2 追加した処理による印字例 (出現回数順)

この処理のために、プログラムの行①の直後に、次の宣言を追加する。

```
int ih, ix, code[256];
```

さらに、プログラムの行②の位置に、次の整列処理部を追加する。

[整列処理部]

```

for (i = 0; i <= 255; i++)
    code[i] = i;
③→ ih = 255;
④→ while (ih > 0) {
⑤→     for (i = 0; i < ih; i++) {
⑥→         if (freq[i] < freq[i+1]) {
            Swap(code[i], code[i+1]);
            Swap(freq[i], freq[i+1]);
⑦→         }
        }
⑧→     ih--;
}
printf("\n");
for (i = 0; i < 64; i++) {
    for (chr = i; chr <= b; chr += c) {
        if ((0x20 <= code[chr]) && (code[chr] <= 0x7E))
            printf(" %10ld %02X '%c'", freq[chr], code[chr], code[chr]);
        else
            printf(" %10ld %02X   ", freq[chr], code[chr]);
    }
    printf("\n");
}

```

ここで、整列処理部で使用する $\text{Swap}(x, y)$ は、 x と y の内容を入れ替えるために用意したマクロである。

整列処理部では、整列対象のデータ数が比較的少なく、また同じ出現回数の文字コードは元の並び順が維持されるので、バブルソートを使用している。

行④の while 文のブロックを 1 回実行すると、配列の走査範囲（要素番号 $0 \sim ih$ ）中の出現回数の最小値とその文字コードが、要素番号 ih の位置に置かれ、配列の走査範囲が 1 だけ狭められる。これを繰り返して整列を行う。この処理では while 文のブロックの実行回数は常に d 回となる。

ここで、while 文のブロックの 1 回の繰返しにおいて、行⑥の if 文のブロック内の処理が最後に実行されたときの i の値を ix とすると、行⑤の for 文のブロックの実行が終了した時点で、配列 freq の要素番号 e 以降の要素の値は整列済みとなっている。これを利用して、整列処理部を表 1 に示すように変更すれば、while 文のブロックの実行回数を減らせる可能性がある。

表 1 整列処理部の変更内容

処置	変更内容
f の直後に追加	$ix = 0;$
行⑦の位置に追加	$ix = i;$
行⑧を置換え	$g;$

d に関する解答群

ア 253 イ 254 ウ 255 エ 256

e に関する解答群

ア $256 - ix$ イ $ix - 1$ ウ ix エ $ix + 1$

f に関する解答群

ア 行③ イ 行④ ウ 行⑤

g に関する解答群

ア $ih = ix - 1$ イ $ih = ix$ ウ $ih = ix + 1$ エ $ix = ih$

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 10 次の COBOL プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

〔プログラムの説明〕

地域の 5 か所に建物（以下、校舎という）をもち、中学生向けの進学指導を行っている P 学習塾では、定期的にテストを実施している。全ての塾生は、国語、数学、英語、理科及び社会の 5 教科のテストを受ける。テスト結果は、1 回分を学年別の結果ファイルに格納する。各校舎の各学年には 1 名以上の塾生が所属しており、1 学年の人数は塾全体で 999 人を超えることはない。

このプログラムは、結果ファイルに格納されたある学年のテスト結果を塾生ごとに集計して、5 教科の得点の合計（以下、合計点という）の降順に並べた順位リストを印字する。

(1) 結果ファイルは、図 1 に示すレコード様式の順ファイルであり、1 学年分のレコードが格納されている。

塾生番号 6 桁	校舎番号 2 桁	テスト結果				
		国語 3 桁	数学 3 桁	英語 3 桁	理科 3 桁	社会 3 桁

図 1 結果ファイルのレコード様式

- ① 塾生番号は、塾全体で一意的な番号である。
- ② 校舎番号には、塾生番号で示される塾生が所属する校舎の番号が 01～05 で格納されている。
- ③ テスト結果には、塾生番号で示される塾生の各教科の得点が 000～100 で格納されている。

(2) 順位リストは、図 2 に示す印字様式である。見出しは印刷済みとする。

全体順位	塾生番号	校舎番号	校舎内順位	合計点
ZZ9	999999	99	ZZ9	ZZ9
ZZ9	999999	99	ZZ9	ZZ9
⋮	⋮	⋮	⋮	⋮

図 2 順位リストの印字様式

- ① 全体順位には塾全体での順位を、校舎内順位には塾生が所属する校舎での順位を印字する。印字行は合計点の降順に並べる。
- ② 同じ合計点の塾生は同じ順位とする。例えば、塾全体で合計点の高い塾生から、492 点が 1 名、488 点が 2 名、487 点が 1 名いた場合は、図 3 のように順位付けして印字する。

全体順位	塾生番号	校舎番号	校舎内順位	合計点
1	182025	02	1	492
2	181007	01	1	488
2	182038	02	2	488
4	185011	05	1	487
⋮	⋮	⋮	⋮	⋮

図 3 順位リストの印字例

〔プログラム〕

(行番号)

```

01 DATA DIVISION,
02 FILE SECTION,
03 SD SRT-FILE,
04 01 SRT-REC,
05 02 SRT-NO          PIC 9(6),
06 02 SRT-BLD        PIC 9(2),
07 02 SRT-TOTAL      PIC 9(3),
08 FD RSL-FILE,
09 01 RSL-REC,
10 02 RSL-NO          PIC 9(6),
11 02 RSL-BLD        PIC 9(2),
12 02 RSL-SCR        PIC 9(3) OCCURS 5,
13 FD PRT-FILE,
14 01 PRT-REC,
15 02 PRT-ALL-RANK    PIC ZZ9,
16 02 PRT-SP1        PIC X(7),
17 02 PRT-NO          PIC 9(6),

```

```

18 02 PRT-SP2 PIC X(4).
19 02 PRT-BLD PIC 9(2).
20 02 PRT-SP3 PIC X(8).
21 02 PRT-BLD-RANK PIC ZZ9.
22 02 PRT-SP4 PIC X(9).
23 02 PRT-TOTAL PIC ZZ9.
24 WORKING-STORAGE SECTION.
25 77 SRT-FLAG PIC X(1) VALUE SPACE.
26 88 SRT-EOF VALUE "E".
27 77 RSL-FLAG PIC X(1) VALUE SPACE.
28 88 RSL-EOF VALUE "E".
29 77 ALL-ORD-NO PIC 9(3) VALUE ZERO.
30 77 ALL-RANK PIC 9(3) VALUE ZERO.
31 77 ALL-CR-TOTAL PIC 9(3) VALUE 999.
32 01 BLD-INF.
33 02 BLD-ORD-NO PIC 9(3) OCCURS 5 VALUE ZERO.
34 02 BLD-RANK PIC 9(3) OCCURS 5 VALUE ZERO.
35 02 BLD-CR-TOTAL PIC 9(3) OCCURS 5 VALUE 999.
36 PROCEDURE DIVISION.
37 MAIN-PROC.
38 OPEN INPUT RSL-FILE
39 OUTPUT PRT-FILE.
40 SORT SRT-FILE a
41 INPUT PROCEDURE IS REL-PROC
42 OUTPUT PROCEDURE IS RET-PROC.
43 CLOSE RSL-FILE PRT-FILE.
44 STOP RUN.
45 REL-PROC.
46 PERFORM UNTIL RSL-EOF
47 READ RSL-FILE AT END SET RSL-EOF TO TRUE
48 NOT AT END PERFORM CALC-PROC
49 END-READ
50 END-PERFORM.
51 CALC-PROC.
52 MOVE RSL-NO TO SRT-NO.
53 MOVE RSL-BLD TO SRT-BLD.
54 COMPUTE SRT-TOTAL = RSL-SCR(1) + RSL-SCR(2) +
55 RSL-SCR(3) + RSL-SCR(4) + RSL-SCR(5).
56 RELEASE SRT-REC.
57 RET-PROC.
58 PERFORM UNTIL SRT-EOF
59 RETURN SRT-FILE AT END SET SRT-EOF TO TRUE
60 NOT AT END PERFORM PRT-PROC

```

```

61      END-RETURN
62      END-PERFORM.
63      PRT-PROC.
64       b
65      IF SRT-TOTAL < ALL-CR-TOTAL THEN
66          MOVE SRT-TOTAL TO ALL-CR-TOTAL
67           c
68      END-IF.
69      IF SRT-TOTAL < BLD-CR-TOTAL(SRT-BLD) THEN
70          MOVE SRT-TOTAL TO BLD-CR-TOTAL(SRT-BLD)
71           d
72      END-IF.
73      INITIALIZE PRT-REC.
74      MOVE ALL-RANK TO PRT-ALL-RANK.
75      MOVE SRT-NO TO PRT-NO.
76      MOVE SRT-BLD TO PRT-BLD.
77      MOVE BLD-RANK(SRT-BLD) TO PRT-BLD-RANK.
78      MOVE SRT-TOTAL TO PRT-TOTAL.
79      WRITE PRT-REC.

```

設問1 プログラム中の に入れる正しい答えを、解答群の中から選べ。

aに関する解答群

- | | | | |
|---|-----------------------|---|--------------------------|
| ア | ASCENDING KEY SRT-NO | イ | ASCENDING KEY SRT-TOTAL |
| ウ | DESCENDING KEY SRT-NO | エ | DESCENDING KEY SRT-TOTAL |

bに関する解答群

- ア ADD 1 TO ALL-ORD-NO BLD-ORD-NO(SRT-BLD)
- イ ADD 1 TO ALL-RANK BLD-RANK(SRT-BLD)
- ウ INITIALIZE BLD-INF
- エ MOVE ALL-ORD-NO TO BLD-ORD-NO(SRT-BLD)

c, dに関する解答群

- ア MOVE ALL-CR-TOTAL TO ALL-RANK
- イ MOVE ALL-ORD-NO TO ALL-RANK
- ウ MOVE ALL-ORD-NO TO BLD-RANK(SRT-BLD)
- エ MOVE BLD-CR-TOTAL(SRT-BLD) TO BLD-RANK(SRT-BLD)
- オ MOVE BLD-ORD-NO(SRT-BLD) TO ALL-RANK
- カ MOVE BLD-ORD-NO(SRT-BLD) TO BLD-RANK(SRT-BLD)

設問2 図4の表示様式に示すとおり，“AVG:”に続けて整全体の平均点を表示し，“BLD-AVG:”に続けて校舎ごとの平均点を，校舎番号 01～05 の昇順に表示するようにプログラムを変更する。表1中の に入れる正しい答えを，解答群の中から選べ。

```
AVG:  Z29.9
BLD-AVG:  Z29.9  Z29.9  Z29.9  Z29.9  Z29.9
```

図4 表示様式

表1 プログラムの変更内容

処置	変更内容
行番号31と32の間に追加	<pre>77 ALL-TOTAL PIC 9(8) VALUE ZERO. 77 ALL-AVG PIC 9(3)V9(1). 77 DSP-AVG PIC Z29.9. 77 CNT PIC 9(2).</pre>
行番号35と36の間に追加	<pre>82 BLD-TOTAL PIC 9(6) OCCURS 5 VALUE ZERO. 82 BLD-AVG PIC 9(3)V9(1) OCCURS 5 VALUE ZERO.</pre>
行番号43と44の間に追加	<pre>COMPUTE ALL-AVG = ALL-TOTAL / ALL-ORD-NO. MOVE ALL-AVG TO DSP-AVG. DISPLAY "AVG: " DSP-AVG. DISPLAY "BLD-AVG: " WITH NO ADVANCING. PERFORM VARYING CNT FROM 1 BY 1 UNTIL CNT > 5 COMPUTE BLD-AVG(CNT) = <input type="text"/> MOVE BLD-AVG(CNT) TO DSP-AVG DISPLAY DSP-AVG " " WITH NO ADVANCING END-PERFORM.</pre>
<input type="text"/> c に追加	<pre>ADD SRT-TOTAL TO ALL-TOTAL BLD-TOTAL(SRT-BLD).</pre>

eに関する解答群

- ア 行番号 50 と 51 の間
- ウ 行番号 62 と 63 の間

- イ 行番号 51 と 52 の間
- エ 行番号 79 の後ろ

fに関する解答群

- ア BLD-TOTAL (CNT) / BLD-ORD-NO (CNT)
- イ BLD-TOTAL (CNT) / BLD-ORD-NO (SRT-BLD)
- ウ BLD-TOTAL (CNT) / BLD-RANK (CNT)
- エ BLD-TOTAL (CNT) / BLD-RANK (SRT-BLD)

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 11 次の Java プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

(Java プログラムで使用する API の説明は、この冊子の末尾を参照してください。)

(プログラムの説明)

升目を用いて表現された迷路と、迷路上に置かれて外部から操作される駒を表すプログラムである。

迷路は、駒が通れる升(以下、通路という)と通れない升(以下、壁という)から成る。迷路の外周は壁である。通路のうちの一つが開始地点であり、開始地点でない通路のうちの一つがゴール地点である。本問で扱う迷路を、図 1 に示す。

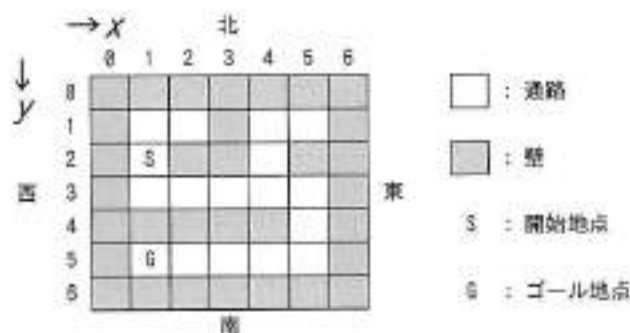


図 1 本問で扱う迷路

迷路上の升の位置は、2次元の座標 (x, y) で表す。 x と y はともに非負整数である。ある位置を基準として、 x の値が大きくなる方角を東、小さくなる方角を西、 y の値が大きくなる方角を南、小さくなる方角を北とする。

駒は、東西南北のいずれかを向いており、向いている方角を基準として、次の三つの操作を外部から受け付ける。

- ① 左の方向に向きを変える。
- ② 右の方向に向きを変える。
- ③ 隣接する前方の升が通路なら、1 升前進する。

- (1) クラス `Maze` は迷路を表す。コンストラクタの引数には、文字列で表現した迷路と、迷路の西端から東端までの升の個数を指定する。迷路を表現する文字列は、1 升を表す文字を西から東に向かって順に並べた 1 行分の文字列を、北から南に向かって順に連結したものである。升の種類は、`char` 型の値で表す。“*” は壁を、それ以外の値は通路を表し、“S” は開始地点を、“G” はゴール地点を表す。引数に誤りはないものとする。

メソッド `getStartLocation` は開始地点の座標を返す。メソッド `isGoal` は指定された座標の升がゴール地点であれば `true` を、それ以外は `false` を返す。メソッド `isBlank` は指定された座標の升が通路ならば `true` を、壁ならば `false` を返す。

- (2) クラス `Piece` は、迷路上に置かれる駒を表す。コンストラクタの引数で迷路を指定する。インスタンスは、最初は、開始地点に位置し、北を向いている。

メソッド `turnLeft` は左の方向に、`turnRight` は右の方向に向きを変える。

メソッド `tryStepForward` は、隣接する前方の升が通路なら 1 升前進し、前進した方角を履歴リストに追加してから `true` を返す。通路でなければ、前進せずに `false` を返す。

メソッド `isAtGoal` は、ゴール地点にいれば `true` を、それ以外は `false` を返す。

メソッド `getHistory` は、履歴リストを返す。

- (3) 列挙 `Direction` は、方角を表す。

メソッド `left` は列挙定数が表す方角に向かって左の方角を、`right` は右の方角を返す。

- (4) クラス `Location` は、迷路上の升の位置を示す座標を表す。

- (5) クラス `PlayMaze` は、図 1 に示す開始地点からゴール地点に至るまで駒を操作し、その後、履歴リストを表示する。

[プログラム 1]

```
public class Maze {
    private final String mazeData;
    private final int width;
    private final Location startLocation;

    public Maze(String mazeData, int width) {
        this.mazeData = mazeData;
    }
}
```

```

    this.width = width;
    startLocation = locationOf('S');
}

public Location getStartLocation() { return startLocation; }

public boolean isGoal(Location loc) {
    return mazeData.charAt(loc.y a width + loc.x) == 'G';
}

public boolean isBlank(Location loc) {
    return mazeData.charAt(loc.y a width + loc.x) != '*';
}

private Location locationOf(char c) {
    int index = mazeData.indexOf(c);
    return new Location(index b width, index / width);
}
}

```

[プログラム 2]

```

import java.util.ArrayList;
import java.util.List;

public class Piece {
    private final Maze maze;
    private Location location;
    private Direction direction = Direction.NORTH;
    private final List<Direction> history = new ArrayList<>();

    public Piece(Maze maze) {
        this.maze = maze;
        location = maze.getStartLocation();
    }

    public void turnLeft() { direction = direction.left(); }

    public void turnRight() { direction = direction.right(); }

    public boolean tryStepForward() {
        Location nextLocation = new Location(c);
        if (maze.isBlank(nextLocation)) {

```

```

        location = nextLocation;
        history.add(direction);
        return true;
    }
    return false;
}

public boolean isAtGoal() { return maze.isGoal(location); }

public List<Direction> getHistory() { return new ArrayList<>(history); }
}

```

[プログラム 3]

```

public enum Direction {
    NORTH(0, -1), EAST(1, 0), SOUTH(0, 1), WEST(-1, 0);

    public int dx, dy;

    private Direction(int dx, int dy) {
        this.dx = dx;
        this.dy = dy;
    }

    // クラスメソッド values は、この列挙で定義している列挙定数を、
    // 定義順に格納した配列を返す。
    // メソッド ordinal は、この列挙定数の定義順（先頭は 0）を返す。
    public Direction left() { return values()[ordinal() - 1]; }

    public Direction right() { return values()[ordinal() + 1]; }
}

```

[プログラム 4]

```

public class Location {
    public final int x, y;

    public Location(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

```

[プログラム5]

```

import java.util.List;

public class PlayMaze {
    public static void main(String... args) {
        Maze maze = new Maze("*****" +
            " *..*.*" +
            " *S**.**" +
            " *....*" +
            " *****" +
            " *G...*" +
            " *****", 7);

        Piece piece = new Piece(maze);
        while (!piece.isAtGoal()) {
            piece.turnLeft();
            while (!piece.tryStepForward()) {
                piece.turnRight();
            }
        }
        List<Direction> history = piece.getHistory(); ← α
        System.out.println(history);
    }
}

```

設問1 プログラム中の に入れる正しい答えを、解答群の中から選べ。

a. bに関する解答群

- | | | | |
|-----|-----|-----|-----|
| ア % | イ & | ウ * | エ + |
| オ - | カ / | キ ^ | ク |

cに関する解答群

- ア direction
- イ direction.dx, direction.dy
- ウ location + direction
- エ location.x + direction.dx, location.y + direction.dy

dに関する解答群

- ア $(\text{ordinal}() + 3) \% 4$ イ $(\text{ordinal}() + 3) / 4$
ウ $(\text{ordinal}() - 1) \% 4$ エ $(\text{ordinal}() - 1) / 4$

設問2 プログラム5のαの位置に次の処理を挿入し、実行結果として、図2に示す方角のリストを得た。駒は、開始地点からリストの方角の順に1升ずつ進むと、直前の升に戻る（正反対の方角に向きを変えて進む）ことなく、ゴール地点に至ることができる。□に入れる正しい答えを、解答群の中から選べ。

```
for (int i = □ e □; i < history.size(); i++) {  
    if (history.get(i - 1) == history.get(i).left().left()) {  
        history.remove(□ f □);  
        history.remove(□ f □);  
        i = i < 2 ? 0 : i - 2;  
    }  
}
```

[SOUTH, EAST, EAST, EAST, EAST, SOUTH, SOUTH, WEST, WEST, WEST, WEST]

図2 方角のリスト

eに関する解答群

- ア -1 イ 0 ウ 1

fに関する解答群

- ア i イ i + 1 ウ i - 1

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 12 次のアセンブラプログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

【プログラムの説明】

主記憶上の連続した語から成る領域（以下、ビット領域という）内において、指定されたビット列（以下、対象ビット列という）に対する操作を行うための副プログラム群である。

対象ビット列は、連続したビットの並びであり、開始位置（以下、ビット位置という）とビット数（以下、ビット長という）で表す。ビット位置は、ビット領域全体を連続したビット列と捉えて、ビット領域の最初の語の最上位ビットを 0 ビット目としたときの相対位置である。対象ビット列はビット領域内に収まるものとする。

図 1 に、ビット領域及び対象ビット列（網掛け部分）の例を示す。ビット位置が 44 でビット長が 24 の場合、ビット領域の最初の語を 0 語目としたとき、対象ビット列の最初のビットを含む語は、2 語目なので、これを相対アドレス 2 と表す。対象ビット列の最初のビットは、相対アドレス 2 の語の最上位ビットを 0 ビット目としたとき、12 ビット目なので、この位置をビットインデックス 12 と表す。対象ビット列の最後のビットの位置を、相対アドレス 4 の語のビットインデックス 3 と表す。

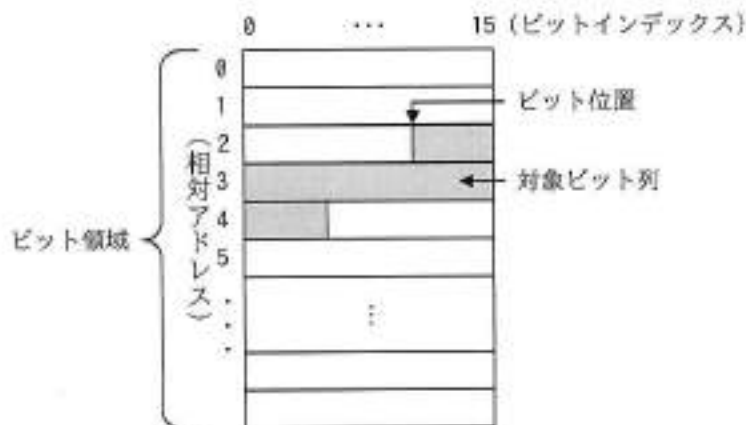


図 1 ビット領域及び対象ビット列の例

- (1) プログラム 1 は、対象ビット列の全ビットを 1 にする副プログラム BITSON である。主プログラムは、連続した 3 語に、ビット領域の先頭アドレス、対象ビット列のビット位置、及び対象ビット列のビット長をこの順に格納して、その 3 語の先頭アドレスを GR1 に設定して呼び出す。ビット位置及びビット長は、符号のない数値である。ビット長が 0 の場合は、対象ビット列に対する操作は行わない。
- (2) プログラム 2 は、GR0 で与えるビットマスクの値とビット領域内で操作の対象となる 1 語の値との論理和を求め、結果を同じ語に格納する副プログラム SETOP である。GR0 に 1 語内のビット操作の対象となるビットを 1 で表すビットマスクを、GR1 にビット領域の先頭アドレスを格納した語のアドレスを、GR2 に操作の対象となる語への相対アドレスを設定して、呼び出される。
- (3) 副プログラムから戻るとき、汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

[プログラム 1]

(行番号)

```

1 BITSON  START
2          RPUSH
3          LD   GR3,1,GR1      ; GR3: ビット位置
4 ; ビット位置を、相対アドレス (GR2) と
5 ; ビットインデックス (GR3) とに変換
6          LD   GR2,GR3
7          a
8          AND  GR3,#000F
9          LD   GR4,2,GR1      ; GR4: ビット長 (初期値)
10 ; (ビットインデックス + ビット長) ≥ 16 の間、ビット操作を行う
11 LOOP    LD   GR0,GR4
12         ADDL GR0,GR3
13         CPL  GR0,BITSPW
14         JMI  BREAK
15         LD   GR5,BITSPW     ; GR5: 語内で対象となるビット数
16         LD   GR3,GR3
17         JZE  FULL
18         b
19         SUBL GR3,GR3        ; GR3 ← 0
20 FULL    SUBL GR4,GR5        ; 操作するビット数だけビット長を減らす
21         LD   GR0,#1
22         SLL  GR0,0,GR5
23         SUBL GR0,#1        ; GR0: ビットマスク

```

```

24      CALL  SETOP
25      ADDL  GR2, =1          ; GR2: 次の語の相対アドレス
26      JUMP  LOOP
27 BREAK LD    GR4, GR4      ; 操作の対象になる未処理のビットあり?
28      
29 ; 未操作のビット数分の右詰めのビットマスクを生成し、
30 ; 必要なビット数だけ左へシフト
31      LD    GR0, =1
32      SLL  GR0, 0, GR4
33      SUBL GR0, =1
34      LD    GR5, BITSPW
35      SUBL  GR5, GR3
36      
37      SLL  GR0, 0, GR5      ; GR0: ビットマスク
38      CALL  SETOP
39 FIN   RPOP
40      RET
41 BITSPW DC  16             ; 1語のビット数
42      END

```

[プログラム 2]

(行番号)

```

1  SETOP  START
2      RPU
3      LD   GR3, 0, GR1
4      ADDL GR3, GR2
5      OR   GR0, 0, GR3
6      ST   GR0, 0, GR3
7      RPOP
8      RET
9      END

```

設問 1 プログラム中の に入れる正しい答えを、解答群の中から選べ。

a に関する解答群

ア SLL GR2, 0, GR3	イ SLL GR2, 4	ウ SLL GR2, 4, GR3
エ SRL GR2, 0, GR3	オ SRL GR2, 4	カ SRL GR2, 4, GR3

b, dに関する解答群

ア	ADDL	GR5, GR2	イ	ADDL	GR5, GR3	ウ	ADDL	GR5, GR4
エ	SUBL	GR5, GR2	オ	SUBL	GR5, GR3	カ	SUBL	GR5, GR4

cに関する解答群

ア	JNZ	FIN	イ	JNZ	LOOP	ウ	JPL	FIN
エ	JPL	LOOP	オ	JZE	FIN	カ	JZE	LOOP

設問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

対象ビット列の全ビットを0にする副プログラムを新規に作成することを考える。

ビット領域内の操作の対象になる語の相対アドレスやビットインデックスを求める処理などは、プログラム1で実装されている。そこで、新規の副プログラムを作成する前に、既存の副プログラム群を再編成し、対象ビット列の全ビットを1にする副プログラム及び0にする副プログラムが、プログラム1の処理を利用できるようにする。

プログラム1に変更を加え、それに伴って副プログラム BITSON を、その仕様は変更せずに再実装する方法を、次の(1)～(3)に示す。

- (1) 次に示すプログラム3を新たな副プログラム BITSON (以下、新 BITSON という) とする。新 BITSON は、既存の副プログラム SETOP のアドレスを GR7 に設定して変更後のプログラム1を呼び出す。

[プログラム3]

(行番号)

1	BITSON	START
2		RPUSH
3		LAD GR7, SETOP
4		CALL BITSOP
5		RPOP
6		RET
7		END

- (2) プログラム 1 の入口名を変更するために、行番号 1 のラベルを にする。
- (3) プログラム 1 の行番号 24 と行番号 38 の CALL 命令のオペランドを に変更する。

このように、既存の副プログラム群を再編成した後に、対象ビット列の全ビットを 0 にする処理のための副プログラム BITSOFF（プログラムは省略）と、それが変更後のプログラム 1 を介して呼び出す副プログラム RESETOP（プログラムは省略）を作成する。二つの副プログラムの概要を、次の (1)、(2) に示す。

- (1) 副プログラム BITSOFF は、新 BITSON と同様に、副プログラム RESETOP のアドレスを GR7 に設定して副プログラム BITSOP を呼び出す。
- (2) 副プログラム RESETOP は、副プログラム SETOP と同様のインタフェースで呼び出される。GR0 には、1 語内でビット操作の対象になるビットを表すビットマスクが設定されているので、GR0 の全ビットについて 0 と 1 を反転させた値と、GR1 と GR2 とで指定された語の値との論理積を求め、結果を同じ語に格納する。

e に関する解答群

- | | | |
|----------|----------|---------|
| ア BITSON | イ BITSOP | ウ SETOP |
|----------|----------|---------|

f に関する解答群

- | | | |
|---------------|---------------|---------------|
| ア 0, GR5 | イ BITSON, GR5 | ウ 0, GR6 |
| エ BITSON, GR6 | オ 0, GR7 | カ BITSON, GR7 |

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 13 次の表計算のワークシート及びマクロの説明を読んで、設問 1, 2 に答えよ。

〔表計算の説明〕

小売販売の Z 店では、会員として登録した顧客に会員カードを発行して販売促進活動に利用している。このたび、会員への 2018 年 10 月 1 日から 2019 年 3 月 31 日までの期間（以下、対象期間という）の販売データを基に、販売促進キャンペーンを実施することにした。

〔ワークシート：販売データ〕

ワークシート“販売データ”には、対象期間における会員への販売データが、販売日の昇順で格納されている。ワークシート“販売データ”の例を、図 1 に示す。

	A	B	C	D	E	F	G	H
1	レシート 番号	販売日	会員 番号	商品 コード	単価	数 量	販売 金額	会員 販売通番
2	81001001	2018-10-01	1115	BQR4500	48,000	1	48,000	1115001
3	81001002	2018-10-01	1299	CHN5102	2,400	3	7,200	1299001
4	81001002	2018-10-01	1299	CAQ5213	8,600	1	8,600	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3334	90331015	2019-03-31	1318	ABC1001	16,000	1	16,000	1318014
3335								

図 1 ワークシート“販売データ”の例

- (1) 列 A のレシート番号は、1 回の販売ごとに発行される一意の番号である。1 回の販売で複数の種類の商品販売したときは商品コードごとに行を替えて連続して格納し、同じレシート番号を付す。
- (2) 列 B の販売日は、商品を販売した日付である。日付は、1900 年 1 月 1 日からの経過日数として記録され、表計算ソフトの機能によって、yyyy-mm-dd（年-月-日）の形式で表示される。

- (3) 列 C の会員番号は、1001～9999 の一意の数値である。
- (4) 列 D の商品コードは販売した商品の商品コード、列 E の単価は販売単価、列 F の数量は販売数量、列 G は販売金額（単価×数量）である。
- (5) 列 H の会員販売通番は、上位 4 桁を会員番号、下位 3 桁を会員ごとの販売の通番とする 7 桁の数値である。会員ごとの販売の通番は、行 2 から当該行までに格納された当該行の会員番号をもつ会員に対する販売回数であり、1,000 未満である。ただし、同じレシート番号が続くときの二つ目以降の当該セルの値は 0 である。
- (6) 同じレシート番号をもつ販売データの販売日及び会員番号は、全て同じである。
- (7) 販売データは 9,998 件以下であり、販売データが入力されていない行の各セルには、空値が格納されている。

〔ワークシート：会員管理〕

ワークシート“会員管理”には、ワークシート“販売データ”のデータを基に、会員への販売状況や、販売促進キャンペーンのための情報を格納する。ワークシート“会員管理”の例を、図 2 に示す。

	A	B	C	D	E	F
1	件数	3,333	件	会員数	531	人
2						
3	会員番号	販売額合計	販売回数	最終販売日	会員クラス	クーポン
4	1001	198,500	7	2019-03-21	S	3,000
5	1002	44,100	6	2019-03-14	A	0
6	1003	0	0		C	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
534	1531	12,000	1	2019-03-29	C	0
535						

図 2 ワークシート“会員管理”の例

- (1) セル B1 の数値は、ワークシート“販売データ”に格納した販売データの行数である。セル E1 の数値は、会員数であり、ワークシート“会員管理”の行 4 以降に格納したデータの行数に等しい。
- (2) 列 A の行 4 以降には、全ての会員の会員番号を昇順に格納する。
- (3) 列 B の販売額合計は、ワークシート“販売データ”の中にある当該行の会員番

号をもつ会員に対する販売金額の合計である。

- (4) 列 C の販売回数は、ワークシート“販売データ”中にある、当該行の会員番号をもつ会員に対する販売回数である。
- (5) 列 D の最終販売日は、当該行の会員番号をもつ会員に対して最後に販売した日付である。対象期間に販売の記録がないときは、空値である。
- (6) 列 E の会員クラスは、当該行の会員番号をもつ会員を、下に示すクラス分類のルールに従って分類した結果である。
- (7) 列 F のクーポンは、キャンペーンで当該行の会員番号をもつ会員に発行するクーポンの総額である。クーポンは Z 店で 1 枚 1,000 円として使用できる券であり、販売額合計が 50,000 円以上 100,000 円未満で 1 枚、100,000 円以上 150,000 円未満で 2 枚、… というように販売額合計の 50,000 円ごとに 1 枚を発行する。

〔クラス分類のルール〕

- (1) 対象期間における各会員の販売額合計と販売回数を、表 1 に示すランク分類基準に従って、それぞれランク 1~3 に分類する。

表 1 ランク分類基準

ランク	販売額合計のランクの基準 (円)	販売回数のランクの基準 (回)
1	0~39,999	0~ 5
2	40,000~79,999	6~11
3	80,000 以上	12 以上

- (2) ワークシート“分類表”を参照して、会員を S, A, B, C のクラスに分類する。表 1 のランク分類基準に基づいて作成したワークシート“分類表”を、図 3 に示す。

	A	B	C	D	E	F
1	会員クラス			販売額合計		
2			ランク	1	2	3
3	ランク		下限	0	40,000	80,000
4	販売回数	1	0	C	B	A
5		2	6	C	A	S
6		3	12	B	A	S

図 3 ワークシート“分類表”

- ① セル D2～F2 には販売額合計のランクを示す値が、セル D3～F3 には販売額合計をセル D2～F2 のランクに分類するときの販売額合計の下限が格納されている。
- ② セル B4～B6 には販売回数のランクを示す値が、セル C4～C6 には販売回数をセル B4～B6 のランクに分類するときの販売回数の下限が格納されている。
- ③ セル D4～F6 には、当該セルの属する販売額合計と販売回数のランクから決まる会員クラスを示す“S”、“A”、“B”、“C”のいずれかが格納されている。

設問1 ワークシート“会員管理”に関する次の記述中の に入れる正しい答えを、解答群の中から選べ。

- (1) セル B4 に販売額合計を求める式を入力し、セル B5～B9002 に複写する。
- (2) セル C4 に販売回数を求める式を入力し、セル C5～C9002 に複写する。
- (3) セル D4 に最終販売日を求める次の式を入力し、セル D5～D9002 に複写する。

IF(論理和(A4=null, C4=0), null, a)

- (4) セル E4 に会員クラスを求める次の式を入力し、セル E5～E9002 に複写する。

IF(A4=null, null, 表引き(分類表!D\$4:F\$6, b))

- (5) セル F4 にクーポンの総額を求める次の式を入力し、セル F5～F9002 に複写する。

IF(A4=null, null, c)

aに関する解答群

- ア 照合検索(A4, 販売データ!C\$2:C\$9999, 販売データ!A\$2:A\$9999)
- イ 照合検索(A4, 販売データ!C\$2:C\$9999, 販売データ!B\$2:B\$9999)
- ウ 照合検索(A4 * 1000 + 1, 販売データ!H\$2:H\$9999, 販売データ!A\$2:A\$9999)
- エ 照合検索(A4 * 1000 + 1, 販売データ!H\$2:H\$9999, 販売データ!B\$2:B\$9999)
- オ 照合検索(A4 * 1000 + C4, 販売データ!H\$2:H\$9999, 販売データ!A\$2:A\$9999)
- カ 照合検索(A4 * 1000 + C4, 販売データ!H\$2:H\$9999, 販売データ!B\$2:B\$9999)

bに関する解答群

- ア 照合一致(B4, 分類表!C\$4:C\$6, 1), 照合一致(C4, 分類表!D\$3:F\$3, 1)
- イ 照合一致(B4, 分類表!D\$3:F\$3, 1), 照合一致(C4, 分類表!C\$4:C\$6, 1)
- ウ 照合一致(C4, 分類表!C\$4:C\$6, 1), 照合一致(B4, 分類表!D\$3:F\$3, 1)
- エ 照合一致(C4, 分類表!D\$3:F\$3, 1), 照合一致(B4, 分類表!C\$4:C\$6, 1)

cに関する解答群

- ア 切捨て(B4, -4) / 50000 * 1000
- イ 切捨て(B4 * 1000, -8) / 50000
- ウ 切捨て(B4 / 50000, 0) * 1000
- エ 切捨て(B4 / 50000 * 1000, 0)
- オ 切捨て(B4 / 50000 * 1000, -4)

設問2 新商品の見本の配布などの販売促進キャンペーンに活用するために、ワークシート“商品別管理”を作成し、マクロ manageMembers を格納した。マクロ manageMembers 中の に入れる正しい答えを、解答群の中から選べ。

[ワークシート：商品別管理]

セル B1 に商品の商品コードを入力して、マクロ manageMembers を実行すると、ワークシート“販売データ”及びワークシート“会員管理”に格納されているデータを基に、列 A の行 3 以降に当該商品を購入した会員の会員番号を、列 B の行 3 以降に会員番号ごとの当該商品の販売金額の合計（以下、商品販売額という）を、列 C の行 3 以降に会員番号ごとに最後に当該商品を販売した日付（以下、商品最終販売日という）を、それぞれ表示する。各行は同一会員に関する項目で構成し、商品販売額の降順に、商品販売額が等しいときは商品最終販売日の昇順に整列して表示する。ここで、マクロの実行開始時には、セル A3～C9001 に空値が格納されている。ワークシート“商品別管理”の例を、図4に示す。

	A	B	C
1	商品コード	ABC1001	
2	会員番号	商品販売額	商品最終販売日
3	1067	96,000	2019-02-14
4	1318	80,000	2019-03-31
⋮	⋮	⋮	⋮
225	1327	16,000	2019-03-30

図 4 ワークシート“商品別管理”の例

[マクロ：manageMembersの説明]

このマクロは、9,001行3列の要素をもつ2次元配列 sumTab を用意し、1以上会員数以下の全ての整数 i に対して、sumTab[$i, 0$] に会員番号を格納し、その会員番号に対応する商品販売額を sumTab[$i, 1$] に、商品最終販売日を sumTab[$i, 2$] に格納する。

- (1) 1以上会員数+1以下の全ての整数 i に対して、sumTab[$i, 1$]に初期値として0を格納する。
- (2) ワークシート“販売データ”の行2以降のデータを上から順に参照し、商品コードがセル B1 で指定した値と等しいとき、当該データの会員番号に対応した2次元配列 sumTab の商品販売額の要素に販売金額を加算するとともに、商品最終販売日の要素に日付を格納する処理を繰り返す。
- (3) 2次元配列 sumTab の会員番号を格納した行を、商品販売額の降順に、商品販売額が等しいときは商品最終販売日の昇順に整列する。
- (4) ワークシート“商品別管理”の行3以降のセルに、会員番号、商品販売額、商品最終販売日を、(3)の処理で整列された順に格納する。商品販売額が0の要素が現れたら、処理を終了する。

{マクロ: manageMembers}

○マクロ: manageMembers

○数値型: i, j, k, memNo, index, temp, sumTab[9001,3]

■ i: 1, i ≤ 会員管理!E1, 1

・ sumTab[i, 0] ← 相対(会員管理!A3, i, 0)

・ sumTab[i, 1] ← 0

■ sumTab[会員管理!E1 + 1, 1] ← 0

■ i: 1, i ≤ 会員管理!B1, 1

▲ 相対(販売データ!A1, i, 3) = B1

・ memNo ← 相対(販売データ!A1, i, 2)

・ index ← 照合一致(memNo, 会員管理!A4:A9002, 0)

・ d

・ sumTab[index, 2] ← 相対(販売データ!A1, i, 1)

■ i: 1, i ≤ 会員管理!E1, 1

■ j: 1, j ≤ 会員管理!E1 - i, 1

▲ e

■ k: 0, k ≤ 2, 1

・ temp ← sumTab[j, k]

・ sumTab[j, k] ← sumTab[j + 1, k]

・ sumTab[j + 1, k] ← temp

・ i ← 1

■ sumTab[i, 1] > 0

■ j: 0, j ≤ 2, 1

・ f

・ i ← i + 1

dに関する解答群

- ア $\text{sumTab}[1, \text{index}] \leftarrow \text{sumTab}[1, \text{index}] + \text{相対}(\text{販売データ}|A1, i, 6)$
- イ $\text{sumTab}[1, \text{index}] \leftarrow \text{sumTab}[i, \text{index}] + \text{相対}(\text{販売データ}|A1, \text{index}, 6)$
- ウ $\text{sumTab}[i, 1] \leftarrow \text{sumTab}[i, 1] + \text{相対}(\text{販売データ}|A1, i, 6)$
- エ $\text{sumTab}[i, 1] \leftarrow \text{sumTab}[i, 1] + \text{相対}(\text{販売データ}|A1, \text{index}, 6)$
- オ $\text{sumTab}[\text{index}, 1] \leftarrow \text{sumTab}[\text{index}, 1] + \text{相対}(\text{販売データ}|A1, i, 6)$
- カ $\text{sumTab}[\text{index}, 1] \leftarrow \text{sumTab}[\text{index}, 1] + \text{相対}(\text{販売データ}|A1, \text{index}, 6)$

eに関する解答群

- ア 論理積($\text{sumTab}[j, 1] < \text{sumTab}[j + 1, 1]$, 論理和($\text{sumTab}[j, 1] = \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] > \text{sumTab}[j + 1, 2]$))
- イ 論理積($\text{sumTab}[j, 1] > \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] > \text{sumTab}[j + 1, 2]$)
- ウ 論理積($\text{sumTab}[j, 1] > \text{sumTab}[j + 1, 1]$, 論理和($\text{sumTab}[j, 1] = \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] < \text{sumTab}[j + 1, 2]$))
- エ 論理和($\text{sumTab}[j, 1] < \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] > \text{sumTab}[j + 1, 2]$)
- オ 論理和($\text{sumTab}[j, 1] < \text{sumTab}[j + 1, 1]$, 論理積($\text{sumTab}[j, 1] = \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] > \text{sumTab}[j + 1, 2]$))
- カ 論理和($\text{sumTab}[j, 1] > \text{sumTab}[j + 1, 1]$, 論理積($\text{sumTab}[j, 1] = \text{sumTab}[j + 1, 1]$, $\text{sumTab}[j, 2] < \text{sumTab}[j + 1, 2]$))

fに関する解答群

- ア $\text{相対}(A2, i, j) \leftarrow \text{sumTab}[i, 1]$
- イ $\text{相対}(A2, i, j) \leftarrow \text{sumTab}[i, j]$
- ウ $\text{相対}(A2, i, j+1) \leftarrow \text{sumTab}[i, j]$
- エ $\text{相対}(A2, j, i) \leftarrow \text{sumTab}[j, 1]$
- オ $\text{相対}(A2, j, i) \leftarrow \text{sumTab}[j, i]$
- カ $\text{相対}(A2, j+1, i) \leftarrow \text{sumTab}[j, 1]$

■ Java プログラムで使用する API の説明

<pre>java.util public interface List<E> リスト（順序付けられたコレクション）のためのインタフェースを提供する。インタフェース Collection を継承する。</pre>
メソッド
<pre>public boolean add(E e) 指定された要素をリストの最後に追加する。 引数： e — リストに追加する要素 戻り値： true</pre>
<pre>public E get(int index) リスト内の指定された位置にある要素を返す。 引数： index — 返される要素のインデックス（先頭は 0） 戻り値： リスト内の指定された位置にある要素 例外： IndexOutOfBoundsException — インデックスがリストの範囲外するとき</pre>
<pre>public E remove(int index) リスト内の指定された位置にある要素を削除し、後続の要素があればそれらを順次詰める。 引数： index — 削除される要素のインデックス（先頭は 0） 戻り値： 削除された要素 例外： IndexOutOfBoundsException — インデックスがリストの範囲外するとき</pre>
<pre>public int size() リスト内の要素数を返す。 戻り値： リスト内の要素数</pre>

<pre>java.util public class ArrayList<E> インタフェース List の配列による実装である。 メソッドの説明は、インタフェース List の項を参照。</pre>
コンストラクタ
<pre>public ArrayList() 空のリストを作る。</pre>
<pre>public ArrayList(Collection<? extends E> c) 指定されたコレクションのイテレータが返す全ての要素を、イテレータが返す順に格納したリストを作る。 ArrayList のイテレータは、インデックスの昇順に、全ての要素を返す。 引数： c — このリストに要素が格納されるコレクション</pre>

java.lang

public final class String

クラス String は、文字列を表す。

メソッド

public char charAt(int index)

この文字列中の指定された位置にある char 型の値を返す。

引数: index — 返される char 型の値のインデックス (先頭は 0)

戻り値: この文字列中の指定された位置にある char 型の値

例外: `IndexOutOfBoundsException` — インデックスが負であるか又はこの文字列の長さ以上のとき

public int indexOf(int ch)

指定された文字が、この文字列中で最初に出現する位置のインデックスを返す。

引数: ch — 文字

戻り値: 指定された文字が最初に出現する位置のインデックス

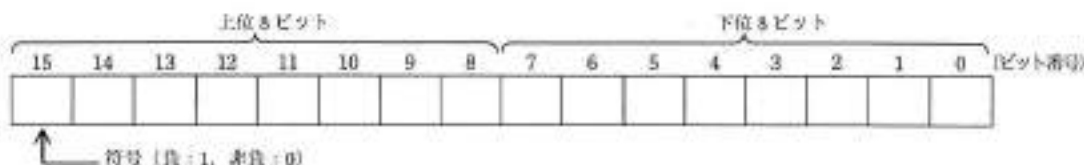
指定された文字がこの文字列中になければ -1

■アセンブラ言語の仕様

1. システム COMET II の仕様

1.1 ハードウェアの仕様

- (1) 1語は16ビットで、そのビット構成は、次のとおりである。



- (2) 主記憶の容量は65536語で、そのアドレスは0～65535番地である。
 (3) 数値は、16ビットの2進数で表現する。負数は、2の補数で表現する。
 (4) 制御方式は逐次制御で、命令語は1語長又は2語長である。
 (5) レジスタとして、GR (16ビット)、SP (16ビット)、PR (16ビット)、FR (3ビット)の4種類がある。
 GR (汎用レジスタ, General Register) は、GR0～GR7の8個があり、算術、論理、比較、シフトなどの演算に用いる。このうち、GR1～GR7のレジスタは、指標レジスタ (index register) としてアドレスの修飾にも用いる。
 SP (スタックポインタ, Stack Pointer) は、スタックの最上段のアドレスを保持している。
 PR (プログラムレジスタ, Program Register) は、次に実行すべき命令語の先頭アドレスを保持している。
 FR (フラグレジスタ, Flag Register) は、OF (Overflow Flag)、SF (Sign Flag)、ZF (Zero Flag) と呼ぶ3個のビットからなり、演算命令などの実行によって次の値が設定される。これらの値は、条件付き分岐命令で参照される。

OF	算術演算命令の場合は、演算結果が-32768～32767に収まらなくなったとき1になり、それ以外のとき0になる。論理演算命令の場合は、演算結果が0～65535に収まらなくなったとき1になり、それ以外のとき0になる。
SF	演算結果の符号が負 (ビット番号15が1) のとき1、それ以外のとき0になる。
ZF	演算結果が零 (全部のビットが0) のとき1、それ以外のとき0になる。

- (6) 論理加算又は論理減算は、被演算データを符号のない数値とみなして、加算又は減算する。

1.2 命令

命令の形式及びその機能を示す。ここで、一つの命令コードに対し2種類のオペランドがある場合、上段はレジスタ間の命令、下段はレジスタと主記憶間の命令を表す。

命 令	書 き 方		命 令 の 説 明	FRの設定
	命 令 コ ー ド	オ ペ ラ ン ド		

(1) ロード、ストア、ロードアドレス命令

ロード Load	LD	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r2)$ $r \leftarrow (\text{実効アドレス})$	○*1
ストア STore	ST	$r, \text{adr} [, x]$	実効アドレス $\leftarrow (r)$	
ロードアドレス Load Address	LAD	$r, \text{adr} [, x]$	$r \leftarrow \text{実効アドレス}$	—

(2) 算術、論理演算命令

算術加算 ADD Arithmetic	ADDA	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) + (r2)$ $r \leftarrow (r) + (\text{実効アドレス})$	○
論理加算 ADD Logical	ADDL	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) +_L (r2)$ $r \leftarrow (r) +_L (\text{実効アドレス})$	
算術減算 SUBtract Arithmetic	SUBA	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) - (r2)$ $r \leftarrow (r) - (\text{実効アドレス})$	
論理減算 SUBtract Logical	SUBL	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) -_L (r2)$ $r \leftarrow (r) -_L (\text{実効アドレス})$	
論理積 AND	AND	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) \text{ AND } (r2)$ $r \leftarrow (r) \text{ AND } (\text{実効アドレス})$	○*1
論理和 OR	OR	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) \text{ OR } (r2)$ $r \leftarrow (r) \text{ OR } (\text{実効アドレス})$	
排他的論理和 eXclusive OR	XOR	$r1, r2$ $r, \text{adr} [, x]$	$r1 \leftarrow (r1) \text{ XOR } (r2)$ $r \leftarrow (r) \text{ XOR } (\text{実効アドレス})$	

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	$r1, r2$ $r, \text{adr} [, x]$	<p>(r1) と (r2), 又は (r) と (実効アドレス) の算術比較又は論理比較を行い、比較結果によって、FR に次の値を設定する。</p> <table border="1"> <thead> <tr> <th rowspan="2">比較結果</th> <th colspan="2">FR の値</th> </tr> <tr> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>(r1) > (r2)</td> <td>0</td> <td>0</td> </tr> <tr> <td>(r) > (実効アドレス)</td> <td>0</td> <td>0</td> </tr> <tr> <td>(r1) = (r2)</td> <td>0</td> <td>1</td> </tr> <tr> <td>(r) = (実効アドレス)</td> <td>0</td> <td>1</td> </tr> <tr> <td>(r1) < (r2)</td> <td>1</td> <td>0</td> </tr> <tr> <td>(r) < (実効アドレス)</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	比較結果	FR の値		SF	ZF	(r1) > (r2)	0	0	(r) > (実効アドレス)	0	0	(r1) = (r2)	0	1	(r) = (実効アドレス)	0	1	(r1) < (r2)	1	0	(r) < (実効アドレス)	1	0	○*1
比較結果	FR の値																										
	SF	ZF																									
(r1) > (r2)	0	0																									
(r) > (実効アドレス)	0	0																									
(r1) = (r2)	0	1																									
(r) = (実効アドレス)	0	1																									
(r1) < (r2)	1	0																									
(r) < (実効アドレス)	1	0																									
論理比較 ComPare Logical	CPL	$r1, r2$ $r, \text{adr} [, x]$																									

(4) シフト演算命令

算術左シフト Shift Left Arithmetic	SLA	$r, \text{adr} [, x]$	<p>符号を除き (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。シフトの結果、空いたビット位置には、左シフトのときは 0、右シフトのときは符号と同じものが入る。</p>	○*2
算術右シフト Shift Right Arithmetic	SRA	$r, \text{adr} [, x]$		
論理左シフト Shift Left Logical	SLL	$r, \text{adr} [, x]$		
論理右シフト Shift Right Logical	SRL	$r, \text{adr} [, x]$		

(5) 分岐命令

正分岐 Jump on Plus	JPL	$\text{adr} [, x]$	<p>FR の値によって、実効アドレスに分岐する。分岐しないときは、次の命令に進む。</p> <table border="1"> <thead> <tr> <th rowspan="2">命令</th> <th colspan="3">分岐するときの FR の値</th> </tr> <tr> <th>OF</th> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>JPL</td> <td></td> <td>0</td> <td>0</td> </tr> <tr> <td>JNI</td> <td></td> <td>1</td> <td></td> </tr> <tr> <td>JNZ</td> <td></td> <td></td> <td>0</td> </tr> <tr> <td>JZE</td> <td></td> <td></td> <td>1</td> </tr> <tr> <td>JOV</td> <td>1</td> <td></td> <td></td> </tr> </tbody> </table>	命令	分岐するときの FR の値			OF	SF	ZF	JPL		0	0	JNI		1		JNZ			0	JZE			1	JOV	1			-
命令	分岐するときの FR の値																														
	OF	SF		ZF																											
JPL		0		0																											
JNI		1																													
JNZ				0																											
JZE				1																											
JOV	1																														
負分岐 Jump on Minus	JMI	$\text{adr} [, x]$																													
非零分岐 Jump on Non Zero	JNZ	$\text{adr} [, x]$																													
零分岐 Jump on Zero	JZE	$\text{adr} [, x]$																													
オーバーフロー分岐 Jump on Overflow	JOV	$\text{adr} [, x]$																													
無条件分岐 unconditional JUMP	JUMP	$\text{adr} [, x]$	無条件に実効アドレスに分岐する。																												

(6) スタック操作命令

プッシュ PUSH	PUSH	adr [, x]	SP ← (SP) - _L 1, (SP) ← 実効アドレス	—
ポップ POP	POP	r	r ← (SP), SP ← (SP) + _L 1	

(7) コール、リターン命令

コール CALL subroutine	CALL	adr [, x]	SP ← (SP) - _L 1, (SP) ← (PR), PR ← 実効アドレス	—
リターン Return from subroutine	RET		PR ← (SP), SP ← (SP) + _L 1	

(8) その他

スーパーバイザコール SuperVisor Call	SVC	adr [, x]	実効アドレスを引数として割出しを行う。実行後の GR と FR は不定となる。	—
ノーオペレーション No OPeration	NOP		何もしない。	

注記 r, r1, r2

adr

x

[]

()

実効アドレス

←

+_L, -_L

FR の設定

いずれも GR を示す。指定できる GR は GR0 ~ GR7

アドレスを示す。指定できる値の範囲は 0 ~ 65535

指標レジスタとして用いる GR を示す。指定できる GR は GR1 ~ GR7

[] 内の指定は省略できることを示す。

() 内のレジスタ又はアドレスに格納されている内容を示す。

adr と x の内容との論理加算値又はその値が示す番地

演算結果を、左辺のレジスタ又はアドレスに格納することを示す。

論理加算、論理減算を示す。

○ : 設定されることを示す。

○*1 : 設定されることを示す。ただし、OF には 0 が設定される。

○*2 : 設定されることを示す。ただし、OF にはレジスタから最後に送り出されたビットの値が設定される。

— : 実行前の値が保持されることを示す。

1.3 文字の符号表

(1) JIS X 0201 ラテン文字・片仮名用 8 ビット符号で規定する文字の符号表を使用する。

(2) 右に符号表の一部を示す。1 文字は 8 ビットからなり、上位 4 ビットを列で、下位 4 ビットを行で示す。例えば、間隔、4、H、¥ のビット構成は、16 進表示で、それぞれ 20、34、48、5C である。16 進表示で、ビット構成が 21 ~ 7E (及び表では省略している A1 ~ DF) に対応する文字を図形文字という。図形文字は、表示 (印刷) 装置で、文字として表示 (印字) できる。

(3) この表にない文字とそのビット構成が必要な場合は、問題中で与える。

列 \ 行	02	03	04	05	06	07
0	間隔	0	@	P	^	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[k	[
12	,	<	L	¥	l	
13	-	=	N]	n	}
14	.	>	O	^	o	-
15	/	?	O	_	o	

2. アセンブラ言語 CASL II の仕様

2.1 言語の仕様

- (1) CASL II は、COMET II のためのアセンブラ言語である。
- (2) プログラムは、命令行及び注釈行からなる。
- (3) 1 命令は 1 命令行で記述し、次の行へ継続できない。
- (4) 命令行及び注釈行は、次に示す記述の形式で、行の 1 文字目から記述する。

行の種類	記述の形式
命令行	オペランドあり [ラベル] [空白] (命令コード) [空白] (オペランド) [[空白] [コメント]]
	オペランドなし [ラベル] [空白] (命令コード) [[空白] [(;) [コメント]]]
注釈行	[空白] (;) [コメント]

注記 [] [] 内の指定が省略できることを示す。

() () 内の指定が必須であることを示す。

ラベル その命令の（先頭の語の）アドレスを他の命令やプログラムから参照するための名前である。長さは 1～8 文字で、先頭の文字は英大文字でなければならない。以降の文字は、英大文字又は数字のいずれでもよい。なお、予約語である GR0～GR7 は、使用できない。

空白 1 文字以上の間隔文字の列である。

命令コード 命令ごとに記述の形式が定義されている。

オペランド 命令ごとに記述の形式が定義されている。

コメント 覚え書きなどの任意の情報であり、処理系で許す任意の文字を書くことができる。

2.2 命令の種類

命令は、4 種類のアセンブラ命令 (START, END, DS, DC)、4 種類のマクロ命令 (IN, OUT, RPUSH, RPOP) 及び機械語命令 (COMET II の命令) からなる。その仕様を次に示す。

命令の種類	ラベル	命令コード	オペランド	機能
アセンブラ命令	ラベル	START	[実行開始番地]	プログラムの先頭を定義 プログラムの実行開始番地を定義 他のプログラムで参照する入口名を定義
		END		プログラムの終わりを明示
	[ラベル]	DS	語数	領域を確保
	[ラベル]	DC	定数 [, 定数] …	定数を定義
マクロ命令	[ラベル]	IN	入力領域, 入力文字長領域	入力装置から文字データを入力
	[ラベル]	OUT	出力領域, 出力文字長領域	出力装置へ文字データを出力
	[ラベル]	RPUSH		GR の内容をスタックに格納
	[ラベル]	RPOP		スタックの内容を GR に格納
機械語命令	[ラベル]		(「1.2 命令」を参照)	

2.3 アセンブラ命令

アセンブラ命令は、アセンブラの制御などを行う。

- (1)

START	[実行開始番地]
-------	----------

START 命令は、プログラムの先頭を定義する。

実行開始番地は、そのプログラム内で定義されたラベルで指定する。指定がある場合はその番地から、省略した場合は START 命令の次の命令から、実行を開始する。

また、この命令につけられたラベルは、他のプログラムから入口名として参照できる。

(2)

END	
-----	--

END 命令は、プログラムの終わりを定義する。

(3)

DS	語数
----	----

DS 命令は、指定した語数の領域を確保する。

語数は、10 進定数 (≥ 0) で指定する。語数を 0 とした場合、領域は確保しないが、ラベルは有効である。

(4)

DC	定数 [, 定数] ...
----	-----------------

DC 命令は、定数で指定したデータを (連続する) 語に格納する。

定数には、10 進定数、16 進定数、文字定数、アドレス定数の 4 種類がある。

定数の種類	書き方	命令の説明
10 進定数	n	n で指定した 10 進数値を、1 語の 2 進数データとして格納する。ただし、n が $-32768 \sim 32767$ の範囲にないときは、その下位 16 ビットを格納する。
16 進定数	h	h は 4 けたの 16 進数 (16 進数字は 0~9, A~F) とする。h で指定した 16 進数値を 1 語の 2 進数データとして格納する ($0000 \leq h \leq FFFF$)。
文字定数	'文字列'	文字列の文字数 (> 0) 分の連続する領域を確保し、最初の文字は第 1 語の下位 8 ビットに、2 番目の文字は第 2 語の下位 8 ビットに、...と順次文字データとして格納する。各語の上位 8 ビットには 0 のビットが入る。文字列には、間隔及び任意の図形文字を書くことができる。ただし、アポストロフィ (') は 2 個続けて書く。
アドレス定数	ラベル	ラベルに対応するアドレスを 1 語の 2 進数データとして格納する。

2.4 マクロ命令

マクロ命令は、あらかじめ定義された命令群とオペランドの情報によって、目的の機能を果たす命令群を生成する (語数は不定)。

(1)

IN	入力領域, 入力文字長領域
----	---------------

IN 命令は、あらかじめ割り当てた入力装置から、1 レコードの文字データを読み込む。

入力領域は、256 語長の作業域のラベルであり、この領域の先頭から、1 文字を 1 語に対応させて順次入力される。レコードの区切り符号 (キーボード入力の復帰符号など) は、格納しない。格納の形式は、DC 命令の文字定数と同じである。入力データが 256 文字に満たない場合、入力領域の残りの部分は実行前のデータを保持する。入力データが 256 文字を超える場合、以降の文字は無視される。

入力文字長領域は、1 語長の領域のラベルであり、入力された文字の長さ (≥ 0) が 2 進数で格納される。ファイルの終わり (end of file) を検出した場合は、-1 が格納される。

IN 命令を実行すると、GR の内容は保存されるが、FR の内容は不定となる。

(2)

OUT	出力領域, 出力文字長領域
-----	---------------

OUT 命令は、あらかじめ割り当てた出力装置に、文字データを、1 レコードとして書き出す。

出力領域は、出力しようとするデータが 1 文字 1 語で格納されている領域のラベルである。格納の形式は、DC 命令の文字定数と同じであるが、上位 8 ビットは、OS が無視するので 0 でなくてもよい。

出力文字長領域は、1 語長の領域のラベルであり、出力しようとする文字の長さ (≥ 0) を 2 進数で格納しておく。

OUT 命令を実行すると、GR の内容は保存されるが、FR の内容は不定となる。

(3)

R PUSH	
--------	--

R PUSH 命令は、GR の内容を、GR1, GR2, …, GR7 の順序でスタックに格納する。

(4)

R POP	
-------	--

R POP 命令は、スタックの内容を順次取り出し、GR7, GR6, …, GR1 の順序で GR に格納する。

2.5 機械語命令

機械語命令のオペランドは、次の形式で記述する。

r, r1, r2 GR は、記号 GR0 ~ GR7 で指定する。

x 指標レジスタとして用いる GR は、記号 GR1 ~ GR7 で指定する。

adr アドレスは、10 進定数、16 進定数、アドレス定数又はリテラルで指定する。
リテラルは、一つの 10 進定数、16 進定数又は文字定数の前に等号 (=) を付けて記述する。CASL II は、等号の後の定数をオペランドとする DC 命令を生成し、そのアドレスを adr の値とする。

2.6 その他

- (1) アセンブラによって生成される命令語や領域の相対位置は、アセンブラ言語での記述順序とする。ただし、リテラルから生成される DC 命令は、END 命令の直前にまとめて配置される。
- (2) 生成された命令語、領域は、主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して、次の取決めがある。

- (1) アセンブラは、未定義ラベル（オペランド欄に記述されたラベルのうち、そのプログラム内で定義されていないラベル）を、他のプログラムの入口名（START 命令のラベル）と解釈する。この場合、アセンブラはアドレスの決定を保留し、その決定を OS に任せる。OS は、実行に先立って他のプログラムの入口名との関係処理を行いアドレスを決定する（プログラムの関係）。
- (2) プログラムは、OS によって起動される。プログラムがロードされる主記憶の領域は不定とするが、プログラム中のラベルに対応するアドレス値は、OS によって実アドレスに補正されるものとする。
- (3) プログラムの起動時に、OS はプログラム用に十分な容量のスタック領域を確保し、その最後のアドレスに 1 を加算した値を SP に設定する。
- (4) OS は、CALL 命令でプログラムに制御を渡す。プログラムを終了し OS に制御を戻すときは、RET 命令を使用する。
- (5) IN 命令に対応する入力装置、OUT 命令に対応する出力装置の割当ては、プログラムの実行に先立って利用者が行う。
- (6) OS は、入出力装置や媒体による入出力手続の違いを吸収し、システムでの標準の形式及び手続（異常処理を含む）で入出力を行う。したがって、IN, OUT 命令では、入出力装置の違いを意識する必要はない。

3.2 未定義事項

プログラムの実行等に関し、この仕様で定義しない事項は、処理系によるものとする。

表計算ソフトの機能・用語（基本情報技術者試験用）

表計算ソフトの機能、用語などは、原則として次による。

なお、ワークシートの保存、読出し、印刷、罫線作成やグラフ作成など、ここで示す以外の機能などを使用するときには、問題文中に示す。

1. ワークシート

- (1) 列と行とで構成される昇目の作業領域をワークシートという。ワークシートの大きさは 256 列、10,000 行とする。
- (2) ワークシートの列と行のそれぞれの位置は、列番号と行番号で表す。列番号は、最左端列の列番号を A とし、A, B, …, Z, AA, AB, …, AZ, BA, BB, …, BZ, …, IU, IV と表す。行番号は、最上端行の行番号を 1 とし、1, 2, …, 10000 と表す。
- (3) 複数のワークシートを利用することができる。このとき、各ワークシートには一意のワークシート名を付けて、他のワークシートと区別する。

2. セルとセル範囲

- (1) ワークシートを構成する各升をセルという。その位置は列番号と行番号で表し、それをセル番地という。
【例】列 A 行 1 にあるセルのセル番地は、A1 と表す。
- (2) ワークシート内のある長方形の領域に含まれる全てのセルの集まりを扱う場合、長方形の左上端と右下端のセル番地及び“:”を用いて、“左上端のセル番地:右下端のセル番地”と表す。これを、セル範囲という。
【例】左上端のセル番地が A1 で、右下端のセル番地が B3 のセル範囲は、A1:B3 と表す。
- (3) 他のワークシートのセル番地又はセル範囲を指定する場合には、ワークシート名と“!”を用い、それぞれ“ワークシート名!セル番地”又は“ワークシート名!セル範囲”と表す。
【例】ワークシート“シート1”のセル B5～G10 を、別のワークシートから指定する場合には、シート1!B5:G10 と表す。

3. 値と式

- (1) セルは値をもち、その値はセル番地によって参照できる。値には、数値、文字列、論理値及び空値がある。
- (2) 文字列は一重引用符“'’”で囲って表す。
【例】文字列“A”, “BC”は、それぞれ'A', 'BC' と表す。
- (3) 論理値の真を true, 偽を false と表す。
- (4) 空値を null と表し、空値をもつセルを空白セルという。セルの初期状態は、空白セルとする。

- (5) セルには、式を入力することができる。セルは、式を評価した結果の値をもつ。
- (6) 式は、定数、セル番地、演算子、括弧及び関数から構成される。定数は、数値、文字列、論理値又は空値を表す表記とする。式中のセル番地は、その番地のセルの値を参照する。
- (7) 式には、算術式、文字式及び論理式がある。評価の結果が数値となる式を算術式、文字列となる式を文字式、論理値となる式を論理式という。
- (8) セルに式を入力すると、式は直ちに評価される。式が参照するセルの値が変化したときには、直ちに、適切に再評価される。

4. 演算子

- (1) 単項演算子は、正符号“+”及び負符号“-”とする。
- (2) 算術演算子は、加算“+”、減算“-”、乗算“*”、除算“/”及びべき乗“^”とする。
- (3) 比較演算子は、より大きい“>”、より小さい“<”、以上“≥”、以下“≤”、等しい“=”及び等しくない“≠”とする。
- (4) 括弧は丸括弧“()”を使う。
- (5) 式中に複数の演算及び括弧があるときの計算の順序は、次表の優先順位に従う。

演算の種類	演算子	優先順位
括弧	()	高 ↑ ↓ 低
べき乗演算	^	
単項演算	+, -	
乗除演算	*, /	
加減演算	+, -	
比較演算	>, <, ≥, ≤, =, ≠	

5. セルの複写

- (1) セルの値又は式を、他のセルに複写することができる。
- (2) セルを複写する場合で、複写元のセル中にセル番地を含む式が入力されているとき、複写元と複写先のセル番地の差を維持するように、式中のセル番地を変化させるセルの参照方法を相対参照という。この場合、複写先のセルとの列番号の差及び行番号の差を、複写元のセルに入力された式中の各セル番地に加算した式が、複写先のセルに入る。

【例】セル A6 に式 $A1 + 5$ が入力されているとき、このセルをセル B8 に複写すると、セル B8 には式 $B3 + 5$ が入る。

- (3) セルを複写する場合で、複写元のセル中にセル番地を含む式が入力されているとき、そのセル番地の列番号と行番号の両方又は片方を変化させないセルの参照方法を絶対参照という。絶対参照を適用する列番号と行番号の両方又は片方の直前には“\$”を付ける。

【例】セル B1 に式 $\$A\$1 + \$A2 + A\5 が入力されているとき、このセルをセル C4 に複写す

ると、セル C4 には式 $\$A\$1 + \$A\$5 + B\$5$ が入る。

- (4) セルを複写する場合で、複写元のセル中に、他のワークシートを参照する式が入力されているとき、その参照するワークシートのワークシート名は複写先でも変わらない。

[例] ワークシート“シート2”のセル A6 に式 シート1!A1 が入力されているとき、このセルをワークシート“シート3”のセル B8 に複写すると、セル B8 には式 シート1!B3 が入る。

6. 関数

式には次の表で定義する関数を利用することができる。

書式	解 説
合計 (セル範囲 ¹⁾)	セル範囲に含まれる数値の合計を返す。 [例] 合計 (A1:B5) は、セル A1 ~ B5 に含まれる数値の合計を返す。
平均 (セル範囲 ¹⁾)	セル範囲に含まれる数値の平均を返す。
標本標準偏差 (セル範囲 ¹⁾)	セル範囲に含まれる数値を標本として計算した標準偏差を返す。
母標準偏差 (セル範囲 ¹⁾)	セル範囲に含まれる数値を母集団として計算した標準偏差を返す。
最大 (セル範囲 ¹⁾)	セル範囲に含まれる数値の最大値を返す。
最小 (セル範囲 ¹⁾)	セル範囲に含まれる数値の最小値を返す。
IF (論理式, 式1, 式2)	論理式の値が true のとき式 1 の値を、false のとき式 2 の値を返す。 [例] IF (B3 > A4, “北海道”, C4) は、セル B3 の値がセル A4 の値より大きいとき文字列 “北海道” を、それ以外の場合セル C4 の値を返す。
個数 (セル範囲)	セル範囲に含まれるセルのうち、空白セルでないセルの個数を返す。
条件付個数 (セル範囲, 検索条件の記述)	セル範囲に含まれるセルのうち、検索条件の記述で指定された条件を満たすセルの個数を返す。検索条件の記述は比較演算子と式の組で記述し、セル範囲に含まれる各セルと式の値を、指定した比較演算子によって評価する。 [例1] 条件付個数 (H5:L9, > A1) は、セル H5 ~ L9 のセルのうち、セル A1 の値より大きな値をもつセルの個数を返す。 [例2] 条件付個数 (H5:L9, = “A4”) は、セル H5 ~ L9 のセルのうち、文字列 “A4” をもつセルの個数を返す。
整数部 (算術式)	算術式の値以下で最大の整数を返す。 [例1] 整数部 (3.9) は、3 を返す。 [例2] 整数部 (-3.9) は、-4 を返す。
剰余 (算術式1, 算術式2)	算術式1 の値を被除数、算術式2 の値を除数として除算を行ったときの剰余を返す。関数 “剰余” と “整数部” は、剰余 (x,y) = x - y * 整数部 (x / y) という関係を満たす。 [例1] 剰余 (10,3) は、1 を返す。 [例2] 剰余 (-10,3) は、2 を返す。
平方根 (算術式)	算術式の値の非負の平方根を返す。算術式の値は、非負の数値でなければならない。
論理積 (論理式1, 論理式2, …) ²⁾	論理式1, 論理式2, … の値が全て true のとき、true を返す。それ以外の場合 false を返す。
論理和 (論理式1, 論理式2, …) ²⁾	論理式1, 論理式2, … の値のうち、少なくとも一つが true のとき、true を返す。それ以外の場合 false を返す。
否定 (論理式)	論理式の値が true のとき false を、false のとき true を返す。

切上げ (算術式, 桁位置)	算術式の値を指定した桁位置で、関数“切上げ”は切り上げた値を、関数“四捨五入”は四捨五入した値を、関数“切捨て”は切り捨てた値を返す。ここで、桁位置は小数第1位の桁を0とし、右方向を正として数えたときの位置とする。 [例1] 切上げ(-314.059,2) は、-314.06 を返す。
四捨五入 (算術式, 桁位置)	[例2] 切上げ(314.059,-2) は、400 を返す。
切捨て (算術式, 桁位置)	[例3] 切上げ(314.059,0) は、315 を返す。
結合(式1,式2,...) ²⁾	式1, 式2, ... のそれぞれの値を文字列として扱い、それらを引数の順につないでできる一つの文字列を返す。 [例] 結合("北海道","九州",123,456) は、文字列“北海道九州123456”を返す。
順位 (算術式, セル範囲 ¹⁾ , 順序の指定)	セル範囲の中での算術式の値の順位を、順序の指定が0の場合は昇順で、1の場合は降順で数えて、その順位を返す。ここで、セル範囲の中に同じ値がある場合、それらを同順位とし、次の順位は同順の個数だけ加算した順位とする。
乱数()	0以上1未満の一律乱数(実数値)を返す。
表引き(セル範囲, 行の位置, 列の位置)	セル範囲の左上端から行と列をそれぞれ1, 2, ... と数え、セル範囲に含まれる行の位置と列の位置で指定した場所にあるセルの値を返す。 [例] 表引き(A3:H11,2,5) は、セル E4 の値を返す。
垂直照合(式, セル範囲, 列の位置, 検索の指定)	セル範囲の左端列を上から下に走査し、検索の指定によって指定される条件を満たすセルが現れる最初の行を探し、その行に対して、セル範囲の左端列から列を1, 2, ... と数え、セル範囲に含まれる列の位置で指定した列にあるセルの値を返す。 ・検索の指定が0の場合の条件: 式の値と一致する値を検索する。 ・検索の指定が1の場合の条件: 式の値以下の最大値を検索する。このとき、左端列は上から順に昇順に整列されている必要がある。 [例] 垂直照合(15,A2:E10,5,0) は、セル範囲の左端列をセル A2, A3, ..., A10 と探す。このとき、セル A6 で15を最初に見つけたとすると、左端列 A から数えて5列目の列 E 中で、セル A6 と同じ行にあるセル E6 の値を返す。
水平照合(式, セル範囲, 行の位置, 検索の指定)	セル範囲の上端行を左から右に走査し、検索の指定によって指定される条件を満たすセルが現れる最初の列を探し、その列に対して、セル範囲の上端行から行を1, 2, ... と数え、セル範囲に含まれる行の位置で指定した行にあるセルの値を返す。 ・検索の指定が0の場合の条件: 式の値と一致する値を検索する。 ・検索の指定が1の場合の条件: 式の値以下の最大値を検索する。このとき、上端行は左から順に昇順に整列されている必要がある。 [例] 水平照合(15,A2:G6,5,1) は、セル範囲の上端行をセル A2, B2, ..., G2 と探す。このとき、15以下の最大値をセル D2 で最初に見つけたとすると、上端行 2 から数えて5行目の行 6 中で、セル D2 と同じ列にあるセル D6 の値を返す。
照合検索(式, 検索のセル範囲, 抽出のセル範囲)	1行又は1列を対象とする同じ大きさの検索のセル範囲と抽出のセル範囲に対して、検索のセル範囲を左端又は上端から走査し、式の値と一致する最初のセルを探し、見つかったセルの検索のセル範囲の中での位置と、抽出のセル範囲の中での位置が同じセルの値を返す。 [例] 照合検索(15,A1:A8,C6:C13) は、セル A1~A8 をセル A1, A2, ... と探す。このとき、セル A5 で15を最初に見つけたとすると、セル C6~C13の上端から数えて5番目のセル C10 の値を返す。

照合一致(式,セル範囲,検索の指定)	<p>1行又は1列を対象とするセル範囲に対して、セル範囲の左端又は上端から走査し、検索の指定によって指定される条件を満たす最初のセルを探す。見つかったセルの位置を、セル範囲の左端又は上端から1, 2, …と数えた値とし、その値を返す。</p> <ul style="list-style-type: none"> ・検索の指定が0の場合の条件: 式の値と一致する値を検索する。 ・検索の指定が1の場合の条件: 式の値以下の最大値を検索する。このとき、セル範囲は左端又は上端から順に昇順に整列されている必要がある。 ・検索の指定が-1の場合の条件: 式の値以上の最小値を検索する。このとき、セル範囲は左端又は上端から順に降順に整列されている必要がある。 <p>[例] 照合一致(15, B2:B12, -1) は、セル B2 ~ B12 をセル B2, B3, …と探す。このとき、15以上の最小値をセルB9で最初に見つけたとすると、セル B2 から数えた値 8 を返す。</p>
条件付合計(検索のセル範囲, 検索条件の記述, 合計のセル範囲 ¹⁾)	<p>行数及び列数が共に同じ検索のセル範囲と合計のセル範囲に対して、検索と合計を行う。検索のセル範囲に含まれるセルのうち、検索条件の記述で指定される条件を満たすセルを全て探す。検索条件の記述を満たした各セルについての左上端からの位置と、合計のセル範囲中で同じ位置にある各セルの値を合計して返す。</p> <p>検索条件の記述は比較演算子と式の組で記述し、検索のセル範囲に含まれる各セルと式の値を、指定した比較演算子によって評価する。</p> <p>[例1] 条件付合計(A1:B8, > E1, C2:D9) は、検索のセル範囲であるセル A1 ~ B8 のうち、セル E1 の値より大きな値をもつ全てのセルを探す。このとき、セル A2, B4, B7 が見つかったとすると、合計のセル範囲であるセル C2 ~ D9 の左上端からの位置が同じであるセル C3, D5, D8 の値を合計して返す。</p> <p>[例2] 条件付合計(A1:B8, = 160, C2:D9) は、検索のセル範囲であるセル A1 ~ B8 のうち、160と一致する値をもつ全てのセルを探す。このとき、セル A2, B4, B7 が見つかったとすると、合計のセル範囲であるセル C2 ~ D9 の左上端からの位置が同じであるセル C3, D5, D8 の値を合計して返す。</p>

注¹⁾ 引数として渡したセル範囲の中で、数値以外の値は処理の対象としない。

²⁾ 引数として渡すことができる式の個数は、1以上である。

7. マクロ

(1) ワークシートとマクロ

ワークシートには複数のマクロを格納することができる。

マクロは一意のマクロ名を付けて宣言する。マクロの実行は、表計算ソフトのマクロの実行機能を使って行う。

[例] ○マクロ: Pro

例は、マクロ Pro の宣言である。

(2) 変数とセル変数

変数の型には、数値型、文字列型及び論理型があり、変数は宣言することで使用できる。変数名にセル番地を使用することはできない。

[例] ○数値型: row, col

例は、数値型の変数 row, col の宣言である。

セルを変数として使用でき、これをセル変数という。セル変数は、宣言せずに使用できる。

セル変数の表現方法には、絶対表現と相対表現とがある。

セル変数の絶対表現は、セル番地で表す。

セル変数の相対表現は、次の書式で表す。

書式	解 説
相対(セル変数, 行の位置, 列の位置)	セル変数で指定したセルを基準のセルとする。そのセルの行番号と列番号の位置を 0 とし、下又は右方向を正として数え、行の位置と列の位置で指定した数と一致する場所にあるセルを表す変数である。

[例1] 相対(B5, 2, 3) は、セル E7 を表す変数である。

[例2] 相対(B5, -2, -1) は、セル A3 を表す変数である。

(3) 配列

数値型、文字列型又は論理型の配列は宣言することで使用できる。添字を “[” 及び “] ” で囲み、添字が複数ある場合はコンマで区切る。添字は 0 から始まる。

なお、数値型及び文字列型の変数及び配列の要素には、空値を格納することができる。

[例] ○文字列型: table[100, 200]

例は、100 × 200 個の文字列型の要素をもつ 2 次元配列 table の宣言である。

(4) 宣言、注釈及び処理

宣言、注釈及び処理の記述は、“共通に使用される擬似言語の記述形式”の〔宣言、注釈及び処理〕に従う。

処理の記述中に式又は関数を使用する場合、その記述中に変数、セル変数又は配列の要素が使用できる。

[例] ○数値型: row

```
■ row: 0, row < 5, 1
  |
  |   - 相対(E1, row, 1) ← 垂直照合(相対(E1, row, 0), A1:B10, 2, 0) * 10
  ■
```

例は、セル E1, E2, …, E5 の各値に対して、セル A1 ~ A10 の中で同じ値をもつセルが現れる最初の行を探し、見つけた行の列 B のセルの値を 10 倍し、セル F1, F2, …, F5 の順に代入する。

〔メモ用紙〕

6. 退室可能時間中に退室する場合は、手を挙げて監督員に合図し、答案用紙が回収されてから静かに退室してください。

退室可能時間	13:40 ~ 15:20
--------	---------------

7. 問題に関する質問にはお答えできません。文意どおり解釈してください。
8. 問題冊子の余白などは、適宜利用して構いません。ただし、問題冊子を切り離して利用することはできません。
9. Java プログラムで使用する API の説明、アセンブラ言語の仕様及び表計算ソフトの機能・用語は、この冊子の末尾を参照してください。
10. 試験時間中、机の上に置けるものは、次のものに限り、
なお、会場での貸出しは行っていません。
受験票、黒鉛筆及びシャープペンシル（B又はHB）、鉛筆削り、消しゴム、定規、時計（時計型ウェアラブル端末は除く。アラームなど時計以外の機能は使用不可）、ハンカチ、ポケットティッシュ、目薬
これら以外は机の上に置けません。使用もできません。
11. 試験終了後、この問題冊子は持ち帰ることができます。
12. 答案用紙は、いかなる場合でも提出してください。回収時に提出しない場合は、採点されません。
13. 試験時間中にトイレへ行きたくなったり、気分が悪くなったりした場合は、手を挙げて監督員に合図してください。

試験問題に記載されている会社名又は製品名は、それぞれ各社又は各組織の商標又は登録商標です。
なお、試験問題では、™ 及び ® を明記していません。