FE

# 平成28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ません。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。
   (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。

〔問3，問4，問6，問7，問9を選択した場合の例〕

   (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 | ㋐ ㋑ ● ㋓ ㋔ ㋕ ㋖ ㋗ ㋘ ㋙ |
|---|---|---|

裏表紙の注意事項も，必ず読んでください。



C　COBOL　Java　アセンブラ　表計算

平成 28 年 4 月 17 日実施

# 正　誤　表

平成 28 年度春期　基本情報技術者試験　午後問題　問 7

| ページ | 行 | 誤 | 正 |
|---|---|---|---|
| 32 | 表 2<br>上から<br>3 行目 | 1, 240 人 | 1, 240 回 |

〔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 | Web サーバに対する不正侵入とその対策 |

●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 | リスト構造で管理されているセルとガーベジコレクタ |
| 問 3 | データベース | 遊園地の入園者情報を管理する関係データベース |
| 問 4 | ネットワーク | イーサネットを介した通信 |
| 問 5 | ソフトウェア設計 | スマートフォンを用いた店舗検索システム |
| 問 6 | プロジェクトマネジメント | ソフトウェアパッケージ導入時の調達先選定 |
| 問 7 | システム戦略 | 販売データの分析 |

●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 | 簡易メモ帳のメモリ管理 |

●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） | フラクタル図形の描画 |
| 問 10 | ソフトウェア開発（COBOL） | 福利厚生施設の宿泊予約管理 |
| 問 11 | ソフトウェア開発（Java） | “すべきこと”の管理 |
| 問 12 | ソフトウェア開発（アセンブラ） | 10,000 m 競走のチーム別順位の表示 |
| 問 13 | ソフトウェア開発（表計算） | 販売分析 |

## 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。 |
| 処理 | ・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<br>↕<br>▼ 処理 | 単岐選択処理を示す。<br>条件式が真のときは処理を実行する。 |
| | ▲ 条件式<br>↕<br>処理 1<br>処理 2<br>▼ | 双岐選択処理を示す。<br>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<br>↕<br>処理<br>■ | 前判定繰返し処理を示す。<br>条件式が真の間，処理を繰り返し実行する。 |
| | ■<br>↕<br>処理<br>■ 条件式 | 後判定繰返し処理を示す。<br>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<br>↕<br>処理<br>■ | 繰返し処理を示す。<br>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 |
| 乗除演算 | ×，÷，％ | ↑ |
| 加減演算 | ＋，－ | |
| 関係演算 | ＞，＜，≧，≦，＝，≠ | |
| 論理積 | and | ↓ |
| 論理和 | or | 低 |

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　Web サーバに対する不正侵入とその対策に関する次の記述を読んで，設問に答えよ。

A 社は，口コミによる飲食店情報を収集し，提供する会員制サービス業者である。会員制サービスを提供するシステム（以下，A 社システムという）を図1に示す。

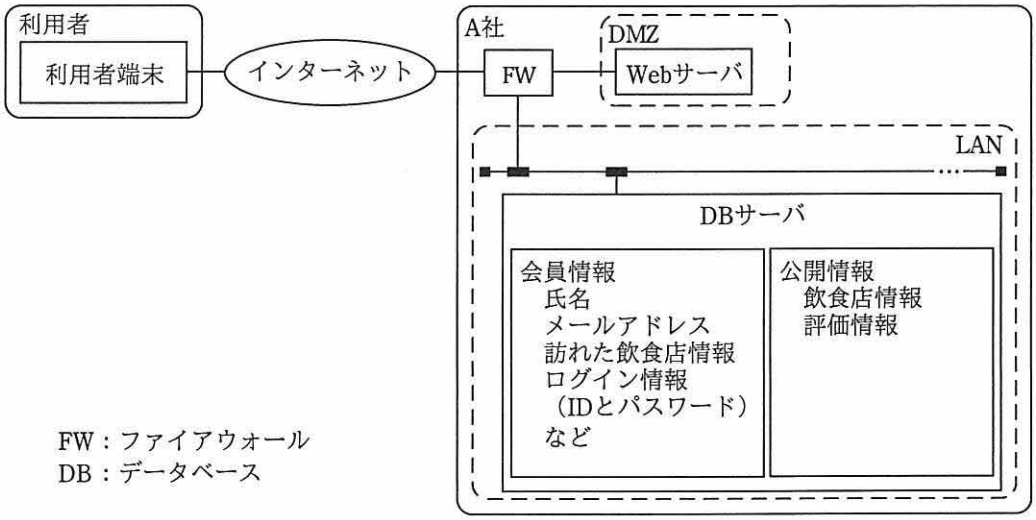



図1　A 社システム

(1) FW，Web サーバ及び DB サーバがあり，スマートフォンなどの利用者端末とはインターネットを介して接続されている。

(2) Web サーバは DMZ に置かれており，DB サーバは LAN に置かれている。また，利用者端末から Web サーバへの接続には，セキュリティを考慮して TLS を用いている。

(3) 会員登録を行った利用者（以下，会員という）には，ID とパスワードが発行される。

(4) DB サーバには，会員情報（氏名，メールアドレス，訪れた飲食店情報，ログイン情報（ID とパスワード）など）と公開情報（飲食店情報，評価情報）が保管されている。

(5) 会員は，公開情報を閲覧することができる。また，Web サーバにログインすることで，DB サーバに保管してある自分の会員情報と自らが書き込んだ公開情報の更新，及び新しい公開情報の追加が行える。

(6) 非会員は，公開情報の閲覧だけができる。

(7) 会員が Web サーバにログインするには，ID とパスワードが必要であり，A 社システムは DB サーバに保管してあるログイン情報を用いて認証する。

(8) Web サーバ及び DB サーバでは，それぞれでアクセスログ（以下，ログという）が記録されており，システム管理者が定期的に内容を確認している。また，システム管理者は，通常，LAN から Web サーバや DB サーバにアクセスして，メンテナンスを行っている。

なお，外部から Telnet や SSH で Web サーバに接続して，インターネットを介したリモートメンテナンスが行えるようにしてあるが，現在はリモートメンテナンスの必要性はなくなっている。

ある日，システム管理者が，ログの確認において，通常とは異なるログが記録されているのを発見した。そのログを詳しく調査したところ，システム管理者以外の者が管理者 ID と管理者パスワードを使って Web サーバに不正侵入したことが明らかになった。

そこで，システム管理者は上司と相談し，会員制サービスを直ちに停止した。次に，今回の不正侵入に対する被害状況の特定と対策の検討を行った。不正侵入による被害状況と対策の一部を抜粋したものを表 1 に示す。

表 1　不正侵入による被害状況と対策（抜粋）

| 被害状況 | 対策 |
|---|---|
| Web サーバへの不正侵入があったことが確認された。秘密鍵への不正アクセスがあったかは確認できなかった。 | a |
| FW を経由し，Web サーバに不正侵入され，さらにそこから DB サーバに不正侵入された。 | リモートメンテナンス用ポートについて，<br>b |
| 一部の会員については会員情報が漏えいしたことが分かっているが，それ以外の会員については漏えいの有無を特定できていない。 | パスワードを変更することにし，<br>c |

また，パスワードの変更に合わせて，パスワードの強度（パスワードの候補数）の検討を行った。これまでパスワードは，英小文字 26 文字だけを受け付け，長さは 6 文字だった。これに対し，他の 3 通りのパスワードの強度を比較した。その比較結果を表 2 に示す。

表 2　パスワードの強度比較（抜粋）

| パスワードとして受け付ける文字種と長さ | 強度の比較 |
|---|---|
| (a) 英小文字，6 文字（不正侵入前の設定） | － |
| (b) 英小文字，8 文字 | (a)と比較して [ d ] 倍 |
| (c) 英大文字・英小文字，8 文字 | (b)と比較して [ e ] 倍 |
| (d) 英大文字・英小文字・数字・記号，8 文字 | (c)と比較して更に多い |

この強度の比較結果を踏まえ，次のように A 社システムを変更し，対策を実施した後に会員制サービスを再開することにした。

(1)　パスワードの文字種としては，英大文字と英小文字，数字，記号を受け付ける。

(2)　長さが 8 文字以上 16 文字以下から成るパスワードを受け付ける。

(3)　辞書に登録されている文字列など推測されやすいパスワードは受け付けない。

設問　表1，2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　TLS を使用していても不正侵入が行われたことから，TLS の使用を直ちに中止し，通常の HTTP で通信を行う。

イ　新たな秘密鍵と公開鍵を生成し，その鍵に対する公開鍵証明書の発行手続を行う。

ウ　公開鍵証明書の再発行手続を行い，同じ秘密鍵を使用する。

エ　秘密鍵へのアクセスが確認できていないことから，秘密鍵の変更や公開鍵証明書の再発行は行わず，念のため秘密鍵の保管場所を，ネットワーク経由でアクセスできないディレクトリに変更する。

bに関する解答群

ア　Telnet や SSH 以外に HTTP も利用できるようにするために，HTTP のポートを開放する。

イ　インターネットからのアクセスを FW で禁止し，Telnet や SSH のポートは閉じる。

ウ　システム管理者がどこからでもすぐに A 社システムのメンテナンスができるように，Telnet や SSH のポートの開放は継続する。

エ　パスワードや A 社システムの実装情報の漏えいを防ぐために，Telnet のポートは閉じ，SSH に限定してポートを開放する。

cに関する解答群

ア　管理者パスワードは変更し，全会員にパスワードの変更を依頼する。

イ　管理者パスワードは変更し，漏えいした会員だけにパスワードの変更を依頼する。

ウ　管理者パスワードはそのままにし，全会員にパスワードの変更を依頼する。

エ　管理者パスワードはそのままにし，漏えいした会員だけにパスワードの変更を依頼する。

dに関する解答群

ア　2×8　　イ　26　　ウ　2×26　　エ　7×8　　オ　10×26　　カ　$26^2$

eに関する解答群

ア　2　　イ　2×8　　ウ　26　　エ　208　　オ　$2^8$　　カ　$26^8$

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　リスト構造で管理されているセルとガーベジコレクタに関する次の記述を読んで，設問1，2に答えよ。

あるアプリケーションプログラム（以下，プログラムという）の実行環境では，プログラムからの要求によって，データ領域をセルとして提供する役割を担う供給源を実装している。プログラムは，使用中のセルをリストで管理している。

〔セルの構造とその管理方法〕

(1)　図1に示すように，セルの構造は，ガーベジコレクタ（以下，GCという）が使用するマークビットという1ビットのフラグ，別のセルへのポインタ，及び固定長のデータ領域をもつ。マークビットの初期値は0である。ただし，以降の説明ではデータ領域を省略する。

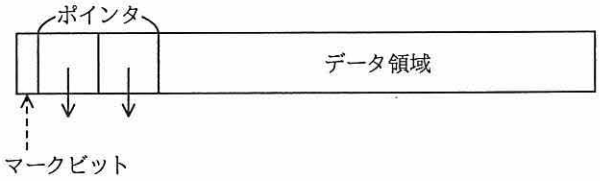



図1　セルの構造

(2)　図2に示すように，セルは，メモリ領域の中に連続して格納される。別のセルを指していないポインタには，NULL（終端記号）が格納されており，斜線を引いて表記している。

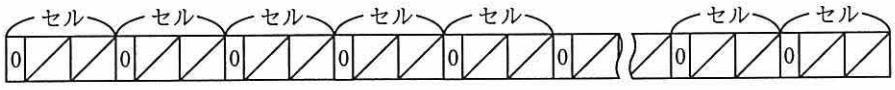



図2　メモリ領域の中に連続して格納されているセルのイメージ

(3) 供給源の機能は次のとおりである。

① プログラムの実行開始時には，全てのセルは未使用セルとして，供給源が管理している。ここで，未使用セルのマークビットは 0 であり，ポインタには NULL が格納されている。

② プログラムから取得要求があると，管理している未使用セルを提供する。

〔プログラムでのセルの使用方法〕

(1) プログラムは新たにセルが必要になると，供給源に要求を出し，未使用セルを受け取り，LIVE から始まるリスト（以下，LIVE リストという）に登録して使用する。LIVE リストに登録され，プログラムが使用している状態のセルを LIVE セルという。

(2) プログラムで不要になったセルは，LIVE リストから切り離すだけで，供給源には返却しない。この状態のセルをガーベジという。ガーベジのままでは使用することができない。LIVE セルがガーベジになる例を図 3 に示す。LIVE セルを実線，ガーベジを破線で表している。図 3 (a) では，LIVE からのポインタ X はセル A を指しているが，プログラムの処理が進み，図 3 (b) では，セル C を指すようになった。このとき，セル A とセル B は，LIVE リストから切り離され，供給源の管理下の未使用セルでもなく，ガーベジになっている。

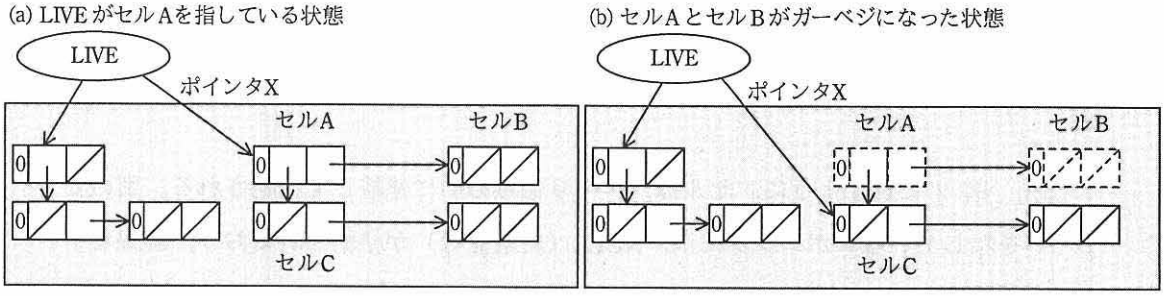


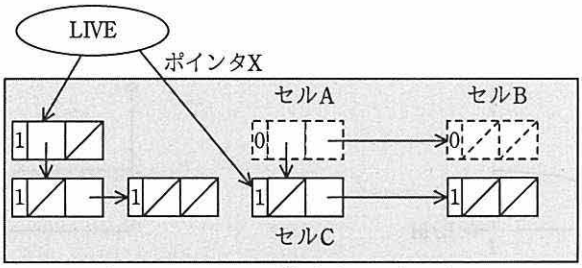

図 3 LIVE リストとガーベジの例

(3) プログラムからのセル取得要求が繰り返されると，供給源が管理する未使用セルが無くなってしまうことがある。このとき，プログラムが新たなセルを要求しても，供給源からセルを受け取ることができず，プログラムの実行が停止する。そこで，ガーベジを未使用セルにする GC が起動される。

設問1　GCは，LIVEからのポインタをたどることによって，全てのLIVEセルを特定することができる。また，メモリ領域を先頭から走査することによって，全てのセルを識別することができる。未使用セルが無くなった状態では，特定されたLIVEセル以外のセルは全てガーベジである。GCの処理方式の一つとして，マークアンドスイープという方式がある。この方式によるGC処理に関する次の記述中の［　　］に入る正しい答えを，解答群の中から選べ。

〔マークアンドスイープ方式によるGC処理の説明〕

(1)　LIVEからのポインタをたどって到達できる全てのLIVEセルのマークビットを1にする。この処理をマーキングという。マーキング終了後のセルの状態の例を図4に示す。




注記　図中の1はマーキングされたことを示す。

図4　マーキング終了後のセルの状態の例

(2)　次に，セルが格納されているメモリ領域中の全てのセルを重複が無いように走査し，各セルのマークビットの値を調べる。マークビットが0ならば，そのセルはガーベジであるので，供給源に未使用セルとして返却する。マークビットが1であれば0にする。以上の処理をスイープという。マーキング終了後で，スイープ開始前のメモリ領域の状態の例を図5に示す。

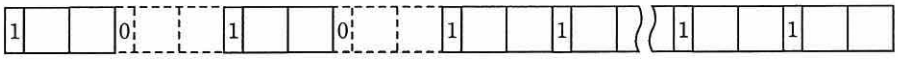



図5　スイープ開始前のメモリ領域の状態の例

GCが作動している間，プログラムの実行は一時中断される。中断の時間はマーキングの処理量とスイープの処理量に依存する。GCの対象となるメモリ領域のセ

ルの数をM，GC開始時のLIVEセルの数をLとしたとき，マーキングの処理量は［ a ］に比例する。また，スイープの処理量は［ b ］に比例する。

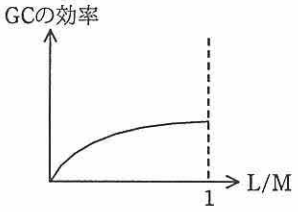
この GC の 1 回の作動で再び利用できるようになるガーベジの数は［ c ］と表せる。Mに対するLの割合（L/M）を横軸に，単位時間当たりに再び利用できるようになるセルの数（GC の効率）を縦軸にとってグラフを描いた。GC の効率と L/M の関係を示すグラフの形として，最も適切なものは［ d ］である。

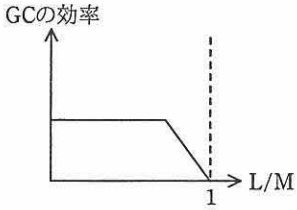
a～cに関する解答群

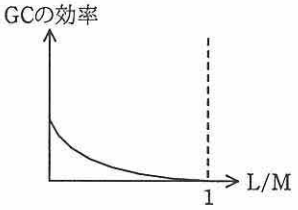
ア　L　　イ　L−M　　ウ　M

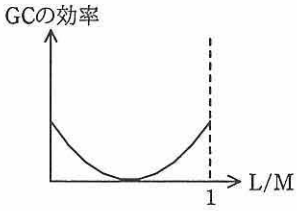
エ　M+L　　オ　M−L

dに関する解答群

ア



イ



ウ



エ



設問２　プログラムの実行開始後，初めて GC が作動した。マーキング終了後の状態において，マークビットが１であるセルについての記述として正しい答えを，解答群の中から選べ。

解答群

ア　供給源の管理下にある。

イ　プログラムが使用中である。

ウ　プログラムの処理が進む過程で，LIVE リストから切り離された。

エ　マーキングに続いて行われるスイープで，供給源に返却されることがある。

問３　遊園地の入園者情報を管理する関係データベースに関する次の記述を読んで，設問１～４に答えよ。

遊園地 Y では，各アトラクションの入り口に IC カードの読取り機を設置して，入園者の利用状況を収集するシステムを導入した。入園者は，全てのアトラクションを追加料金なしで利用できるパスポートか，アトラクション利用の都度，料金が課金される入園券のいずれかを購入し，対応する IC カードを受け取る。IC カードは退園時に料金を精算してから返却する。ここで，入園者は退園まで遊園地を出ることはないものとする。

遊園地 Y では，システム導入前は入園者の情報を図 1 に示す表で構成されるデータベースで管理していた。下線付きの項目は主キーを表す。

入園者表

| 入園者番号 | 券種 | 入園日 | 入園時刻 |
|---|---|---|---|
| 00025987 | 02 | 20150115 | 084517 |
| ⋮ | ⋮ | ⋮ | ⋮ |

券種表

| 券種 | 券種名 | 料金 |
|---|---|---|
| 01 | 入園券 | 500 |
| 02 | 小学生パスポート | 2000 |
| ⋮ | ⋮ | ⋮ |

図１　システム導入前の表の構成とデータの格納例

設問１　システムの導入に当たり，データベースの表を図２に示すとおり再設計した。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。

入園者表に退園時刻の項目を追加することによって，［ a ］が分かるようになった。また，利用表を追加することによって，［ b ］が分かるようになった。各アトラクションには一意のアトラクション番号を割り振って，利用表から分離したアトラクション表を作成した。

入園者表

| 入園者番号 | 券種 | 入園日 | 入園時刻 | 退園時刻 |
|---|---|---|---|---|
| 00025987 | 02 | 20150115 | 084517 | 150851 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

券種表

| 券種 | 券種名 | 料金 |
|---|---|---|
| 01 | 入園券 | 500 |
| 02 | 小学生パスポート | 2000 |
| ⋮ | ⋮ | ⋮ |

利用表

| 入園者番号 | 利用時刻 | アトラクション番号 |
|---|---|---|
| 00025987 | 091806 | 04 |
| 00025987 | 102611 | 05 |
| 00025992 | 102757 | 05 |
| ⋮ | ⋮ | ⋮ |

アトラクション表

| アトラクション番号 | アトラクション名 | 料金 |
|---|---|---|
| 01 | ジェットコースター | 500 |
| ⋮ | ⋮ | ⋮ |

図２　システム導入後の表の構成とデータの格納例

a，bに関する解答群

ア　アトラクションの待ち時間　　イ　休園日

ウ　入園者数　　エ　入園者のアトラクション利用状況

オ　入園者の滞在時間

設問２　アトラクションごとの延べ利用者数を表示する。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT　利用表.アトラクション番号，アトラクション表.アトラクション名，
        [　　c　　]
  FROM　利用表，アトラクション表
  WHERE　利用表.アトラクション番号 = アトラクション表.アトラクション番号
  GROUP BY　利用表.アトラクション番号，アトラクション表.アトラクション名
```

解答群

ア　AVG(利用表.アトラクション番号)

イ　COUNT(*)

ウ　MAX(利用表.アトラクション番号)

エ　SUM(利用表.アトラクション番号)

設問３　入園券でアトラクションを利用した入園者のうち，退園時の精算において2,000 円以上を支払った入園者について，精算額が多い入園者から降順に，入園者番号と精算額を表示する。入園券の購入者は入園者表の券種に“01”が設定されている。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT 入園者表.入園者番号, SUM(アトラクション表.料金) AS 支払金額
  FROM 入園者表, 利用表, アトラクション表
  WHERE 入園者表.入園者番号 = 利用表.入園者番号 AND
        利用表.アトラクション番号 = アトラクション表.アトラクション番号 AND
        [          d          ]
  ORDER BY 支払金額 DESC
```

解答群

ア　入園者表.券種 = '01' AND
　　アトラクション表.料金 >= 2000
　　GROUP BY 利用表.アトラクション番号

イ　入園者表.券種 = '01'
　　GROUP BY 入園者表.入園者番号
　　HAVING MAX(アトラクション表.料金) >= 2000

ウ　入園者表.券種 = '01'
　　GROUP BY 入園者表.入園者番号
　　HAVING SUM(アトラクション表.料金) >= 2000

エ　入園者表.券種 = '01'
　　GROUP BY 利用表.アトラクション番号
　　HAVING SUM(アトラクション表.料金) >= 2000

設問4　パスポートを購入する際に提示することで料金が割引になる会員証を発行することになった。そこで，図3に示すとおり，会員情報を格納する会員表を作成し，入園者表に会員番号を格納する項目を追加する。次の SQL 文で抽出できるようになる会員についての正しい答えを，解答群の中から選べ。ここで，1回も入園していない会員はいないものとする。また，会員登録をしていない入園者の場合，入園者表の会員番号にはNULLを設定する。

会員表

| 会員番号 | 氏名 | 住所 | 生年月日 | 入会日 |
|---|---|---|---|---|
| 000108 | 情報太郎 | 東京都文京区○○○ | 19920901 | 20160114 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

入園者表

| 入園者番号 | 会員番号 | 券種 | 入園日 | 入園時刻 | 退園時刻 |
|---|---|---|---|---|---|
| 00028707 | 000108 | 04 | 20160218 | 100555 | 182021 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

図3　会員情報を追加した表の構成とデータの格納例

```
SELECT 会員表.会員番号, 会員表.氏名
  FROM 会員表, 入園者表
  WHERE 会員表.会員番号 = 入園者表.会員番号
    GROUP BY 会員表.会員番号, 会員表.氏名
    HAVING MAX(入園者表.入園日) < '20160101'
```

解答群

ア　2015年以前に1回も入園していない会員

イ　2016年以降に1回以上，入園した会員

ウ　2016年以降に1回だけ，入園した会員

エ　2016年以降に1回も入園していない会員

問4　イーサネットを介した通信に関する次の記述を読んで，設問1，2に答えよ。

IP ネットワークにおいて，あるホストが別のホストと通信する場合，通信相手のホストの IP アドレスを指定して通信する。下位層にイーサネットを用いるときには，通信相手のホストの MAC アドレス，又は通信相手のホストに到達可能なルータの MAC アドレスが必要になる。しかし，IP ネットワークで通信を行うアプリケーションでは，通信相手の IP アドレスやホスト名を明示的に指定することはあっても，MAC アドレスを明示的に指定することはない。したがって，IP アドレスを手掛かりとして必要な MAC アドレスを得るために，IP ネットワークでは ARP（アドレス解決プロトコル）というプロトコルが用いられる。

〔MAC アドレスに関する説明〕

イーサネットと IP を OSI 基本参照モデルに当てはめた場合，イーサネットは物理層とデータリンク層に該当し，IP はネットワーク層に該当する。つまり，IP ネットワークでの通信で取り扱う IP データグラムを，下位層のイーサネットで送信するためには，IP データグラムを［　a　］したイーサネットフレームを送信する。このとき，イーサネットフレームの宛先を表すアドレスとして用いられるのが MAC アドレスである。MAC アドレスの長さは 48 ビットであり，表現可能なアドレスの個数は［　b　］個となる。

なお，ここでは MAC アドレスを表記する際，8 ビットごとに 2 桁の 16 進数 00 ～ FF で表し，それぞれの間はコロンで区切る。例えば，00:53:00:12:C5:8A のように表す。

〔ARP の機能の説明〕

IP アドレスを基に MAC アドレスを得る ARP の機能は，問合せとして“ARP 要求”を送信し，それに対する回答として“ARP 応答”を受け取ることで実現される。

例えば，セグメント 10.1.1.0/24 において，ホスト A（IP アドレス 10.1.1.10，MAC アドレス 00:53:00:DA:C7:0B）がホスト B（IP アドレス 10.1.1.20，MAC アドレス 00:53:00:EC:17:27）宛てに IP データグラムを送信しようとしたとき，ホスト B の MAC アドレスは ARP によって，次のようにして得られる。

(1) ホスト A は，IP アドレス 10.1.1.20 に対する ARP 要求を送信する。このとき，ARP 要求は［ c ］される。

(2) ARP 要求を受け取ったホスト B は，その ARP 要求が自分の IP アドレスに対する問合せであることを確認すると，自分の IP アドレス 10.1.1.20 と MAC アドレス 00:53:00:EC:17:27 を格納した ARP 応答を送信する。同じ ARP 要求を受け取ったその他のホストは，それが自分の IP アドレスに対する問合せではないので，無視する。

(3) ホスト A は，ホスト B が送信した ARP 応答を受け取ることによって，IP アドレス 10.1.1.20 に対応する MAC アドレスが 00:53:00:EC:17:27 であることが分かる。

ホスト A は，ホスト B の IP アドレスと得られた MAC アドレスの対応付けをキャッシュする。キャッシュが破棄されるまで 10.1.1.20 宛ての IP データグラムを送る際，イーサネットフレームの宛先 MAC アドレスとして 00:53:00:EC:17:27 を使用する。

設問 1　本文中の［　　］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　宛先として格納　　イ　送信元として格納　　ウ　データ部に格納
エ　プリアンブルに格納　　オ　ヘッダ部に格納

b に関する解答群

ア　48　　イ　254　　ウ　256
エ　$2^{32}$　　オ　$2^{48}$

c に関する解答群

ア　TCP セグメントとして送信　　イ　UDP データグラムとして送信
ウ　ブロードキャスト　　エ　ホスト B の MAC アドレス宛てに送信
オ　ユニキャスト

**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。

図1は，ある企業の社内ネットワークの構成（一部）である。

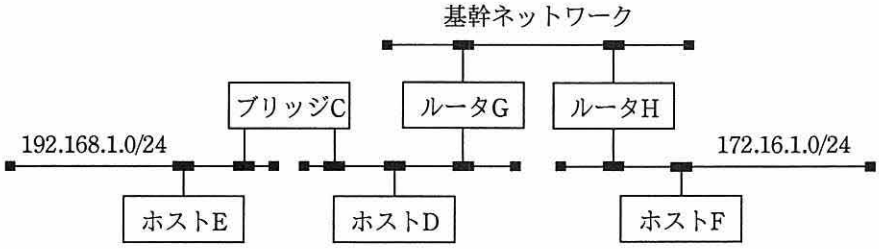



**図1　ある企業の社内ネットワークの構成（一部）**

このネットワークにおいて，ホストDが，幾つかの宛先にIPデータグラムを送信しようとするとき，ホストDはARPによって送信に必要なMACアドレスを得る。ここで，ホストDがIPデータグラムを送信しようとしたとき，宛先のMACアドレスはホストDにキャッシュされていないものとする。

ホストEに対してIPデータグラムを送信しようとするとき，ホストDは［　d　］のIPアドレスに対するARP要求を送信する。

ホストFに対してIPデータグラムを送信しようとするとき，ホストDは［　e　］のIPアドレスに対するARP要求を送信する。

解答群

| | | |
|---|---|---|
| ア　ブリッジC | イ　ホストD | ウ　ホストE |
| エ　ホストF | オ　ルータG | カ　ルータH |

問５　スマートフォンを用いた店舗検索システムに関する次の記述を読んで，設問 1～3 に答えよ。

E 社は首都圏に約 50 店舗をもつ生鮮食料品の販売店である。来店客数の増加を目的として，指定した市区町村にある店舗情報をスマートフォンを用いて検索するシステムを構築することになった。このシステムは，スマートフォン上で動作するアプリケーションソフトウェア（以下，アプリという），Web サーバ及びファイルサーバから構成されている。

〔システム機能の要件〕

(1)　利用者はブラウザを起動し，本システムの URL を指定して Web ページを表示する。Web ページには，検索を行うための市区町村名の入力欄がある。

(2)　利用者が入力した市区町村名で店舗検索し，その市区町村にある店舗名と住所，電話番号，お気に入り数及び特売情報を抽出して，お気に入りボタンとともに表示する。利用者は表示された店舗を気に入ったならば，お気に入りボタンを押す。

(3)　検索対象となる E 社の店舗データは，ファイルサーバ上の店舗データファイルに保管されている。店舗データファイルのレコード様式を図 1 に示す。

| 店舗番号 | 店舗名 | 市区町村名 | 住所 | 電話番号 | 特売情報 |
|---|---|---|---|---|---|

図 1　店舗データファイルのレコード様式

(4)　店舗ごとに，お気に入りボタンが押されるたびに 1 が加算されるお気に入り数の値を，お気に入りデータファイルに保持する。お気に入り数は，Web ページで店舗のお気に入りボタンが押された回数である。お気に入りデータファイルのレコード様式を図 2 に示す。

| 店舗番号 | お気に入り数 |
|---|---|

図 2　お気に入りデータファイルのレコード様式

〔処理の流れ〕

本システムの処理は，次の処理Aから処理Cの順番で行われる。

処理A：店舗を検索するための市区町村名の入力欄と，検索実行ボタンが含まれるWeb画面を生成し，表示する。

処理B：利用者が入力した“市区町村名”をキーに，読み込んだ店舗データファイルとお気に入りデータファイルを検索する。検索で得られた店舗名，住所，電話番号，特売情報，お気に入り数，お気に入りボタン，再検索を行うための市区町村名の入力欄，及び検索実行ボタンを検索結果画面に表示する。市区町村に複数の店舗がある場合は列挙して表示する。検索実行ボタンが押された場合は，再び処理Bが行われる。

処理C：利用者が検索結果画面に表示されている各店舗のお気に入りボタンを押すと，お気に入りデータファイルの該当する店舗のお気に入り数の値に1を加える処理を行う。続いて，お気に入りデータファイルから新しいお気に入り数を読み込んで，検索結果画面を再表示する。新たに店舗のお気に入りボタンが押された場合は，処理Cを継続する。

〔実装方法の設計〕

システム機能の要件を満たし，処理の流れを実装する方式として，Web方式とWebアプリ方式の二つを比較検討する。それぞれの特徴を表1に，それぞれの方式のシーケンス図を図3，図4に示す。図において，上段が空欄の矢印はメッセージを省略している。

表1　Web方式とWebアプリ方式の特徴

| 比較項目 | Web方式 | Webアプリ方式 |
|---|---|---|
| スマートフォン上で起動されるアプリ | ブラウザ | ブラウザと，ブラウザが読み込んだWebアプリ |
| 処理Bを行う主な場所 | Webサーバ | スマートフォン |
| 画面の生成 | ブラウザ上の画面を切り替えるたびに，Webサーバが生成した画面の情報をブラウザが読み込み，表示する。 | ブラウザが読み込んだWebアプリが，まず全ての店舗データファイルの内容をダウンロードする。その店舗データを基にしてWebアプリが画面の情報を生成し，ブラウザが表示する。このとき，画面を切り替えるたびに必ずしもWebサーバにアクセスしなくともよい。 |

〔シーケンス図〕

(1)　Web 方式で実装する場合

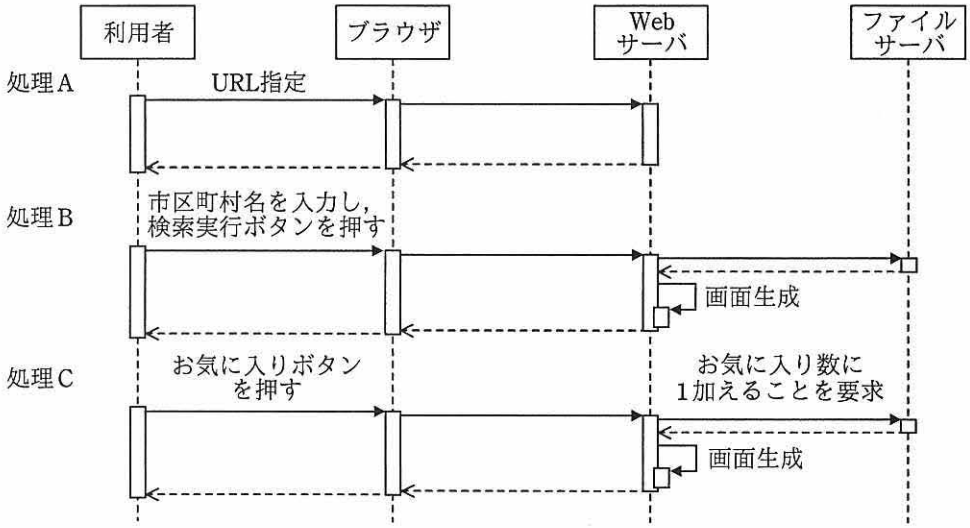



図 3　Web 方式のシーケンス図

処理 B ではファイルサーバは［　a　］から得た情報を，処理 C ではファイルサーバは［　b　］から得た情報を Web サーバに返却する。

(2)　Web アプリ方式で実装する場合

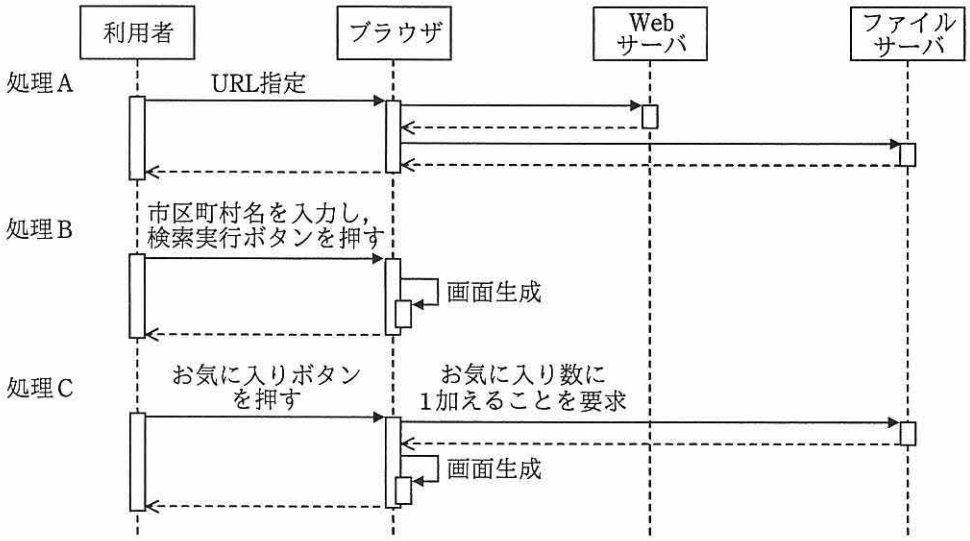



図 4　Web アプリ方式のシーケンス図

処理 A ではファイルサーバは［ a ］から得た情報を，処理 C ではファイルサーバは［ b ］から得た情報をブラウザに返却する。

ここで，Web 方式と Web アプリ方式の比較検討では，スマートフォンでの利便性を考慮して，次の要件を追加した。

〔新規要件〕
スマートフォンなどのモバイル端末で用いる無線通信は，有線通信のネットワークに比べて通信不能（以下，圏外という）になる頻度が高い。そのため，スマートフォンなどのモバイル端末で動作するアプリは，圏外になっても検索の操作が継続できるように設計する。

Web 方式の場合，表示する画面は［ c ］ので，処理 B の実行中に［ d ］と［ e ］の間の無線通信が圏外になると，次画面を読み込めなくなる。

一方，Web アプリ方式の場合は，通信を行わなくても，ブラウザが読み込んだ Web アプリで処理を行うことができる。処理 B の実行中に［ d ］と［ e ］の間の無線通信が圏外になって切断されても，検索の操作を続けることが可能となる。

したがって，Web アプリ方式で実装することにした。

**設問 1**　本文中の［　　］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　お気に入りデータファイル

イ　店舗データファイル

ウ　店舗データファイルとお気に入りデータファイル

cに関する解答群

ア　ブラウザがWebサーバに都度要求し，Webサーバが生成する必要がある

イ　ブラウザがファイルサーバに都度要求し，ファイルサーバから取得する必要がある

ウ　ブラウザが生成するが，生成中に圏外を検知すると停止する

エ　ブラウザが生成し，都度表示させる必要がある

d，eに関する解答群

ア　Webサーバ　　イ　ファイルサーバ　　ウ　ブラウザ

エ　利用者

設問2　処理Bでお気に入り数を抽出するための必須項目として正しい答えを，解答群の中から選べ。

解答群

ア　店舗番号　　イ　店舗名　　ウ　市区町村名

エ　店舗番号，店舗名　　オ　店舗番号，市区町村名　　カ　店舗名，市区町村名

設問3　Web方式とWebアプリ方式ともに，ファイルサーバから情報を取得する。ファイルサーバから見て情報の要求元として正しい答えを，解答群の中から選べ。

解答群

| | Web方式の場合 | Webアプリ方式の場合 |
|---|---|---|
| ア | Webサーバ | Webサーバ |
| イ | Webサーバ | ブラウザ |
| ウ | ブラウザ | Webサーバ |
| エ | ブラウザ | ブラウザ |

問6　ソフトウェアパッケージ導入時の調達先選定に関する次の記述を読んで，設問 1，2に答えよ。

C 社では，会計システムの再構築を考えており，プロジェクト計画の策定中である。本プロジェクトの目的は，業務の効率向上と，標準化による運用保守コストの削減である。目的実現のための方式案として，ソフトウェアパッケージ（以下，パッケージという）の活用を検討している。ここで，業務プロセス及び運用フローについては，パッケージの機能だけで実現する計画である。

〔比較検討の進め方〕

パッケージの比較検討は，次の手順で進める。

手順 1：C 社の業務要件に近いと思われるパッケージを調査した結果，有力と判断した 3 種類のパッケージ製品について，要件定義から本番稼働までの概算見積を含む提案を，それぞれのパッケージベンダ（L 社，M 社，N 社）に依頼する。各社提案について，評価項目に基づいた評価を実施する。

手順 2：提案されたパッケージを導入した場合のリスク分析を行い，対策を検討した上で，リスクに対する評価を実施し，総合的に判断する。

設問1　本プロジェクトのプロジェクトマネージャ（PM）は，手順 1 に従い，L 社，M 社，N 社からの提案内容を評価した。その結果を表 1 に示す。評価項目における各社の評点は，評価項目ごとの評価基準に重みを乗じて算出し，評点の合計を総合評価とする。PM が行った作業に関する次の記述中及び表中の［　　］に入れる正しい答えを，解答群の中から選べ。

表１　各社提案の評価

| | | 評価項目 | | | | | 総合評価 |
|---|---|---|---|---|---|---|---|
| | | パッケージの機能の適合度と拡張性 | | ベンダの業界知識 | ベンダのプロジェクト経験数 | 概算見積金額（万円） | |
| 評価基準 | | 要求機能数 50 に対する適合数<br>80%以上：4<br>60%以上～80%未満：2<br>60%未満：0 | 拡張性の有無<br>有：2<br>無：0 | 高：4<br>中：2<br>低：0 | 多：4<br>中：2<br>少：0 | C 社予算上限 5,000 万円に対する見積金額<br>80%以下：4<br>80%超 100%以下：2<br>100%超：0 | |
| 重み | | 20 | 10 | 30 | 20 | 30 | |
| L社 | 提案内容 | 適合数：42 | 有 | 高 | 多 | 4,700 | |
| | 評点 | 80 | 20 | 120 | 80 | 60 | 360 |
| M社 | 提案内容 | 適合数：44 | 有 | 低 | 多 | 3,800 | |
| | 評点 | 80 | 20 | 0 | 80 | a | |
| N社 | 提案内容 | 適合数：32 | 有 | 高 | 中 | 4,500 | |
| | 評点 | | 20 | 120 | 40 | 60 | |

注記　網掛けの部分は表示していない。

表１で総合評価の値が最も高いのは［　b　］社であった。

aに関する解答群

ア　0　　イ　60　　ウ　120

bに関する解答群

ア　L　　イ　M　　ウ　N

**設問2**　手順2でPMが行ったリスク評価，及びパッケージ選択に関する次の記述中及び表中の［　　］に入れる最も適切な答えを，解答群の中から選べ。

〔パッケージ導入に関するリスク分析の状況〕

C社のプロジェクト管理規程では，表2に示すリスク評価マトリックスを用いて，発生確率と影響度の積の値で対策の優先度を決定する。値が6以上のリスク項目は，リスク発生時の対策が必要と判断する。プロジェクト計画時に，対策が必要なリスク項目ごとに，概算見積費用の10%を，リスク発生時の対策費用として盛り込むことにしている。PMが表1を基にリスクを洗い出し，評価を行った結果を表3に示す。

表2　リスク評価マトリックス

| 発生確率＼影響度 | | 小 | 中 | 大 |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| 高い | 3 | 3 | 6 | 9 |
| 普通 | 2 | 2 | 4 | 6 |
| 低い | 1 | 1 | 2 | 3 |

（凡例）
- 非常に高い優先度
- 高い優先度
- 低い優先度

表3　PMによるリスク評価の結果

| 項番 | リスク | L社 | | | M社 | | | N社 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 発生確率 | 影響度 | リスク対策内容 | 発生確率 | 影響度 | リスク対策内容 | 発生確率 | 影響度 | リスク対策内容 |
| 1 | 要求機能に対するパッケージの適合度が低いので，業務プロセス，運用フローの変更項目が多く，業務設計が遅延する。 | 低い | 大 | | 低い | 大 | | 普通 | 大 | ［c］ |
| 2 | C社の会計業務に対する理解が進まず，適用可能な業務プロセス，運用フローの合意が取れず，業務設計が遅延する。 | 低い | 中 | | 高い | 中 | | 低い | 中 | |
| 3 | プロジェクト経験数が想定よりも少なく，プロジェクト遂行能力が十分でないので，重大な課題の解決が進まない。 | 低い | 大 | | 低い | 大 | | 普通 | 大 | |

注記　網掛けの部分は表示していない。

E
後

リスク評価の結果から，例えば項番 1 のリスクに対する N 社パッケージへの対応として，プロジェクト立上げ時点に予防措置を講じていたとしても，[ d ]を契機にリスク対策を発動しなければならなくなったときのために，[ e ]万円のリスク対策費用を盛り込む必要があるといえる。

リスク評価結果から必要となるリスク対策費用を各社の概算見積金額に加えて，各社提案の評価をやり直した結果，総合評価の値の大小関係に変わりがないことから，表 1 の総合評価の値が最も大きいパッケージを採択して会計システムを再構築するプロジェクト計画の策定を進めることにした。

c に関する解答群

ア　C 社内のシステムに関するプロジェクトの管理経験を有する，N 社以外のベンダ要員を増員する。

イ　C 社の会計業務に関する業務マニュアルを整備し，N 社の要員に研修を実施する。

ウ　パッケージの機能を C 社の業務向けにカスタマイズするために，N 社の要員を増員する。

エ　パッケージを適用する際の C 社の業務プロセス，運用フローの見直しについて，C 社内の有識者社員から成る検討要員を増員する。

d に関する解答群

ア　解決されずに残っている課題の増加

イ　業務プロセス，運用フローの改訂作業量の増加

ウ　業務プロセス，運用フローのレビューでの指摘の増加

エ　作り込み機能の増加

e に関する解答群

ア　380　　イ　450　　ウ　470　　エ　900

問7　販売データの分析に関する記述を読んで，設問1，2に答えよ。

B社は，洋菓子を店舗販売する小売店である。B社は，来期の事業方針として，顧客が自社の商品を繰り返し購入する回数を増やすことによって，売上を増加させることを考えた。そのために，効果的な販売促進活動につなげることを目的として，既に導入しているID-POSを利用して，顧客一人一人の購買履歴を分析した。

ID-POSは，各顧客に顧客IDを設定し，顧客IDとその顧客の商品の購買履歴を管理するシステムであり，誰が，いつ，何を，幾つ購入したのかが分かる仕組みである。

設問1　顧客の購買履歴の分析に関する次の記述中及び図中の［　　］に入れる適切な答えを，解答群の中から選べ。

B社は，ID-POSの購買履歴を活用し，ある商品カテゴリーを対象に，商品を購入した顧客と，その顧客が同じ商品を再び購入する傾向にあるかどうかの分析を行った。分析の方法として，購入人数と延べ購入回数からリピート率を算出した。

購入人数は，対象期間中に同一顧客が複数回購入したときも1人としてカウントした場合の人数であり，延べ購入回数は，顧客が購入した延べの回数である。例えば，ある対象期間中に5回購入した顧客が2人，3回購入した顧客が1人，1回だけ購入した顧客が3人いた場合，購入人数は6人，延べ購入回数は16回となる（図1参照）。

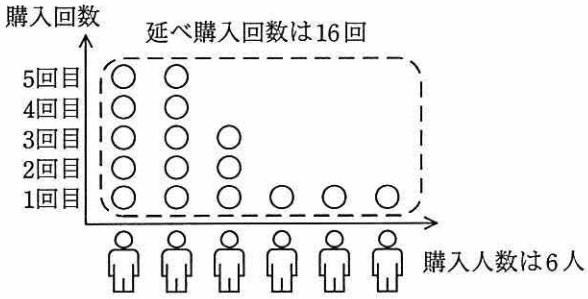



図1　購入人数と延べ購入回数の説明

ここで，リピート率の算出式は，

リピート率 ＝（延べ購入回数－購入人数）÷ 延べ購入回数 × 100

とする。リピート率は，延べ購入回数に対して，2 回目以降の延べ購入回数（＝延べ購入回数 － 購入人数）の割合を示す指標である。図 1 の例では，延べ購入回数が 16 回であり，2 回目以降の延べ購入回数は図 2 に示すように 10 回であるので，リピート率は 62.5％となる。

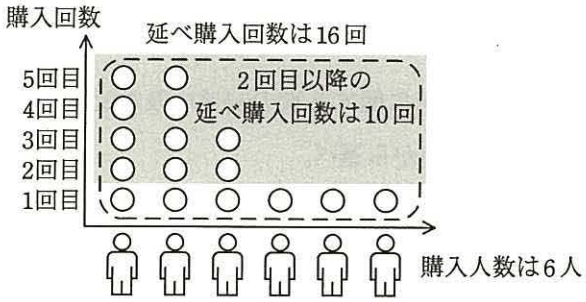



図２　２回目以降の延べ購入回数の説明

B 社の商品のうちで日常的に販売されている商品 S～X について，顧客は，1 回の購入で平均して同一商品を 2 個購入している。また，限られた一部の顧客だけが購入を繰り返すというような購入回数の大きな偏りはなかった。各商品の購入人数と延べ購入回数，売上金額は表 1 のとおりである。

表１　各商品の購入人数，延べ購入回数，売上金額

| 商品 | 購入人数(人) | 延べ購入回数(回) | 売上金額(千円) |
|---|---|---|---|
| S | 4,782 | 5,217 | 2,400 |
| T | 1,321 | 1,729 | 951 |
| U | 1,312 | 3,465 | 2,079 |
| V | 4,732 | 24,121 | 4,824 |
| W | 3,100 | 4,013 | 1,605 |
| X | 1,689 | 2,712 | 1,356 |

これらの商品の購買履歴の分析を行うために，図 3 のバブルチャートを作成した。ここで，バブルの大きさは，売上金額を表す。

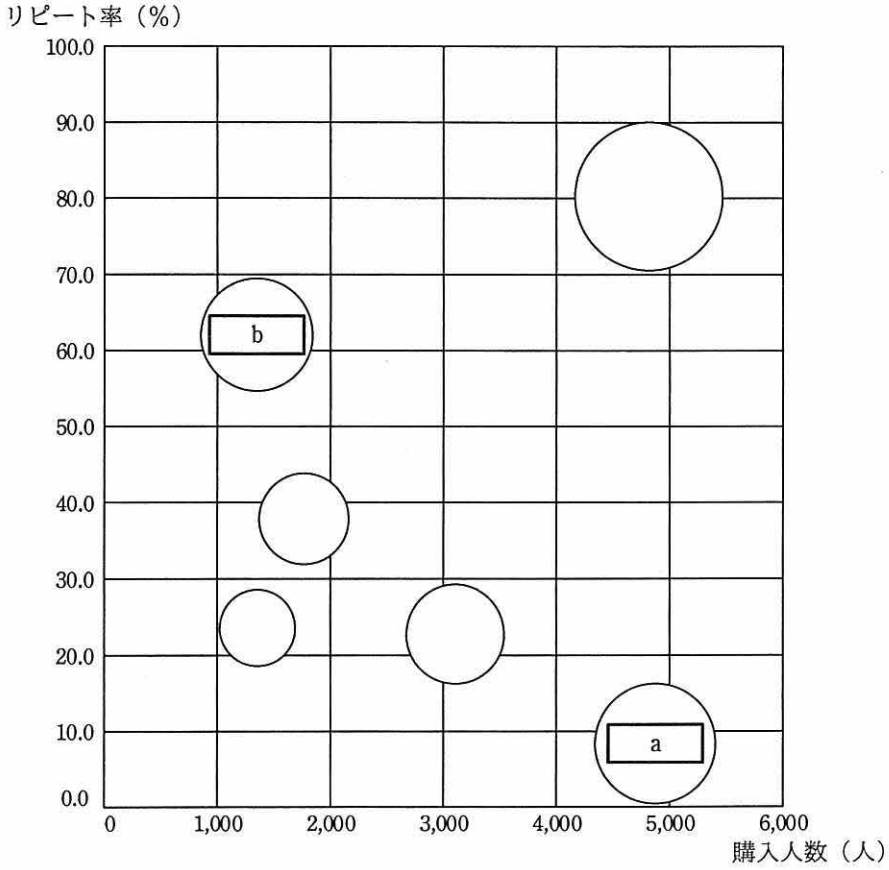



図３　購買履歴の分析結果

今回分析した商品の中で，商品 V は，購入人数が多くリピート率も高いことから，商品 V を繰り返し購入してくれる顧客が多いと考えられる。商品 [ a ] は，購入人数は多いが，リピート率が低いので，商品そのものに魅力がないと考えられ，売上を増加させるためには，[ c ] 必要がある。商品 [ b ] は，購入人数は少ないが，リピート率が高いので，限られた顧客に人気が高い商品と考えられる。したがって，試食会を開催するなどして，[ d ] ことによって [ e ]，売上の増加が見込まれる。

a，b に関する解答群

ア　S　　イ　T　　ウ　U　　エ　W　　オ　X

c，dに関する解答群

ア　商品が欠品しないようにする

イ　商品の素材や味を見直す

ウ　商品の認知度を高める

エ　より価格が高い商品を勧める

eに関する解答群

ア　価格を上げれば　　イ　購入人数が増えれば

ウ　来店客数が増えれば　　エ　リピート率が上がれば

設問２　売上拡大の施策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。

B社では，広告を行って新規顧客を開拓し，その新規顧客にも繰り返し購入してもらうことで売上の増加を図る計画を立て，商品Uについて効果を見積もってみた。費用と効果の見積りを表2に示す。

商品Uの価格は300円，1回の購入で商品Uを平均2個購入するものとする。また，効果見積りは，過去1年間の実績値に対する値とする。

ここで，効果額は，売上金額の増加額から広告費用を引いた額とする。投資効果は，広告費用に対する効果額の割合であり，その割合が1よりも大きければ，投資効果があると評価する。

表２　費用と効果の見積り

| | | |
|---|---|---|
| 費用 | 広告費用 | 300千円 |
| 効果 | 購入人数の増加 | 600人 |
| | 延べ購入回数の増加 | 1,240人 |

この施策によって，商品Uの［　f　］となる。したがって，この施策は，商品Uに関しては［　g　］すると考えられる。

fに関する解答群

ア　売上金額は 1,147 千円となり，効果額は 444 千円

イ　売上金額は 1,412 千円となり，効果額は－968 千円

ウ　売上金額は 2,439 千円となり，効果額は 60 千円

エ　売上金額は 2,823 千円となり，効果額は 444 千円

gに関する解答群

ア　投資効果はあるが，リピート率は低下

イ　投資効果はないが，リピート率は向上

ウ　投資効果はあり，リピート率も向上

エ　投資効果はなく，リピート率も低下

次の問８は必須問題です。必ず解答してください。

問８　次の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。

携帯端末上で稼働する簡易メモ帳の機能のうち，メモの編集処理（メモの追加・削除・変更・移動）を行う部分のプログラムである。図１は，簡易メモ帳に４件のメモ“Aoki”，“Imai”，“Uno”及び“Endo”を登録した場合の表示例である。

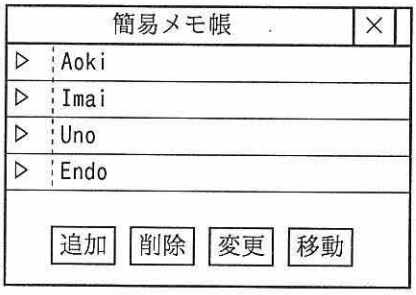



図１　簡易メモ帳の表示例

〔プログラムの説明〕

(1)　メモは，画面に表示可能な１バイトで表現できる文字から成る文字列である。各メモは，文字列の前に，文字列の長さ（0～255）を１バイトの符号なし２進整数の形式で付け加えて格納する。例えば，メモ“Hello!”は，次の形式で格納する（以下，文字列の長さは10進数で表記し，その値に下線を付けて表す）。

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | H | e | l | l | o | ! |

(2)　メモの格納と管理のために，２個の配列 Memo[]，Data[]と，４個の変数 MemoCnt，MemoMax，DataLen，DataMax を使用する。

各メモは，配列 Data[] の先頭から順に，１要素に１バイトずつ (1) で示した形式で格納し，その格納位置の情報を配列 Memo[] に設定して管理する。

MemoMax は格納できるメモの最大件数（配列 Memo[] の要素数），MemoCnt は現在格納されているメモの件数である。

DataMax は格納できる最大文字数（配列 Data[] の要素数），DataLen は現在格納されている文字数である（文字数には，文字列の長さの情報を含む）。

(3) 簡易メモ帳の画面には，配列 Memo[] の要素番号の昇順に，それが指すメモを取り出して，メモを表示する。図 1 の表示例は，図 3（後出）の状態に対応している。

(4) メモの編集処理を行うための関数の概要は，次の ① ～ ⑤ のとおりである。これらの関数が呼ばれるとき，引数の内容や配列の空き状態などは事前に検査済みで，正しく実行できるものとする。

なお，以降の図に示す実行例では，MemoMax = 5, DataMax = 25 としている。

① 関数: resetMemo()

全てのメモを消去する。MemoCnt と DataLen に 0 を設定することによって，Memo[] と Data[] の全要素を“空き”の状態にする。

resetMemo() を実行した後の配列・変数の状態を，図 2 に示す。以降の図で，網掛け部分 ▒ は，その配列要素が“空き”であることを表す。

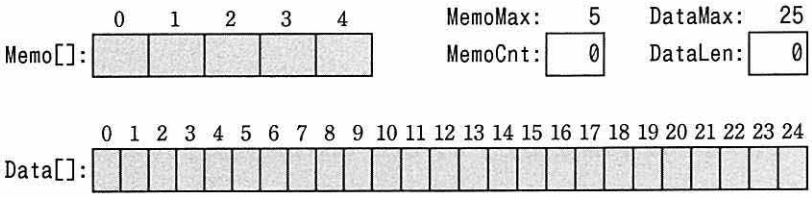



図 2　関数 resetMemo 実行後の配列・変数の状態

② 関数: addMemo(整数型: textLen, 文字列型: text)

1 件のメモを追加する。長さ textLen の文字列 text を Data[] の最初の空き要素以降に格納し，その格納位置の情報を Memo[] に設定する。図 2 の状態から，

```
addMemo(4, "Aoki")
addMemo(4, "Imai")
addMemo(3, "Uno")
addMemo(4, "Endo")
```

をこの順に実行した後の配列・変数の状態を，図 3 に示す。

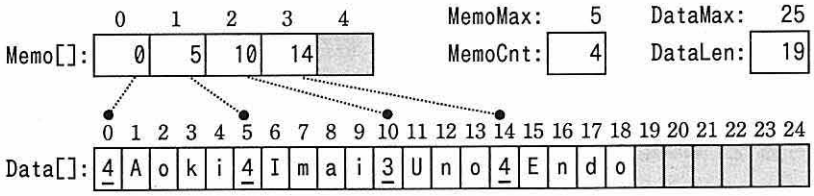



図 3　関数 addMemo（4 件）実行後の配列・変数の状態

E 後

③　関数: deleteMemo(整数型: pos)

1 件のメモを削除する。Memo[] の要素番号 pos＋1 以降の内容をそれぞれ一つ前の要素に移し，MemoCnt から 1 を減じることによって，Memo[pos] が指すメモを削除する（表示の対象から除く）。Data[] 中の参照されなくなったメモは，そのまま残す。図 3 の状態から，deleteMemo(0) を実行した後の配列・変数の状態を，図 4 に示す。以降の図で，斜線部分 ▨ は，参照されなくなったメモであることを表す。

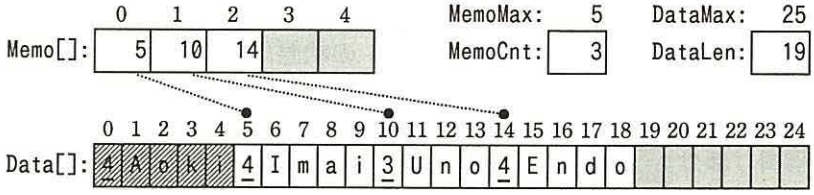



図 4　関数 deleteMemo 実行後の配列・変数の状態

④　関数: changeMemo(整数型: pos, 整数型: textLen, 文字列型: text)

1 件のメモの内容を変更する。長さ textLen の文字列 text を Data[] の最初の空き要素以降に格納し，その格納位置の情報を Memo[pos] に設定することによって，Memo[pos] が指すメモの内容を変更する。Data[] 中の参照されなくなったメモは，そのまま残す。図 4 の状態から changeMemo(2, 3, "Abe") を実行した後の配列・変数の状態を，図 5 に示す。

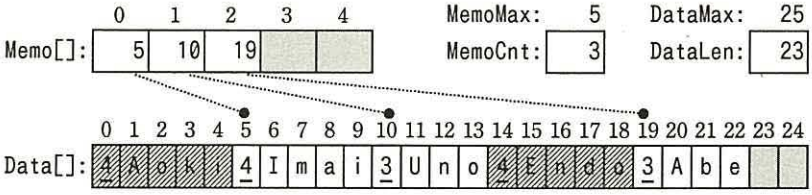



図 5　関数 changeMemo 実行後の配列・変数の状態

⑤　関数: moveMemo(整数型: fromPos, 整数型: toPos)

1 件のメモを移動する。Memo[] の要素の並び順を変えて，Memo[fromPos] の内容を Memo[toPos] の位置に移動する。fromPos ＜ toPos の場合は，Memo[fromPos] の値を取り出し，Memo[fromPos＋1] ～ Memo[toPos] の内容を前方に 1 要素分ずらし，取り出した値を Memo[toPos] に設定するという操作を行

う。fromPos ＞ toPos の場合も，これと同様の操作を行う。図 5 の状態から moveMemo(2, 0) を実行した後の配列・変数の状態を，図 6 に示す。

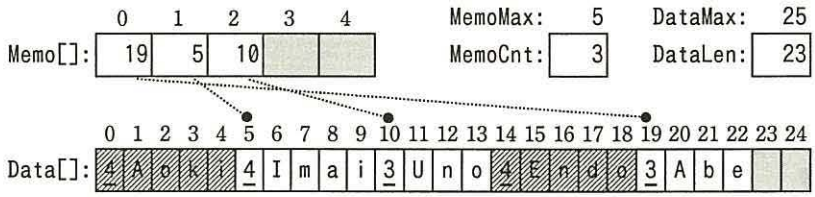



図 6　関数 moveMemo 実行後の配列・変数の状態

〔プログラム〕

○大域　整数型: MemoCnt, MemoMax, Memo[MemoMax]
○大域　整数型: DataLen, DataMax
○大域　8 ビット論理型:　Data[DataMax]

○関数: resetMemo()

- MemoCnt ← 0
- DataLen ← 0

○関数: addMemo(整数型: textLen, 文字列型: text)
　○整数型: i

- Memo[MemoCnt] ← [　　a　　]
- MemoCnt ← MemoCnt ＋ 1
- Data[DataLen] ← textLen
- DataLen ← DataLen ＋ 1
- i: 0, i ＜ textLen, 1
  - Data[DataLen ＋ i] ← text[i]
- DataLen ← [　　b　　]

○関数: deleteMemo(整数型: pos)
　○整数型: i

- i ← [　　c　　]
- i ＜ MemoCnt
  - Memo[i － 1] ← Memo[i]
  - i ← i ＋ 1
- MemoCnt ← MemoCnt － 1

○関数: changeMemo(整数型: pos, 整数型: textLen, 文字列型: text)
　○整数型: i

```
・Memo[pos] ← DataLen
・Data[DataLen] ← textLen
・DataLen ← DataLen + 1
■ i: 0, i < textLen, 1
  ・Data[DataLen + i] ← text[i]
■
・DataLen ← [ b ]
```

○関数: moveMemo(整数型: fromPos, 整数型: toPos)
　○整数型: i, m

```
・m ← Memo[fromPos]
▲ fromPos < toPos
  ■ i: fromPos, i ≦ toPos − 1, 1
    ・Memo[i] ← Memo[i + 1]
  ■
▼
▲ fromPos > toPos
  ■ i: [ d ]
    ・Memo[i] ← Memo[i − 1]
  ■
▼
・Memo[toPos] ← m
```

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　DataLen | イ　DataLen + 1 |
| ウ　DataLen + textLen | エ　DataLen + textLen + 1 |
| オ　textLen | カ　textLen + 1 |

cに関する解答群

| | |
|---|---|
| ア　MemoCnt − pos | イ　pos − 1 |
| ウ　pos | エ　pos + 1 |

dに関する解答群

| | |
|---|---|
| ア　fromPos, i ≧ toPos − 1, −1 | イ　fromPos, i ≧ toPos + 1, −1 |
| ウ　toPos, i ≦ fromPos − 1, 1 | エ　toPos, i ≦ fromPos + 1, 1 |

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。

このメモの管理方法では，削除されたメモや変更前のメモは，Data[] 中に参照されない状態で残っている。その結果，DataLen の値は一方的に増加し，やがて Data[] 中の空き要素が枯渇する。次に示す関数 clearGarbage() は，Data[] 中の参照されなくなったメモを取り除き，空き要素を増やすための関数である。

```
○関数: clearGarbage()
  ○整数型： d, i, m
  ○８ビット論理型： temp[DataMax]

  ・DataLen ← 0
  MemoCnt = 0
  ・return

  m: 0, m < MemoCnt, 1
  ・d ← Memo[m]
  ・Memo[m] ← DataLen
    i: 0, i ≦ Data[d], 1
    ・temp[DataLen] ← Data[d + i]
    ・DataLen ← DataLen + 1

  d: 0, d < DataLen, 1
  ・Data[d] ← temp[d]
```

プログラムの配列・変数が図 6 に示す状態のときに，clearGarbage() を実行すると，実行が終了した時点で，Memo[1] の値は［　e　］，Memo[2] の値は［　f　］，DataLen の値は［　g　］となる。

解答群

| | | | | |
|---|---|---|---|---|
| ア　0 | イ　3 | ウ　4 | エ　5 | オ　7 |
| カ　9 | キ　10 | ク　12 | ケ　13 | コ　23 |

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1～3に答えよ。

〔プログラムの説明〕

図形の一部を拡大すると，再び同じパターンの図形が現れる自己相似性をもつ図形を，フラクタル図形と呼ぶ。関数print_fracは，文字“*”及び空白文字を二次元の格子状に並べてフラクタル図形を描画するプログラムである。

(1)　関数print_fracが描画するフラクタル図形の例を，図1に示す。

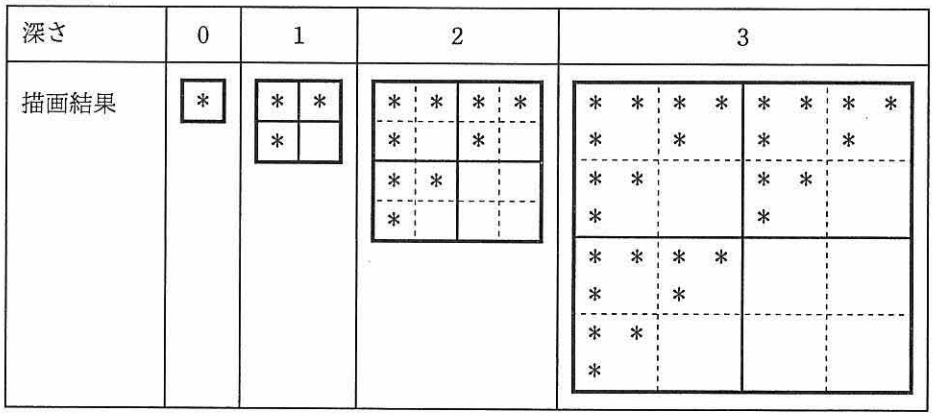



図1　関数print_fracが描画するフラクタル図形の例

①　深さが0の図形は，1行1列の文字“*”から成る図形である。

②　深さが1以上の図形は，深さが0の図形に対して，(2)で説明する生成規則を，深さの回数だけ繰返し適用して得られる図形である。

(2)　深さがd（1以上）の図形は，深さがd－1の図形を構成する一つ一つの文字を，文字“*”であるか空白文字であるかに応じて，図2に示す生成規則のとおりに，文字の行列で置換したものである。

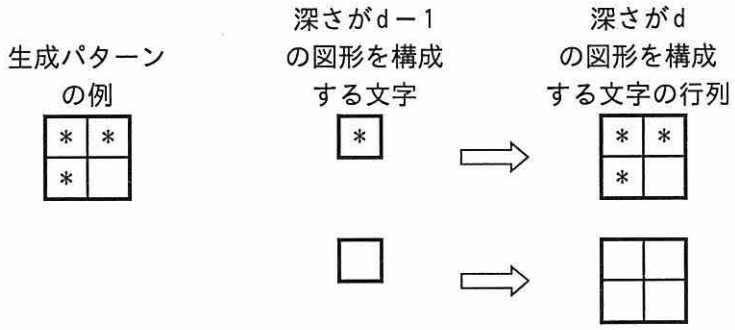



図2　フラクタル図形の生成規則

① 文字“*”の部分は，生成パターンと呼ぶ文字の行列で置換する。

② 空白文字の部分は，生成パターンと同じ大きさで全ての要素が空白文字の行列で置換する。

(3) 生成パターンは，二次元の配列変数 pat によって与える。pat の各要素の値は，空白文字を表す 0，又は文字“*”を表す 1 である。pat の行数，列数及び内容を変更することで，異なるフラクタル図形を描画することができる。

(4) 関数print_fracの仕様は次のとおりである。

機能：　深さが d のフラクタル図形を描画する。

引数：　d　フラクタル図形の深さ

(5) 関数print_fracで使用している関数exists_atの仕様は次のとおりである。

機能：　深さが d のフラクタル図形の i 行 j 列目が空白文字であるか文字“*”であるかを判定する。

引数：　i　行数（一番上の行を0行目とする）

　　j　列数（一番左の列を0列目とする）

　　d　フラクタル図形の深さ

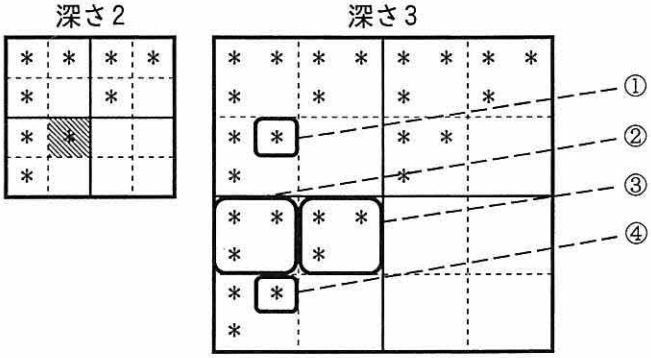
返却値：　判定結果（0：空白文字，1：文字“*”）

ここで，関数の引数に誤りはないものとする。

〔プログラム〕

```
#include <stdio.h>

int pat[2][2] = {
    { 1, 1 },
    { 1, 0 }
};
int p_rn = sizeof pat / sizeof pat[0];
int p_cn = sizeof pat[0] / sizeof pat[0][0];

void print_frac(int);
int exists_at(int, int, int);

void print_frac(int d) {
    int i, j, rn, cn;

    rn = cn = 1;
    for (i = 0; i < d; i++) {
        [ a ]
    }

    for (i = 0; i < rn; i++) {
        for (j = 0; j < cn; j++) {
            putchar(exists_at(i, j, d) ? '*' : ' ');
        }
        putchar('\n');
    }
}

int exists_at(int i, int j, int d) {
    if (d == 0) {
        return [ b ];
    } else if (exists_at(i / p_rn, j / p_cn, d - 1) == 0) {
        return [ c ];
    } else {
        return [ d ];
    }
}
```

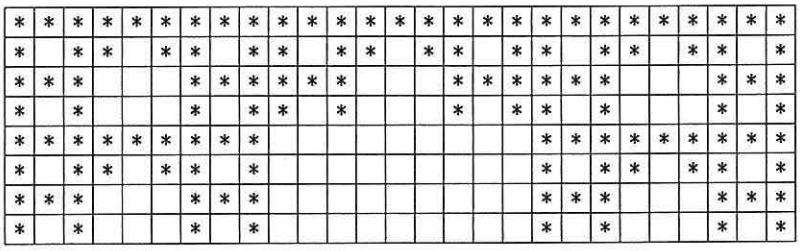
**設問１**　深さが２の図形と深さが３の図形は次のとおりである。深さが３の図形において，深さが２の図形の斜線部を置換した部分として正しい答えを，解答群の中から選べ。




解答群

ア　①　　イ　②　　ウ　③　　エ　④

**設問２**　配列変数 pat を変更して，深さが３の図形を描画したところ，次のとおりになった。配列変数 pat の変更内容として正しい答えを，解答群の中から選べ。



解答群

ア
```
int pat[2][3] = {
    { 0, 1, 0 },
    { 1, 0, 1 }
};
```

イ
```
int pat[2][3] = {
    { 1, 1, 1 },
    { 1, 0, 1 }
};
```

ウ
```
int pat[3][3] = {
    { 0, 1, 0 },
    { 1, 1, 1 },
    { 1, 0, 1 }
};
```

エ
```
int pat[3][3] = {
    { 1, 1, 1 },
    { 1, 0, 1 },
    { 1, 0, 1 }
};
```

設問３　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
```
p_rn += rn;
p_cn += cn;
```

イ
```
p_rn *= rn;
p_cn *= cn;
```

ウ
```
rn += p_rn;
cn += p_cn;
```

エ
```
rn *= p_rn;
cn *= p_cn;
```

b～dに関する解答群

ア　0

イ　1

ウ　pat[i][j]

エ　pat[i % d][j % d]

オ　pat[i / d][j / d]

カ　pat[i % p_rn][j % p_cn]

キ　pat[i / p_rn][j / p_cn]

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〕

Z社では，従業員向けの福利厚生施設として，6部屋の客室を有する保養所を運営している。1部屋には4人まで宿泊可能で，宿泊希望者は1～6までの部屋数を指定できる。宿泊日数は1泊だけである。このプログラムは宿泊予約を管理するサブプログラムであり，予約情報を受け取り，予約結果を呼出し元へ返す。

(1)　このプログラムのパラメタは二つで，各パラメタの様式は，図1のとおりである。

予約情報

| 予約番号<br>6桁 | 希望日<br>8桁 | 部屋数<br>1桁 |
|---|---|---|

予約結果

| 結果<br>1桁 | 空き部屋数<br>1桁 |
|---|---|

図1　パラメタの様式

①　予約番号には，予約希望に対して一意に割り振られた000001～999999の番号が格納されている。

②　希望日には，宿泊する日の年，月，日が，それぞれ西暦の4桁，2桁，2桁で格納されている。

③　部屋数には，希望する部屋数が格納されている。

④　結果には，希望どおりに予約できた場合は0を，できなかった場合は9を設定する。

⑤　空き部屋数には，希望どおりに予約できなかった場合に，希望日の空き部屋数を設定する。例えば，3部屋を希望している場合に2部屋しか空きがなかったときは，2を設定する。希望どおりに予約できた場合は，0を設定する。

⑥　予約情報に誤りはないものとする。

(2) 予約ファイルは，日付を主キーとする図2に示すレコード様式の索引ファイルであり，予約状況を管理する。

| 日付<br>8桁 | 予約状況 | | | | | |
|---|---|---|---|---|---|---|
| | 部屋1<br>6桁 | 部屋2<br>6桁 | 部屋3<br>6桁 | 部屋4<br>6桁 | 部屋5<br>6桁 | 部屋6<br>6桁 |

図2　予約ファイルのレコード様式

① 日付には，年，月，日が，それぞれ西暦の4桁，2桁，2桁で格納されている。

② 部屋1～6のうち，予約済みの部屋には予約番号が格納される。予約が入っていない部屋には0が格納されている。

③ 予約が1部屋も入っていない日のレコードは存在しない。

COBOL

〔プログラム〕

（行番号）

```
 1  DATA  DIVISION.
 2  FILE SECTION.
 3  FD RSV-FILE.
 4  01 RSV-REC.
 5     02 RSV-DATE        PIC X(8).
 6     02 RSV-ROOM        OCCURS 6.
 7        03 RSV-NO          PIC 9(6).
 8  WORKING-STORAGE SECTION.
 9  77 CNT           PIC 9(1).
10  77 GET-ROOM      PIC 9(1).
11  LINKAGE SECTION.
12  01 PRM1.
13     02 PRM-NO          PIC 9(6).
14     02 PRM-DATE        PIC 9(8).
15     02 PRM-ROOM        PIC 9(1).
16  01 PRM2.
17     02 PRM-RSLT        PIC 9(1).
18     02 PRM-FREE        PIC 9(1).
19  PROCEDURE DIVISION USING PRM1 PRM2.
20  MAIN-PROC.
21      OPEN I-O RSV-FILE.
22      MOVE PRM-DATE TO RSV-DATE.
23      READ RSV-FILE INVALID KEY     PERFORM RSV-PROC
```

```
24                      NOT INVALID KEY PERFORM CHK-PROC
25        END-READ.
26        CLOSE RSV-FILE.
27        EXIT PROGRAM.
28    RSV-PROC.
29        INITIALIZE RSV-REC.
30        MOVE PRM-DATE TO RSV-DATE.
31        PERFORM VARYING CNT FROM 1 BY 1 UNTIL CNT > PRM-ROOM
32            [          a          ]
33        END-PERFORM.
34        WRITE RSV-REC.
35        MOVE ZERO TO PRM-RSLT PRM-FREE.
36    CHK-PROC.
37        MOVE ZERO TO GET-ROOM PRM-FREE.
38        PERFORM VARYING CNT FROM 1 BY 1 UNTIL [          b          ]
39           IF RSV-NO(CNT) = ZERO THEN
40              IF PRM-ROOM > GET-ROOM THEN
41                 MOVE PRM-NO TO RSV-NO(CNT)
42                 ADD 1 TO GET-ROOM
43              END-IF
44              ADD 1 TO PRM-FREE
45           END-IF
46        END-PERFORM.
47        IF [          c          ] THEN
48           REWRITE RSV-REC
49           MOVE ZERO TO PRM-RSLT PRM-FREE
50        ELSE
51           MOVE 9 TO PRM-RSLT
52        END-IF.
```

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　ADD 1 TO GET-ROOM

イ　ADD 1 TO PRM-FREE

ウ　MOVE PRM-NO TO RSV-NO(CNT)

エ　MOVE ZERO TO RSV-NO(CNT)

b，c に関する解答群

ア　CNT > 6　　イ　CNT > GET-ROOM

ウ　CNT > PRM-ROOM　　エ　GET-ROOM < PRM-ROOM

オ　GET-ROOM = PRM-ROOM　　カ　GET-ROOM > PRM-ROOM

設問2　予約をキャンセルする場合は，宿泊する日の7日前からキャンセル料が発生する。希望どおり予約できた場合には，キャンセル料の発生開始日もパラメタで返すように，プログラムを変更する。キャンセル料の発生開始日の年，月，日は，それぞれ西暦の4桁，2桁，2桁で表し，予約できなかった場合には0を設定する。キャンセル料の発生開始日を求める処理は 2016 年～2099 年で動作を保証する。表中の［　　］に入れる正しい答えを，解答群の中から選べ。

COBOL

表1　プログラムの変更内容

| 処置 | 変更内容 |
|---|---|
| 行番号10と11の間に追加 | 01 MONTH-TABLE.<br>02 MONTH-VAL PIC X(24) VALUE "312831303130313130313031".<br>02 MONTH-ELM REDEFINES MONTH-VAL OCCURS 12 PIC 9(2).<br>77 URU-SUPPLY PIC 9(1).<br>77 QUOTIENT PIC 9(3). |
| 行番号18と19の間に追加 | 02 PRM-CANCEL.<br>03 CAN-YYYY PIC 9(4).<br>03 CAN-MM PIC 9(2).<br>03 CAN-DD PIC 9(2). |
| ［　d　］に追加 | IF PRM-RSLT = ZERO THEN<br>PERFORM SET-CANCEL<br>ELSE<br>MOVE ZERO TO PRM-CANCEL<br>END-IF. |
| 行番号52の後ろに追加 | SET-CANCEL.<br>MOVE PRM-DATE TO PRM-CANCEL.<br>IF CAN-DD > 7 THEN<br>SUBTRACT 7 FROM CAN-DD<br>ELSE<br>IF CAN-MM = 1 THEN<br>SUBTRACT 1 FROM CAN-YYYY<br>MOVE 12 TO CAN-MM<br>COMPUTE CAN-DD = 31 + CAN-DD - 7 |

```
        ELSE
           SUBTRACT 1 FROM CAN-MM
           [          e          ]
           IF CAN-MM = 2 THEN
              DIVIDE CAN-YYYY BY 4
                     GIVING QUOTIENT REMAINDER URU-SUPPLY
              IF [          f          ] THEN
                 ADD 1 TO CAN-DD
              END-IF
           END-IF
        END-IF
     END-IF.
```

dに関する解答群

ア　行番号25と26の間　　イ　行番号35と36の間

ウ　行番号37と38の間　　エ　行番号46と47の間

eに関する解答群

ア　COMPUTE CAN-DD = 30 + CAN-DD - 7

イ　COMPUTE CAN-DD = MONTH-ELM(CAN-MM - 1) - 7

ウ　COMPUTE CAN-DD = MONTH-ELM(CAN-MM) + CAN-DD - 7

エ　SUBTRACT 7 FROM CAN-DD

fに関する解答群

ア　QUOTIENT > 100

イ　QUOTIENT > 400

ウ　URU-SUPPLY = ZERO

エ　URU-SUPPLY NOT = ZERO

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〕

“すべきこと”（以下，ToDo という）を管理するプログラムである。

(1)　クラス ToDo は ToDo を表すクラスであり，コンストラクタで主題，期限，重要度を指定する。期限は，年月日を表す 8 桁又は年月日時分を表す 12 桁の数字から成る文字列（以下，日時という）であり，例えば，2016 年 4 月 16 日を表す文字列は“20160416”，2016 年 4 月 16 日午後 1 時 0 分を表す文字列は“201604161300”である。ここで，日時に誤りはないものとする。

主題，期限，重要度を取得する各メソッドと，状態を設定及び取得するメソッド，ToDo を識別するフィールド id をもつ。

列挙 Priority は ToDo の重要度を表す列挙であり，重要度が低い順に LOW，MIDDLE，HIGH である。

列挙 State は ToDo の状態を表す列挙であり，NOT_YET_STARTED は未着手，STARTED は着手済み，DONE は完了を表す。

(2)　クラス ToDoList は ToDo のリストを保持するクラスである。

リスト中に，フィールド id の値が同じ ToDo を複数個含まないことを保証する。

ToDo を追加するメソッド add と，ToDo の更新を行うメソッド update，条件に合う ToDo のリストを返すメソッド select をもつ。

メソッド add の引数に，既にリストに保持されている ToDo を指定したとき，及びメソッド update の引数に，リストにない ToDo を指定したときは何もしない。

メソッド select の引数には，条件を 0 個以上指定できる。条件を指定したときは，全ての条件に合致する ToDo から成るリストを返す。条件を指定しないときは，保持する全ての ToDo から成るリストを返す。

(3)　インタフェース Condition は，ToDo を選択する際の条件を表すクラスが実装するインタフェースである。メソッド test は条件に合致するときに true を返す。

(4)　クラス ToDoListTester は，テスト用のクラスである。

Java

F
午

〔プログラム1〕

```
import java.util.UUID;

public class ToDo {
   public enum Priority { LOW, MIDDLE, HIGH }
   public enum State { NOT_YET_STARTED, STARTED, DONE }

   // 8桁又は12桁の数字から成る文字列と一致する正規表現
   private static final String DEADLINE_PATTERN = "\\d{8}|\\d{12}";

   private final String id;
   private String subject;
   private String deadline;
   private Priority priority;
   private State state;

   private ToDo(String subject, String deadline, Priority priority,
                String id, State state) {
      if (!deadline.matches(DEADLINE_PATTERN)) {
         throw new IllegalArgumentException();
      }
      this.subject = subject;
      this.deadline = deadline;
      this.priority = priority;
      this.id = id;
      this.state = state;
   }

   public ToDo(String subject, String deadline, Priority priority) {
      this(subject, deadline, priority,
           UUID.randomUUID().toString(), State.NOT_YET_STARTED);
   }

   public ToDo(ToDo todo) {
      this(todo.subject, todo.deadline, todo.priority, todo.id, todo.state);
   }

   public String getSubject() { return subject; }
   public String getDeadline() { return deadline; }
   public Priority getPriority() { return priority; }
   public State getState() { return state; }
   public void setState(State state) { this.state = state; }
   public int hashCode() { return id.hashCode(); }
```

Java

```
    public boolean equals(Object o) {
        return o instanceof ToDo && [   a   ];
    }

    public String toString() {
        return String.format("主題: %s, 期限: %s, 優先度: %s, 状態: %s",
                             subject, deadline, priority, state);
    }
}
```

〔プログラム 2〕

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

public class ToDoList {
    private List<ToDo> todoList = new ArrayList<ToDo>();

    public void add(ToDo todo) {
        if ([   b   ]) {
            todoList.add(new ToDo(todo));
        }
    }

    public void update(ToDo todo) {
        int index = todoList.indexOf(todo);
        if (index [   c   ]) {
            todoList.set(index, todo);
        }
    }

    public List<ToDo> select(Condition... conditions) {
        List<ToDo> result = new ArrayList<ToDo>();
        for (ToDo todo : todoList) {
            [   d   ];
            for (Condition condition : conditions) {
                selected [   e   ] condition.test(todo);
                if (!selected) break;
            }
            if (selected) {
                result.add(new ToDo(todo));
            }
        }
```

Java

```
        return result;
    }
}
```

〔プログラム3〕

```
public interface Condition {
    boolean test(ToDo todo);
}
```

〔プログラム4〕

```
public class ToDoListTester {
    public static void main(String[] args) {
        ToDoList list = new ToDoList();
        list.add(new ToDo("メール送信", "201604181500", ToDo.Priority.HIGH));
        list.add(new ToDo("ホテル予約", "20160420", ToDo.Priority.LOW));
        list.add(new ToDo("チケット購入", "20160430", ToDo.Priority.MIDDLE));
        list.add(new ToDo("報告書作成", "20160428", ToDo.Priority.HIGH));
        list.add(new ToDo("会議室予約", "201605301200", ToDo.Priority.HIGH));
        list.update(new ToDo("PC購入", "20160531", ToDo.Priority.HIGH));
        for (ToDo todo : list.select()) {
            todo.setState(ToDo.State.STARTED);
            list.update(todo);
        }
        Condition condition1 = new Condition() {
            public boolean test(ToDo todo) {
                return todo.getDeadline().compareTo("20160501") < 0;
            }
        };
        Condition condition2 = new Condition() {
            public boolean test(ToDo todo) {
                return todo.getPriority().equals(ToDo.Priority.HIGH);
            }
        };
        for (ToDo todo : list.select(condition1, condition2)) {
            System.out.println(todo);
        }
    }
}
```

Java

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　((ToDo) o).id.equals(id)　　イ　(ToDo) o.id.equals(id)

ウ　id.equals(id)　　エ　o.id.equals(id)

bに関する解答群

ア　!todoList.contains(todo)　　イ　!todoList.isEmpty()

ウ　todoList.contains(todo)　　エ　todoList.isEmpty()

cに関する解答群

ア　!= -1　　イ　< todoList.size()

ウ　== -1　　エ　>= todoList.size()

dに関する解答群

ア　boolean selected = false　　イ　boolean selected = true

ウ　int selected = 0　　エ　int selected = todoList.size()

eに関する解答群

ア　+=　　イ　=　　ウ　==　　エ　|=

Java

設問2　プログラム 4 の実行結果を図 1 に示す。[　　]に入れる正しい答えを，解答群の中から選べ。

```
[ f ], 状態: STARTED
[ g ], 状態: STARTED
```

図 1　プログラム 4 の実行結果

解答群

ア　主題: PC 購入, 期限: 20160531, 優先度: HIGH

イ　主題: 会議室予約, 期限: 201605301200, 優先度: HIGH

ウ　主題: チケット購入, 期限: 20160430, 優先度: MIDDLE

エ　主題: 報告書作成, 期限: 20160428, 優先度: HIGH

オ　主題: ホテル予約, 期限: 20160420, 優先度: LOW

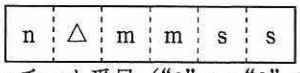
カ　主題: メール送信, 期限: 201604181500, 優先度: HIGH

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 1～4 に答えよ。

〔プログラムの説明〕

10,000 m 競走に，1 チーム 8 名で，9 チーム 72 名が参加し，各チームの合計タイムを競う。プログラム RACE は，個人別のタイムを所属するチーム番号とともに入力し，チーム別合計タイムを求め，チーム別合計タイムの昇順に，チーム番号とそのチームの 1 名当たりの平均タイムを出力する。

(1)　個人ごとの入力データ及びチームごとの出力データの形式は図 1 のとおりとする。ここで，60 分以上のデータはないものとする。




n：チーム番号（“1”～“9”の数字）
△：空白文字
mm：分（“00”～“59”の数字列）
ss：秒（“00”～“59”の数字列）

図 1　入出力データの形式

(2)　プログラム RACE の入力データと出力結果の例を図 2 に示す。

図 2　入力データと出力結果の例

(3)　データの入力処理と出力処理は，それぞれ副プログラム INPUT と OUTPUT を呼び出して行う。

(4)　副プログラム INPUT は，呼ばれるごとに図 1 に示す形式の個人のデータを 1 件読み込む。読み込んだデータのチーム番号及び秒数に変換したタイムを 2 進数で，それぞれ GR1 と GR2 に設定し，GR0 には正の値を設定して，呼出し元に返す。

全員のデータを読み終わっている場合は，GR0 に－1 を設定して呼出し元に返す。

(5) 副プログラム OUTPUT は，GR1 と GR2 のそれぞれに 2 進数で設定されたチーム番号と平均タイム（1 秒未満を切り捨てた秒数）を，図 1 に示す形式に変換して書き出す。

〔プログラム 1〕

```
RACE     START
LOOP11   CALL  INPUT
         LD    GR0,GR0
         JMI   IEND
         ADDA  GR2,TOTAL,GR1     ; タイム
         ST    GR2,TOTAL,GR1
         JUMP  LOOP11
IEND     LD    GR3,=9
         LD    GR0,=#7FFF        ; 最大値
LOOP12   LD    GR2,GR0
         LD    GR4,=9
LOOP13   CPA   GR2,TOTAL,GR4
         JPL   CHANGE
         JUMP  NEXT
CHANGE   LD    GR2,TOTAL,GR4     ; GR2←新たなタイムの最小値
         LD    GR1,GR4           ; GR1←そのチーム番号
NEXT     SUBA  GR4,=1
         [        a        ]
         SRL   GR2,3             ; 平均タイム
         CALL  OUTPUT
         ST    GR0,TOTAL,GR1     ; 処理済みとして最大値を設定
         SUBA  GR3,=1
         [        b        ]
         RET
TOTAL    DC    0,0,0,0,0,0,0,0,0,0      ; （チーム数＋1）個の領域
         END
```

**設問 1**　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。

解答群

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ア | JMI | LOOP12 | イ | JMI | LOOP13 | ウ | JPL | LOOP12 | エ | JPL | LOOP13 |
| オ | JUMP | LOOP12 | カ | JUMP | LOOP13 | キ | JZE | LOOP12 | ク | JZE | LOOP13 |

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。

同じ合計タイムのチームが複数ある場合，［　c　］。

解答群

ア　チーム番号の昇順に出力される

イ　チーム番号の降順に出力される

ウ　どのチームが先に出力されるかは時々で異なる

設問３　副プログラム OUTPUT を次に示す。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〕

（行番号）

```
 1  OUTPUT   START
 2           RPUSH
 3           OR     GR1,=#0030
 4           ST     GR1,TEAM
 5           LAD    GR3,TIME
 6           LD     GR4,=0
 7           LD     GR5,=3              ; ループ回数
 8  LOOP21   LD     GR6,DIVISOR,GR4     ; GR6 ← 除数
 9           LD     GR7,=0              ; GR7 ← 商
10  LOOP22   CPA    GR2,GR6
11           JMI    SETTIME
12           SUBA   GR2,GR6
13           ADDA   GR7,=1
14           [          d          ]
15  SETTIME  OR     GR7,=#0030
16           ST     GR7,0,GR3
17           ADDA   GR4,=1
18           ADDA   GR3,=1
19           SUBA   GR5,=1
20           [          e          ]
21           OR     GR2,=#0030
22           ST     GR2,0,GR3
23           OUT    TEAM,LEN
```

```
24            RPOP
25            RET
26  TEAM      DS    1
27            DC    ' '
28  TIME      DS    4
29  LEN       DC    6
30  DIVISOR   DC    600,60,10
31            END
```

dに関する解答群

ア JMI LOOP21　イ JMI LOOP22　ウ JPL LOOP21　エ JUMP LOOP21

オ JUMP LOOP22　カ JZE LOOP22

eに関する解答群

ア JMI LOOP21　イ JPL LOOP21　ウ JPL LOOP22　エ JUMP LOOP22

オ JZE LOOP21　カ JZE LOOP22

**設問4**　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。

出力されたデータが次のとき，プログラム 2 の行番号 12 の SUBA 命令は［　f　］回実行された。

| 7 | △ | 2 | 7 | 0 | 8 |
|---|---|---|---|---|---|

解答群

ア 9　イ 17　ウ 20　エ 270　オ 299

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〕

卸売業の P 社は，家庭用品部門で販売している商品の取扱いを見直すことにした。そこで，販売管理システムから対象商品の連続した過去 3 年間の販売データを抽出し，表計算ソフトを利用して販売状況の分析を行い，今後の取扱いを検討することにした。対象商品は，商品コード A001，A002，B101，B201，C010，C011，C012，D070 及び D080 の 9 商品である。販売データは，各年の 1 月 1 日から 12 月 31 日までの 1 年を単位として分析し，分析する対象の各年で売上高が 0 の商品はない。

〔ワークシート：販売分析〕

対象商品の販売状況を分析する指標を計算して，今後の取扱いを判定するワークシート“販売分析”を作成した。ワークシート“販売分析”の例を図 1 に示す。

(1)　ワークシートのセル A1 ～ L21 は“売上高分析表”である。“売上高分析表”では，対象商品の年ごとの売上高を基にして，販売状況を分析する指標を求めている。

①　セル C2 ～ K2 には，対象商品の商品コードが昇順に格納されている。

②　マクロ Sales_Total を実行すると，セル A3，セル A8 及びセル A13 には，分析する過去 3 年の西暦年が昇順に格納され，セル C3 ～ K3，セル C8 ～ K8 及びセル C13 ～ K13 には，対象商品の当該年の売上高が，千円未満を四捨五入した千円単位の値で格納される。

③　対前年比率は，各商品の前年の売上高を基準にした当該年の売上高の割合である。

④　構成比率は，当該年の対象 9 商品の売上高の合計に占める各商品の売上高の割合である。

⑤　構成比率累計は，当該年の各商品の構成比率を，大きいものから順に当該商品まで累計した値である。ここで，当該商品と構成比率の同じ商品が複数あったときは，当該商品の分だけを加算する。

| | A | B | C | D | E | … | J | K | L |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 売上高分析表 | | | | | | | | 単位　千円 |
| 2 | | 商品コード | A001 | A002 | B101 | … | D070 | D080 | 合計 |
| 3 | 2013 | 売上高 | 35,376 | 11,544 | 13,005 | … | 7,480 | 12,310 | 151,847 |
| 4 | | 対前年比率 | － | － | － | … | － | － | － |
| 5 | | 構成比率 | 23.3% | 7.6% | 8.6% | … | 4.9% | 8.1% | |
| 6 | | 構成比率累計 | 23.3% | 81.6% | 65.8% | … | 100.0% | 74.0% | |
| 7 | | 管理区分 | A | B | A | … | C | B | |
| 8 | 2014 | 売上高 | 36,592 | 10,376 | 12,510 | … | 7,110 | 15,230 | 155,194 |
| 9 | | 対前年比率 | 103.4% | 89.9% | 96.2% | … | 95.1% | 123.7% | 102.2% |
| 10 | | 構成比率 | 23.6% | 6.7% | 8.1% | … | 4.6% | 9.8% | |
| 11 | | 構成比率累計 | 23.6% | 90.1% | 76.6% | … | 100.0% | 68.5% | |
| 12 | | 管理区分 | A | C | B | … | C | A | |
| 13 | 2015 | 売上高 | 39,480 | 9,032 | 13,392 | … | 6,170 | 12,830 | 159,524 |
| 14 | | 対前年比率 | 107.9% | 87.0% | 107.1% | … | 86.8% | 84.2% | 102.8% |
| 15 | | 構成比率 | 24.7% | 5.7% | 8.4% | … | 3.9% | 8.0% | |
| 16 | | 構成比率累計 | 24.7% | 90.6% | 70.2% | … | 100.0% | 78.3% | |
| 17 | | 管理区分 | A | C | B | … | C | B | |
| 18 | 判定 | 売上高２年連続減 | N | Y | N | … | Y | N | |
| 19 | | 対前年比 85%未満 | N | N | N | … | N | Y | |
| 20 | | 管理区分２年連続 C | N | Y | N | … | Y | N | |
| 21 | | 取扱方針 | 維持 | 見直し | 維持 | … | 見直し | 注意 | |
| 22 | | | | | | | | | |
| 23 | 管理区分表 | | | | | | | | |
| 24 | 管理区分 | 構成比率累計の範囲 | | | | | | | |
| 25 | A | 0% | を超え | 70% | 以下 | | | | |
| 26 | B | 70% | を超え | 90% | 以下 | | | | |
| 27 | C | 90% | を超え | 100% | 以下 | | | | |
| 28 | | | | | | | | | |
| 29 | 取扱判定表 | | | | | | | | |
| 30 | | 判定コード | 0 | 1 | 2 | … | 7 | | |
| 31 | 条件 | 売上高２年連続減 | N | N | N | … | Y | | |
| 32 | | 対前年比 85%未満 | N | N | Y | … | Y | | |
| 33 | | 管理区分２年連続 C | N | Y | N | … | Y | | |
| 34 | 取扱方針 | | 維持 | 維持 | 注意 | … | 見直し | | |

注記　対前年比率，構成比率及び構成比率累計は，%形式で小数第２位を四捨五入した値を表示している。

図１　ワークシート“販売分析”の例

⑥　管理区分は，各商品の構成比率累計が，表１の構成比率累計の範囲のどの区分に属するかを判定した結果であり，商品の管理方法を決定する指標である。

表１　管理区分の判定基準

| 管理区分 | 構成比率累計の範囲 | 商品の管理方法 |
|---|---|---|
| A | 0%を超え，70%以下 | 重要管理商品 |
| B | 70%を超え，90%以下 | 通常管理商品 |
| C | 90%を超え，100%以下 | 簡易管理商品 |

⑦　セル C18～K20 には，表 2 に示した各条件の真偽を商品ごとに判定した結果を設定する。セル C21～K21 には，各条件の真偽の組合せに基づいて，各商品の今後の取扱方針を設定する。

表２　今後の取扱方針を判定する条件

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 条件 | 直近の２年の売上高が連続して前年より減少 | N | N | N | N | Y | Y | Y | Y |
| | 直近の年の対前年比率が85%未満 | N | N | Y | Y | N | N | Y | Y |
| | 直近の２年の管理区分が連続してC | N | Y | N | Y | N | Y | N | Y |
| 取扱方針 | | 維持 | 維持 | 注意 | 見直し | 注意 | 見直し | 注意 | 見直し |

(a)　“Y”は条件が真，“N”は条件が偽であることを表す。

(b)　取扱方針は，条件の真，偽が各列の組合せに該当する商品の，今後の取扱方針を表す。

(2)　ワークシートのセル A23～E27 は“管理区分表”である。“管理区分表”には，管理区分と対応する構成比率累計の範囲が，表１のとおりに格納されている。

(3)　ワークシートのセル A29～J34 は“取扱判定表”である。

①　セル C31～J34 には取扱方針を判定するための条件の組合せと，その組合せに基づいた取扱方針が，表２のとおりに格納されている。

②　セル C30～J30 には，セル C31～J33 の各列の条件の“Y”を 1，“N”を 0 とし，上から順に２進数の３桁目，２桁目，１桁目として並べたものを 10 進数に変換した値が，判定コードとして格納されている。

設問1　ワークシート“販売分析”の作成に関する記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

(1)　セル L3，セル L8 及びセル L13 には，当該年の対象 9 商品の売上高の合計を求める式を入力する。

(2)　セル C4～L4 には，“－”を入力する。セル C9～L9 及びセル C14～L14 には，対前年比率を求める式を入力する。

(3)　セル C5～K5，セル C10～K10 及びセル C15～K15 には，構成比率を求める式を入力する。

(4)　セル C6 には，構成比率累計を求める次の式を入力し，セル D6～K6，セル C11～K11 及びセル C16～K16 に複写する。

［　　a　　］

(5)　セル C7 には，管理区分を表示する次の式を入力し，セル D7～K7，セル C12～K12 及びセル C17～K17 に複写する。

［　　b　　］

(6)　セル C18 には，直近 2 年の売上高が連続して前年よりも減少しているかどうかを判定する次の式を入力し，セル D18～K18 に複写する。

［　　c　　］

(7)　セル C19～K19 には，当該商品の直近の年の対前年比率が 85％未満かどうかを判定する式を入力する。

(8)　セル C20～K20 には，当該商品の直近 2 年の管理区分が連続して“C”かどうかを判定する式を入力する。

(9)　セル C21 には，“取扱判定表”の条件がセル C18～C20 の値と一致する列に対応する取扱方針を表示する次の式を入力し，セル D21～K21 に複写する。

［　　d　　］

aに関する解答群

ア　条件付合計($C5:$K5, ＞C5, $C5:$K5)

イ　条件付合計($C5:$K5, ＞C5, $C5:$K5)＋C5

ウ　条件付合計($C5:$K5, ≧C5, $C5:$K5)

エ　条件付合計($C5:$K5, ≧C5, $C5:$K5)－C5

bに関する解答群

ア　表引き($A$25:$A$27, 条件付個数($B$25:$B$27, ＜C6), 1)

イ　表引き($A$25:$A$27, 条件付個数($D$25:$D$27, ≦C6), 1)

ウ　表引き($A$25:$A$27, 照合一致(C6, $B$25:$B$27, 1), 1)

エ　表引き($A$25:$A$27, 照合一致(C6, $D$25:$D$27, －1), 1)

cに関する解答群

ア　IF(論理積(C9＜1, C14＜1), 'Y', 'N')

イ　IF(論理積(C9＞1, C14＞1), 'N', 'Y')

ウ　IF(論理和(C9＜1, C14＜1), 'Y', 'N')

エ　IF(論理和(C9＞1, C14＞1), 'N', 'Y')

dに関する解答群

ア　水平照合(IF(C18＝'Y', 3, 0)＋IF(C19＝'Y', 2, 0)＋
　　IF(C20＝'Y', 1, 0), $C30:$J34, 4, 0)

イ　水平照合(IF(C18＝'Y', 3, 0)＋IF(C19＝'Y', 2, 0)＋
　　IF(C20＝'Y', 1, 0), $C30:$J34, 5, 0)

ウ　水平照合(IF(C18＝'Y', 4, 0)＋IF(C19＝'Y', 2, 0)＋
　　IF(C20＝'Y', 1, 0), $C30:$J34, 4, 0)

エ　水平照合(IF(C18＝'Y', 4, 0)＋IF(C19＝'Y', 2, 0)＋
　　IF(C20＝'Y', 1, 0), $C30:$J34, 5, 0)

設問2　図2に示すワークシート“販売データ”のデータを基に，〔ワークシート：販売分析〕の説明の(1)の②で示した処理を実行するマクロ Sales_Total を作成して，ワークシート“販売分析”に格納した。マクロ Sales_Total 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：販売データ〕

ワークシート“販売データ”の例を図2に示す。

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | 年 | 月 | 日 | 商品コード | 得意先コード | 単価 | 数量 |
| 2 | 2013 | 1 | 4 | A001 | 1109 | 800 | 300 |
| 3 | 2013 | 1 | 5 | A001 | 1115 | 800 | 170 |
| 4 | 2013 | 1 | 6 | A001 | 1119 | 800 | 180 |
| 5 | 2013 | 1 | 6 | A001 | 1107 | 800 | 50 |
| 6 | 2013 | 1 | 7 | A001 | 1104 | 800 | 120 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 2292 | 2014 | 12 | 24 | D080 | 1118 | 1,000 | 90 |
| 2293 | 2014 | 12 | 28 | D080 | 1106 | 1,000 | 230 |
| 2294 | 2015 | 1 | 4 | A001 | 1117 | 800 | 220 |
| 2295 | 2015 | 1 | 6 | A001 | 1104 | 800 | 110 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

図2　ワークシート“販売データ”の例

(1)　ワークシート“販売データ”には，販売管理システムから，対象商品の連続した過去3年間のデータを抽出して格納してある。

(2)　行1には，見出しが格納されている。行2以降には，販売した年，月，日，商品コード，得意先コード，単価及び数量のデータが，年を第1キー，商品コードを第2キーとして昇順に格納されている。

(3)　データが格納されていないセルの値は，null である。

〔マクロ：Sales_Total〕

```
○マクロ: Sales_Total
○数値型: data_row, total_row, total_col, subtotal
・data_row ← 0
・total_row ← 0
・subtotal ← 0
■ 相対(販売データ!A2, data_row, 0) ≠ null
  ・相対(A3, total_row, 0) ← 相対(販売データ!A2, data_row, 0)
  ・total_col ← 0
  ■ 相対(A3, total_row, 0) = 相対(販売データ!A2, data_row, 0)
    ■ [ e ]
    ・[ f ]
    ・data_row ← data_row + 1
    ■
    ・相対(C3, total_row, total_col)
        ← 四捨五入(subtotal / 1000, 0)
    ・subtotal ← 0
    ・total_col ← total_col + 1
  ■
  ・[ g ]
■
```

eに関する解答群

ア　相対(C2, 0, total_col) = 相対(販売データ!D2, data_row, 0)

イ　相対(C2, 0, total_col) ≠ 相対(販売データ!B2, data_row, 0)

ウ　相対(C2, 0, total_col) ≠ 相対(販売データ!D2, data_row, 0)

エ　相対(C2, total_row, 0) = 相対(販売データ!B2, data_row, 0)

オ　相対(C2, total_row, 0) = 相対(販売データ!D2, data_row, 0)

カ　相対(C2, total_row, 0) ≠ 相対(販売データ!B2, data_row, 0)

f, gに関する解答群

ア subtotal ← subtotal +

相対(販売データ!A1, data_row, 5) * 相対(販売データ!A1, data_row, 6)

イ subtotal ← subtotal +

相対(販売データ!A2, total_row, 5) * 相対(販売データ!A2, total_row, 6)

ウ subtotal ← subtotal +

相対(販売データ!F2, data_row, 0) * 相対(販売データ!G2, data_row, 0)

エ total_row ← 0

オ total_row ← total_row + 1

カ total_row ← total_row + 5

## ■ Java プログラムで使用する API の説明

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 Iterable を継承する。

メソッド

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

public E set(int index, E e)

リスト内の指定された位置にある要素を，指定された要素に置き換える。

引数： index ― 置き換えられる要素のインデックス

e ― 指定された位置に格納される要素

戻り値：リスト内の指定された位置にあった要素

例外： IndexOutOfBoundsException ― インデックスが 0 未満又は要素数以上のとき

public boolean contains(Object obj)

指定された要素がこのリストにあればtrueを返す。

引数： obj ― 要素

戻り値：指定された要素がこのリストにあれば true

それ以外は false

public boolean isEmpty()

リストに要素がなければ true を返す。

戻り値：リストに要素が一つもなければ true

それ以外は false

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

public int indexOf(Object obj)

指定された要素がリスト内で最初に検出された位置のインデックスを返す。

引数： obj ― 要素

戻り値：指定された要素がリスト内で最初に検出された位置のインデックス

指定された要素がリストになければ −1

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。

コンストラクタ

public ArrayList()

空のリストを作る。

java.lang

public final class String

クラスStringは，文字列を表す。

メソッド

public boolean matches(String regex)

この文字列が指定された正規表現と一致するかどうかを判定する。

例えば，正規表現 "\\d{8}|\\d{12}" は 8 桁又は 12 桁の数字から成る文字列と一致する。

引数： regex ― 正規表現

戻り値：この文字列が指定された正規表現と一致すれば true

それ以外は false

public int compareTo(String str)

この文字列と指定された文字列を辞書的に比較する。

引数： str ― 文字列

戻り値：この文字列と指定された文字列が等しいときは 0

この文字列が指定された文字列より辞書的に小さい場合は 0 より小さい値

この文字列が指定された文字列より辞書的に大きい場合は 0 より大きい値

java.util

public final class UUID

クラスUUIDは，128ビットの値であるユニバーサル固有識別(UUID)を表す。

メソッド

public static UUID randomUUID()

ユニバーサル固有識別をランダムに生成する。

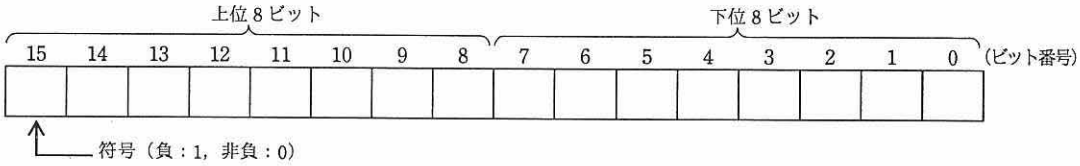
戻り値：ランダムに生成された固有識別

# ■アセンブラ言語の仕様

## 1. システム COMETⅡ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<br>LoaD | LD | r1,r2 | r1 ← (r2) | ○*1 |
| | | r,adr [,x] | r ← (実効アドレス) | |
| ストア<br>STore | ST | r,adr [,x] | 実効アドレス ← (r) | — |
| ロードアドレス<br>Load ADdress | LAD | r,adr [,x] | r ← 実効アドレス | |

(2) 算術，論理演算命令

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

(3) 比較演算命令

| | | | | |
|---|---|---|---|---|
| 算術比較<br>ComPare Arithmetic | CPA | r1,r2 | (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。 | ○*1 |
| | | r,adr [,x] | | |
| 論理比較<br>ComPare Logical | CPL | r1,r2 | | |
| | | r,adr [,x] | | |

| 比較結果 | FR の値 SF | FR の値 ZF |
|---|---|---|
| (r1) ＞ (r2)<br>(r) ＞ (実効アドレス) | 0 | 0 |
| (r1) ＝ (r2)<br>(r) ＝ (実効アドレス) | 0 | 1 |
| (r1) ＜ (r2)<br>(r) ＜ (実効アドレス) | 1 | 0 |

(4) シフト演算命令

| | | | | |
|---|---|---|---|---|
| 算術左シフト<br>Shift Left Arithmetic | SLA | r,adr [,x] | 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。<br>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。 | ○*2 |
| 算術右シフト<br>Shift Right Arithmetic | SRA | r,adr [,x] | | |
| 論理左シフト<br>Shift Left Logical | SLL | r,adr [,x] | 符号を含み (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。<br>シフトの結果，空いたビット位置には 0 が入る。 | |
| 論理右シフト<br>Shift Right Logical | SRL | r,adr [,x] | | |

(5) 分岐命令

| | | | | |
|---|---|---|---|---|
| 正分岐<br>Jump on PLus | JPL | adr [,x] | 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。 | — |
| 負分岐<br>Jump on MInus | JMI | adr [,x] | | |
| 非零分岐<br>Jump on Non Zero | JNZ | adr [,x] | | |
| 零分岐<br>Jump on ZEro | JZE | adr [,x] | | |
| オーバフロー分岐<br>Jump on OVerflow | JOV | adr [,x] | | |
| 無条件分岐<br>unconditional JUMP | JUMP | adr [,x] | 無条件に実効アドレスに分岐する。 | |

| 命令 | 分岐するときの FR の値 OF | SF | ZF |
|---|---|---|---|
| JPL | | 0 | 0 |
| JMI | | 1 | |
| JNZ | | | 0 |
| JZE | | | 1 |
| JOV | 1 | | |

(6) スタック操作命令

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

(7) コール，リターン命令

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

(8) その他

| | | | |
|---|---|---|---|
| スーパバイザコール<br>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<br>No OPeration | NOP | 何もしない。 | |

注記　r，r1，r2　いずれも GR を示す。指定できる GR は GR0 ～ GR7
adr　アドレスを示す。指定できる値の範囲は 0 ～ 65535
x　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。
実効アドレス　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。
$+_L$，$-_L$　論理加算，論理減算を示す。
FR の設定　○　：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。

| 行＼列 | 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 | - | = | M | ] | m | } |
| 14 | . | > | N | ^ | n | ~ |
| 15 | / | ? | O | _ | o | |

## 2. アセンブラ言語 CASLⅡの仕様

### 2.1 言語の仕様

(1) CASLⅡは，COMETⅡ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（COMETⅡ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<br>プログラムの実行開始番地を定義<br>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−32768～32767 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 ≦ h ≦ 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。<br>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)

| RPUSH | |
|---|---|

RPUSH 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)

| RPOP | |
|---|---|

RPOP 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。<br>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。CASLⅡ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連係処理を行いアドレスを決定する（プログラムの連係）。

(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升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“(”及び“)”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 |
| べき乗演算 | ^ | ↑ |
| 単項演算 | ＋，－ | |
| 乗除演算 | ＊，／ | |
| 加減演算 | ＋，－ | ↓ |
| 比較演算 | ＞，＜，≧，≦，＝，≠ | 低 |

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 ＋ $A5 ＋ 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。<br>［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。<br>［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，それ以外のときセル 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。<br>［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。<br>［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。<br>［例1］整数部 (3.9) は，3 を返す。<br>［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。<br>［例1］剰余 (10,3) は，1 を返す。<br>［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。それ以外のとき 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。それ以外のとき 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）<br>四捨五入（算術式，桁位置）<br>切捨て（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。<br>［例1］切上げ(−314.059,2) は，−314.06 を返す。<br>［例2］切上げ(314.059, −2) は，400 を返す。<br>［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。<br>［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 未満の一様乱数（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。<br>［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す。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。<br>・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。<br>・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。<br>［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す。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。<br>・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。<br>・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。<br>［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す。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。<br>［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) | 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。<br>・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。<br>・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。<br>・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。<br>［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。 |
| 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]) | 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。<br>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。<br>［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。<br>［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。 |

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 〕

〔 メ モ 用 紙 〕

6. 退室可能時間に途中で退室する場合には，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。
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，机上に置けるものは，次のものに限ります。
    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。

©2016　独立行政法人情報処理推進機構