平成１８年度　春期　ソフトウェア開発技術者　午後Ｉ問題

問１　IP アドレスとルーティングに関する次の記述を読んで，設問 1～4 に答えよ。

複数のホスト上で動作しているプログラムの間で送受信されるアプリケーションのデータは，IP パケットによって送信元ホストから送信先ホストまで転送される。IP パケットは，送信先ホストの IP アドレスや送信元ホストの IP アドレスが書かれたヘッダ部と，アプリケーションのデータが格納されたデータ部からなる。

IP パケットに含まれる送信先ホストの IP アドレスは，ネットワーク内のホストやルータに設定されたルーティングテーブルの IP アドレスのネットワークアドレス部と比較され，転送先のルータが決定される。この動作が順次繰り返されて，最終的に送信先ホストに送信される。

IP パケットの転送においては，システムごとに IP パケットの生存時間が決められている。生存時間は，IP パケットがルータを一つ通過することに一つずつ減らされる。生存時間が 0 になると IP パケットが破棄され，送信元に［　a　］を通知する。これは，ルーティングテーブルの不具合によって，IP パケットの転送が［　b　］してしまうことを防止するためである。

S 社では，次の図に示すネットワーク構成の社内 LAN を新たに構築することにした。

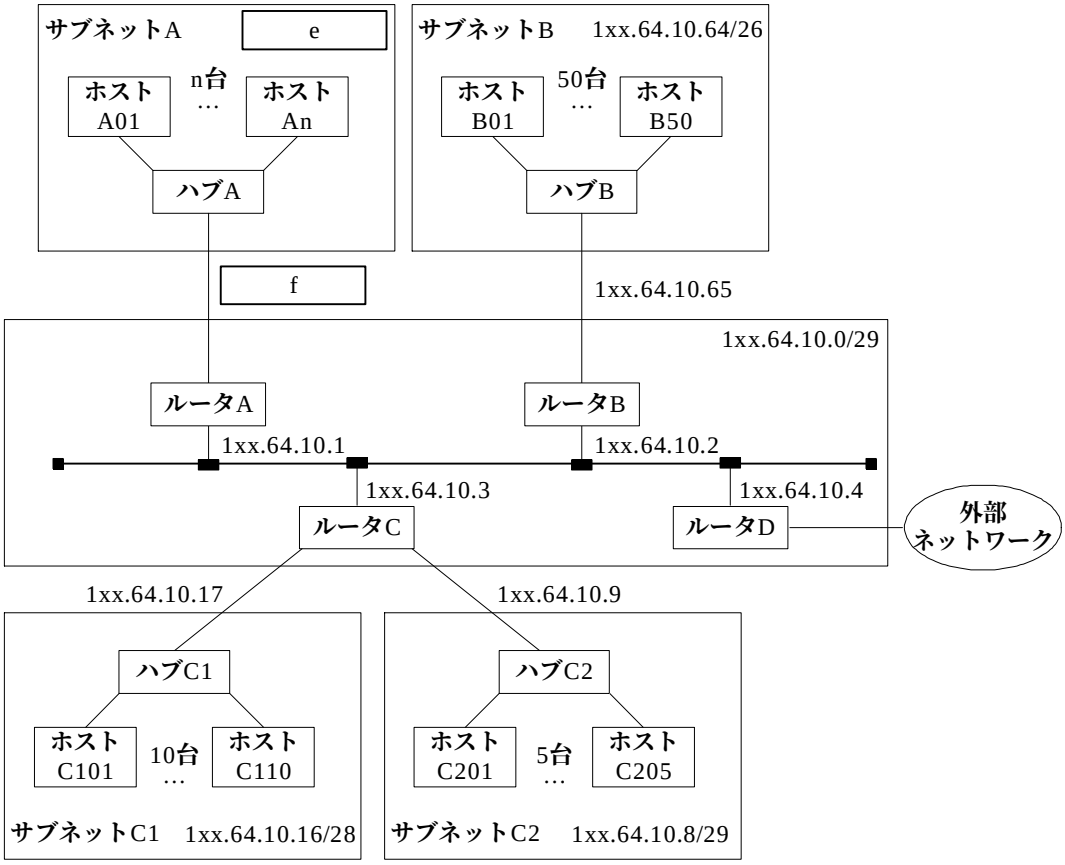


図　ネットワーク構成

S 社の社内 LAN には，IP アドレス 1xx.64.10.0/25 が割り振られている。この場合のサブネットマスクは 255.255.255.128 であり，ネットワークアドレスは [c]，ブロードキャストアドレスは [d] である。

〔S 社の社内 LAN のサブネットにおける IP アドレスの付与基準〕
(1) 最も若いアドレスは，サブネットアドレスである。
(2) サブネットアドレスの次に若いアドレスは，ルータに割り振る。
(3) ホストの番号の順に残りのアドレスを割り振る。
(4) ホストアドレス部のすべてのビットが 1 のアドレスは，ホストのアドレスとして使用しない。

S 社のシステム管理者が, 図の社内 LAN のルータ A～C に対応したルーティングテーブルを次の表 1～3 に示すように設定した。

表１　ルータ A のルーティングテーブル

IP アドレス	転送先
1xx.64.10.8/29	1xx.64.10.3
[g]	1xx.64.10.3
1xx.64.10.0/29	1xx.64.10.1
1xx.64.10.32/27	1xx.64.10.33
1xx.64.10.64/26	1xx.64.10.2
0.0.0.0/0	1xx.64.10.4

表２　ルータ B のルーティングテーブル

IP アドレス	転送先
1xx.64.10.8/29	1xx.64.10.3
[g]	1xx.64.10.3
1xx.64.10.0/29	1xx.64.10.2
1xx.64.10.32/27	1xx.64.10.1
1xx.64.10.64/26	1xx.64.10.65
0.0.0.0/0	1xx.64.10.4

表３　ルータ C のルーティングテーブル

IP アドレス	転送先
1xx.64.10.8/29	1xx.64.10.9
1xx.64.10.16/28	[h]
1xx.64.10.0/29	1xx.64.10.3
1xx.64.10.32/27	1xx.64.10.1
1xx.64.10.64/26	1xx.64.10.2
0.0. 0.0/0	1xx.64.10.4

設問１　[a]，[b] に入れる適切な字句を，それぞれ 3 字以内で答えよ。

設問２　[c]，[d] に入れる適切な値を答えよ。

設問３　表１～３を参考にして，図中の [e]，[f] に入れる適切な IP アドレスを答えよ。また，図中の n（サブネット A のホスト台数）の最大値を求めよ。

設問４　表１～３中の [g]，[h] に入れる適切な IP アドレスを答えよ。

問２　制御パステストに関する次の記述を読んで，設問 1～3 に答えよ。

ソフトウェア開発会社の A 社は，ホワイトボックステストの一つである制御パステストでプログラムを検証することにした。

制御パステストには，すべての命令を最低 1 回実行するテストである命令網羅（C0 網羅），すべての条件分岐の真と偽の両方を最低 1 回テストする分岐網羅（C1 網羅），分岐条件の真と偽のすべての組合せを少なくとも 1 回テストする条件網羅（C2 網羅）などがある。

C2 網羅は，プログラムの開始から終了までの経路に注目したテストである。経路は，プログラムをフローグラフに置き換えることで，フローグラフから求めることができる。

フローグラフは，プログラムを連続した逐次命令群と，while や if などの分岐命令に分け，それぞれをノードとし，処理の順にノードとノードを有向線分（エッジ）で結んだものである。

分岐命令には，AND，OR などを用いない一つだけの条件で，真又は偽に分岐する単独分岐と，単独分岐を AND，OR などで組み合わせた複合分岐がある。複合分岐は単独分岐に分解してからフローグラフに置き換える。

図 1 のプログラムをフローグラフで表すと，図 2 になる。フローグラフのノード番号①～⑩は，図 1 のプログラム中の①～⑩に対応する。①，③，④，⑥，⑦は単独分岐を，②，⑤，⑧，⑨，⑩は連続する逐次命令群を表すノードである。ノード⑪は，出口を表すただ一つの特別なノードである。

C2 網羅の一つの経路とは，フローグラフの開始から，同一のエッジを複数回通過しないで出口に達するノードの列である。例えば，①，②，③，⑤，⑩，①，⑪は経路として扱うが，①，②，③，⑤，⑩，①，②，…のように，①から②へのエッジを複数回通過するものは経路として扱わない。

フローグラフから得られるすべての経路の数を，サイクロマチック数という。サイクロマチック数 S は，フローグラフのエッジ数 E とノード数 N から，次のように求めることができる。

S＝E－N＋2

```
While (①(a=1)) {
        ②
        If (③(b=2) or ④(b=3)) {
                ⑤
        }
        Else {
              If (⑥(c>0) and ⑦(c<8)) {
                      ⑧
              }
              Else {
                      ⑨
                      Break;
              }
        }
        ⑩
}
```

注　Break を実行すると，While の繰返しを終了する。

図１　プログラム

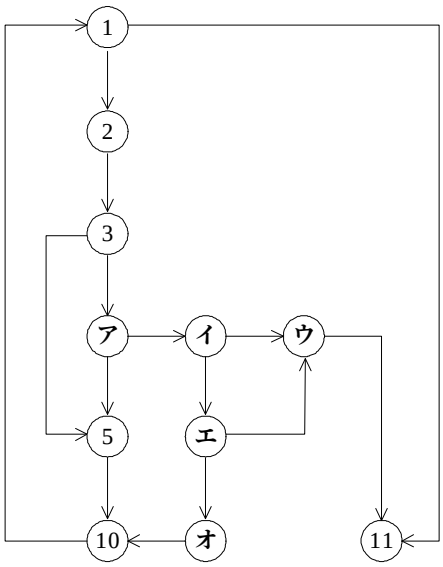


図２　フローグラフ

設問１　図 2 のフローグラフ中の (ア) ～ (オ) に入れる適切なノード番号を答えよ。

設問２　図 2 のフローグラフのサイクロマチック数を答えよ。

設問３　図 1 のプログラムに対して，“①が真，③が偽，④が偽，⑥が偽，⑦が真”となるようなテストケースを実行するために必要な，変数 a，b 及び c の設定値を，考えられる最も大きな値で答えよ。

ここで，変数 a，b 及び c は，16 ビットで表された 2 の補数の整数型変数である。これらの変数を，②で変更することはない。

問３　IC カードを使った,社員の入退室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。

多数の個人情報を取り扱っている T 社では，各社員に社員 ID（公開データ）や認証用の暗号化鍵を書き込んだ IC カードを配付し，事務室や機器室の入退室管理を行うシステム（以下，本システムという）を構築することになった。本システムは，サーバ及び各部屋の出入口に設置された端末から構成され，これらの機器は専用のネットワークで接続される。端末はカードリーダを備え，挿入された IC カードの認証を行って出入口の施錠及び解錠の制御を行い，正規の IC カードをもつ社員だけを入退室可能とする。

セキュリティ強度のほか，利便性やコストも考慮して，本システムでは CPU とメモリをもつ IC カードを使用する。IC カードが正しいことを確認するために，次のような認証方式（図）の採用を検討することにした。

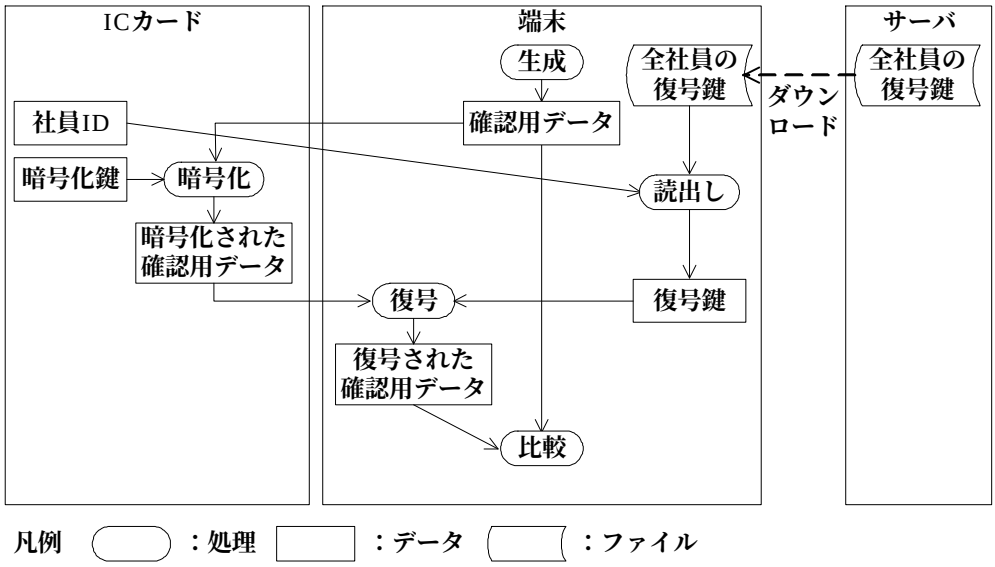


図　IC カードの認証方式

端末は，IC カードが挿入されると，適切な確認用データを生成して IC カードに入力する。IC カードは，書き込まれている暗号化鍵を使って確認用データを暗号化し，社員 ID とともに出力する。端末は，暗号化された確認用データを当該社員用の復号鍵で復号して，元の確認用データと比較し，両者が一致すれば，挿入された IC カードが正しいことを認証する。

社員 ID 及び認証に必要なデータはサーバで設定や保管を行うが，サーバやネットワークの障害時でも端末単独で入退室を可能にするために，それらのデータは端末にダウンロードしておく。

暗号化鍵は，IC カードの耐タンパ性（内部の情報を不正に読み書きすることを困難にする特性）をもつメモリに格納し，IC カードの外には出さない。したがって，たとえ IC カードのハードウェアの作成は可能であっても，IC カードを不正に発行することは困難である。

認証に使用する暗号方式に関して，次の 4 種類について検討した。

① 共用の共通鍵を用いる暗号方式

共通鍵暗号方式を採用し，全社員で同じ共通鍵を共用する。

② 個別の共通鍵を用いる暗号方式

共通鍵暗号方式を採用し，各社員に個別の共通鍵を割り当てる。

③ 公開鍵暗号方式

公開鍵暗号方式を採用し，暗号化に秘密鍵，復号に公開鍵を使う。各社員に個別の一対の秘密鍵と公開鍵を割り当てる。

④ 公開鍵暗号+認証局方式

方式③の公開鍵暗号方式に加え，認証局（以下，CA という）を設置し，社員 ID 及び公開鍵を含む情報に CA の秘密鍵でディジタル署名した公開鍵証明書を，各社員に対して発行する。各社員の公開鍵はサーバや端末へ配付せずに，端末では認証の都度，公開鍵と公開鍵証明書を IC カードから読み出し，その正当性を CA の公開鍵を使って確認した上で使用する。

なお，IC カードの紛失時又は盗難時に，鍵を変えて IC カードを再発行する場合には，以前発行した公開鍵証明書を無効にする証明書失効リスト（以下，CRL という）を作成する。

これらの各方式について，IC カード，端末，サーバ及び CA の各装置でもつ必要がある認証用データを，表 1 に示す。

表１　各装置でもつ必要がある認証用データ

方式	IC カード	端末，サーバ	CA
① 共用の共通鍵を用いる暗号方式	共用の共通鍵	共用の共通鍵	―
② 個別の共通鍵を用いる暗号方式	当人の共通鍵	a	―
③ 公開鍵暗号方式	当人の秘密鍵	b	―
④ 公開鍵暗号＋CA 方式	当人の秘密鍵 当人の公開鍵 当人の公開鍵証明書	c CRL	d CA の公開鍵 公開鍵証明書発行・廃棄履歴 CRL

各方式について，各装置（1 台又は 1 枚）でもつ認証用データが漏えいした場合に，IC カードを不正に発行されるおそれがある社員の範囲を表 2 に示す。

なお，IC カードのハードウェアが作成されるおそれはあるものとする。

表２　各装置でもつ認証用データが漏えいした場合に，IC カードを不正に発行されるおそれがある社員の範囲

方式	IC カード	端末，サーバ	CA
① 共用の共通鍵を用いる暗号方式	e	全員	―
② 個別の共通鍵を用いる暗号方式	当人だけ	f	―
③ 公開鍵暗号方式	当人だけ	g	―
④ 公開鍵暗号＋CA 方式	当人だけ	h	i

設問１　表１中の [a] ～ [d] に入れる適切な字句を解答群の中から選び，記号で答えよ。

解答群

ア　CA の公開鍵	イ　CA の秘密鍵
ウ　全員の共通鍵	エ　全員の公開鍵
オ　全員の公開鍵証明書	カ　全員の秘密鍵
キ　当人の共通鍵	ク　当人の公開鍵
ケ　当人の公開鍵証明書	コ　当人の秘密鍵

設問２　表 2 中の [e] ～ [i] に入れる適切な字句を解答群の中から選び，記号で答えよ。解答は重複して選んでもよい。

解答群

ア　全員　　イ　当人だけ　　ウ　なし

設問３　認証に使用する暗号方式①～④の評価に関する次の記述中の [j], [k] に入れる適切な字句を答えよ。

評価項目として，データ漏えいに対するセキュリティ強度や漏えいした場合の影響の度合いのほか，社員の追加，削除に当たってのデータの生成，設定，配付の手間などの運用性も挙げられる。使用条件によって，各評価項目の重要性を考慮して各方式を評価することが必要である。

社員数が多い，社員の入社，退職の頻度が高いなどの事情から，運用性が特に重要な場合には，方式①が適している。ただし，この方式では，IC カードの耐タンパ性が破られた場合の影響は大きい。

サーバ及び CA は，厳重に管理及び運営することによってデータ漏えいのおそれを十分に小さくできる一方，全端末の管理を厳重に行うのは困難な場合，データ漏えい時の影響の大きさを考慮すると，[j] 鍵暗号を用いた方式が望ましい。ただし，サーバ障害時などの入退室管理の続行は考えず，端末では認証用データをもたずにオンラインでサーバにおいて認証を行うことにすれば，[j] 鍵暗号を用いた方式でなくてもよい。

方式④は，CRL の更新頻度が低ければ運用性は比較的良いが，もし CA でもつデータが漏えいすれば [k] が不正に作成され，IC カードが不正に発行されるおそれが大きくなってしまう。

なお，公開鍵暗号を用いた方式では，十分なセキュリティ強度を実現するための処理負荷が大きいので，応答時間を短く抑えるには IC カードに相応の性能が必要である。

問４　バッチジョブ処理時間の評価に関する次の記述を読んで，設問 1，2 に答えよ。

X 社の営業分析システムは，バッチ処理が主体であり，毎日午前 2 時に処理を開始し，午前 5 時半までにすべての処理を終了しなければならない。すべての処理が終了すると，営業部の PC から営業分析情報を参照することができる。X 社の情報システム部に所属する Y 氏は，営業分析システムのバッチジョブ群 Z1 の稼働状況を調査した。図 1 に Z1 のジョブチャートを，表に Z1 の処理概要と処理時間をそれぞれ示す。

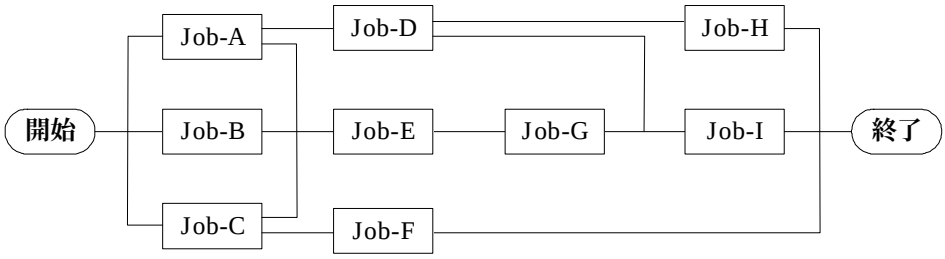


図１　Z1 のジョブチャート

表　Z1 の処理概要と処理時間

ジョブ名称	処理概要	処理時間（分）
Job-A	販売情報取得	30
Job-B	顧客情報取得	35
Job-C	商品情報取得	25
Job-D	売上・コスト分析	85
Job-E	一次集計分析	30

ジョブ名称	処理概要	処理時間（分）
Job-F	商品在庫分析	75
Job-G	営業分析	60
Job-H	財務分析	65
Job-I	レポート作成	50

〔Z1 の処理条件〕

- 毎日午前 2 時に処理を開始する。
- 各ジョブは，図 1 の“開始”から“終了”の方向へ順次処理される。
- 各ジョブは，先行ジョブのすべてが終了した時点で開始できる。
- 図 1 の処理順序を保証した上で，複数ジョブの並行処理が可能である。
- ジョブ間の待ち時間はない。
- CPU，メモリ，ディスク装置などのシステム資源に不足はない。

その後，営業部の強い要望によって，新たな機能として，毎日の売上ランキングを提供することになった。Y 氏はこの機能を実現するために，Z1 の Job-D 及び Job-G の後続ジョブとして，新たにランキング分析ジョブ Job-J を追加し，このジョブ群を Z2 とした。図 2 に Z2 のジョブチャートを示す。

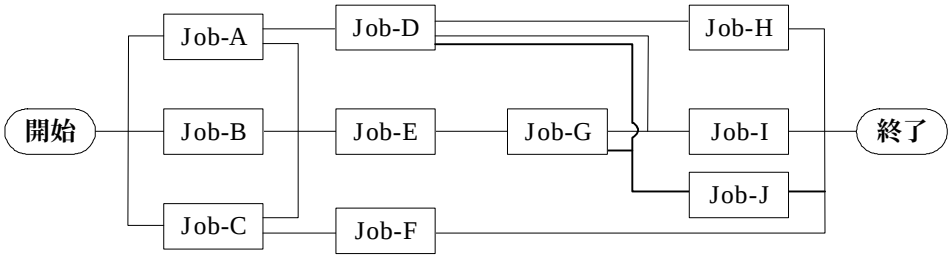


図２　Z2 のジョブチャート

〔Z2 の処理条件〕

・Job-J において，データ 1 件当たりの処理時間を 0.1 秒とする。
・そのほかは，〔Z1 の処理条件〕と同一とする。

さらに，毎月末日に過去半年分の月間売上ランキングを提供するために，Job-J は過去 5 か月分の集計済売上上位データを併せて読み込んで処理する必要があった。このままでは Job-J によって全体処理時間が長くなりすぎ，月末日の処理が遅れてしまうことは明らかであった。そこで Y 氏は，処理時間短縮策として，Job-J のジョブ分割を検討した。図 3 に，Job-J 分割後の Job-JJ のジョブ構成を示す。

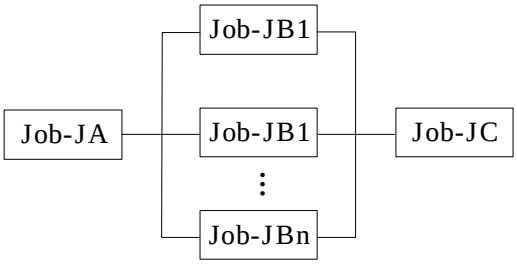


図３　Job -JJ のジョブ構成

〔Job-JJ のジョブ分割条件及び処理時間算出条件〕

・Job-JBk（k：1～n）の n は，ジョブ分割数を表す。
・Job-JJ は，“ Job-JA → Job-JBk → Job-JC ”の順に待ち時間なく処理される。
・分割後の各ジョブ“ Job-JBk ”のデータ処理件数は均等とする。前処理 Job-JA が終了した後，同時に開始し，同時に終了する。
・Job-JBk において，データ 1 件当たりの処理時間を 0.1 秒とする。
・Job-JA と Job-JC の合計処理時間は，“ ジョブ分割数（n）×5 ”分とする。
・CPU，メモリ，ディスク装置などのシステム資源に不足はない。

設問１　次の記述中の［　a　］，［　b　］に入れる適切な字句を答えよ。

なお，［　a　］については，次の（例）にならい，“ Job-X ”（ジョブ名称，X は A～I）と“ → ”を用いて答えよ。

（例）　Job-A → Job-B → Job-C → …

〔Z1 の処理時間〕

各ジョブの処理時間を表のとおりとすると，Z1 のクリティカルパスは［　a　］であり，Z1 のすべてのジョブが終了するには，［　b　］分かかる。

設問２　次の記述中の［　c　］～［　f　］に入れる適切な字句を答えよ。

〔Z2 の処理時間〕

月中日における Job-J のデータ処理件数を 36,000 件とすると，Job-J の処理時間は，［　c　］分である。

また，月末日における Job-J のデータ処理件数を 216,000 件（月中日の 6 倍）とすると，Z2 が午前 5 時半までに終了するためには，Job-JJ の本処理部分（Job-JBk）を［　d　］本又は［　e　］本にジョブ分割すればよい。このときの Z2 の処理時間は，どちらも［　f　］分である。

なお，Job-JJ 以外の処理時間は，表のとおりとする。

問５　グラフの最短経路に関する次の記述を読んで，設問 1～3 に答えよ。

都市間を道路で移動するときの最短の移動経路を求める問題は，それぞれの都市をグラフのノード，都市間の道路をグラフのエッジ，あるノードから別のノードへエッジやノードを通って移動する道筋（パス）を経路と考えることで，グラフの問題として解くことができる。つまり，グラフ G において，ノード間のエッジと距離が与えられているとして，二つのノード間を最も短い距離で移動する経路（最短経路）を求める問題として考えればよい。

グラフ G の例を図 1 に示す。G は A～E の五つのノードと，ノード間を結ぶエッジからなる。各エッジには移動可能な方向と距離が決められている。図 1 では，ノード間で移動可能な方向を矢印で，距離（0 より大きい）を矢印に添えた数字で示している。

なお，二つのノード間にエッジがない場合は，ノード間を直接移動できないことを示している。

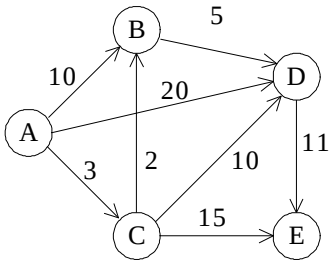


図１　グラフ G の例

ここで，G 内に一つの始点ノードを定め，始点ノードから各ノードまでの最短経路での移動距離（以下，最短距離という）を求めることを考える。ただし，すべてのノードは始点ノードから到達可能であるとする。最短距離を求めるアルゴリズムの一つとして，ダイクストラのアルゴリズムが知られている。このアルゴリズムは，始点ノードからの最短距離を，始点ノードの周辺のノードから一つずつ順番に確定し，確定しているノードの範囲を徐々に拡大していきながら，最終的にすべてのノードまでの最短距離を求めていく。ダイクストラのアルゴリズムの手順を①～④に示す。

① 始点ノードからの最短距離が確定しているノードの集合を V，未確定のノードの集合を U とする。初期状態では，すべてのノードが U に属している。また，ノードごとに始点ノードからの距離（以下，ノードの距離という）をもたせる。ノードの距離は初期値として十分に大きな値をもつ。
② 始点ノードの距離を 0 とする。
③ U に属するノードの中で，ノードの距離が最も小さいノードを探し，そのノードを n とする。その時点でのノード n の距離を n の最短距離として確定し，ノード n を U から V へ移す。このとき，U に属するほかのノードについて距離を更新する。U に属するノードの中で，n からのエッジがあるノード（k とする）すべてについて距離の更新を行い，ノード k の距離を，その時点までの k の距離と，n の距離と n から k へのエッジの距離の和を比較して，小さい方の値に置き換える。
④ U に属するノードが全部 V へ移されるまで，③の操作を繰り返す。

ここで，図 1 のグラフ G の例において，始点ノードを A として，①～④で説明した手順によって，各ノードまでの最短距離を順次求めることを考える。表 1 は，U から V へノードが移され，各ノードまでの距離が更新された時点での各ノードの距離を示した進行表である。

なお，M は①の初期値を表す。

表１　ダイクストラのアルゴリズムの進行表

UからVへ移されるノード	A	B	C	D	E
（初期状態）	0	M	M	M	M
A	0	10	3	20	M
a					
b					
D	0	5	3	10	18
E	0	5	3	10	18

①～④で示したダイクストラのアルゴリズムを実現するために，表 2 に示す定数，配列（配列の添字は 1 から始まる）及び変数を使用することにした。

表２　使用する定数，配列，変数

定数名／配列名／変数名	定数の意味／保持するデータ内容
N_NODE	最大のノード数
MAX_VALUE	ノード間の距離の総和よりも十分に大きな値
VISITED，UNVISITED	ノードの状態
distance[N_NODE]	ノードごとに始点ノードからの距離を保持する配列で，初期値は MAX_VALUE
status[N_NODE]	ノードごとの状態（VISITED 又は UNVISITED）を保持する配列で，初期値は UNVISITED
w[N_NODE,N_NODE]	ノード間の直接の経路の距離をもつ二次元配列で，ノード i からノード j への距離は w[i,j]となる。ノード間に経路がない場合，該当する要素の値は MAX_VALUE とする。
start	始点ノードの番号
datasize	グラフのノード数

ダイクストラのアルゴリズムを図 2 に示す。

```
配列 distance, status を初期化する
distance[start] ← 0

for (j を１から datasize まで１ずつ増やす)
    minValue ← MAX_VALUE
    /*距離が最小のノードを見つける*/
    for(i を１から datasize まで１ずつ増やす)
      if(status[i] = UNVISITED かつ [    c    ] )
        nodeNumber ← i
        minValue ← distance[i]
(ア) ⟶
      endif
    endfor
    if(minValue が MAX_VALUE に等しい)
      エラーメッセージを表示して終了する
    endif
    [    d    ] ← VISITED
(イ) ⟶
    /*距離を更新する*/
    for(i を１から datasize まで１ずつ増やす)
      if(status[i] = UNVISITED かつ
          distance[i]が [    e    ] より大きい)
        distance[i] ← [    e    ]
(ウ) ⟶
      endif
    endfor
(エ) ➞
endfor
```

図２　ダイクストラのアルゴリズム

設問１　表 1 中の [a] , [b] に入れる適切な字句を答えよ。

設問２　図 2 中の [c] ～ [e] に入れる適切な字句を答えよ。

設問３　図 2 のアルゴリズムでは，各ノードの始点ノードからの最短距離は求まるが，どのノードを通過してきたかという経路は示せない。そこで，最短距離で移動するときの経路を示すために，配列 previous[N_NODE]を導入する。この配列には，最短経路をたどるとき，各ノードに至る経路の最後のエッジがどのノードから来たかを，ノードごとに保持する。

（1）最短経路を記録するためには，図 2 のどこに配列 previous の要素への代入式を挿入すればよいか。挿入する場所を図 2 中の（ア）～（エ）の中から選べ。また，挿入する代入式を答えよ。

（2）始点ノードから各ノードに至る最短経路を表示するアルゴリズムを，図 3 に示す。図 3 は，図 2 のアルゴリズムに（1）の修正を加えて実行した後に実行される。図 3 中の [f] ～ [i] に入れる適切な字句を答えよ。

なお，スタックを操作する関数 push，pop は，最短経路を表示するアルゴリズム内で使用される関数である。また，図 3 中で使用する配列，変数及び定数を，表 3 に示す。

表３　図 3 のアルゴリズムで使用する配列，変数，定数

配列名／変数名／定数名	保持するデータ内容／定数の意味
route[N_NODE]	最短経路中のノード番号を保持する配列で，スタックとして使用する。
top	配列 route をスタックとして使用する場合に，スタックトップの位置を示す。初期値は 0 とする。
NULL	スタックが空であることを示す定数

```
for(i を１から datasize まで１ずつ増やす)
  push(i)
  j ← i
  while(j が start でない)
    j ← previous[j]
    [ f ]
  endwhile
  j ←pop( )
  while（j が NULL でない）
    j を表示する
    [ g ]
  endwhile
  改行を表示する
endfor

function push(x)
  if（top が N_NODE より小さい）
    top ← top+1
    [ h ]
  endif
endfunction

function pop( )
  if（top が 0 より大きい）
    [ i ]
    route[top+1]を戻り値として戻る
  else
    NULL を戻り値として戻る
  endif
endfunction
```

図３　最短経路を表示するアルゴリズム

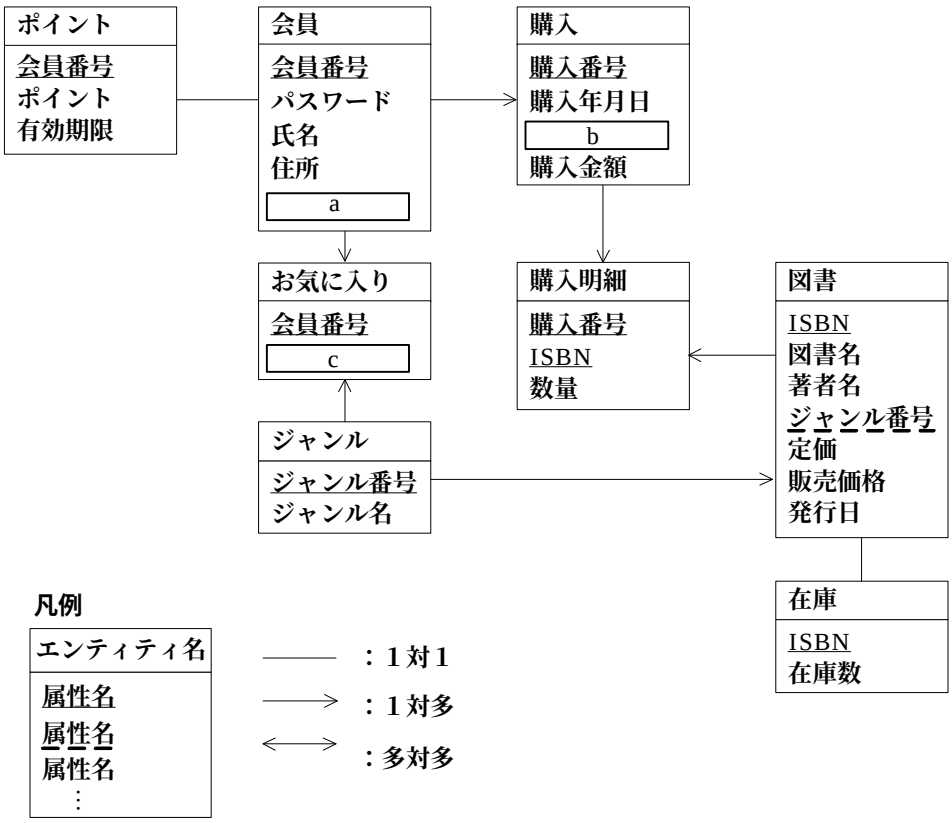
問６　インターネット図書販売システムに関する次の記述を読んで，設問 1～4 に答えよ。

M 社は，インターネット上で図書を販売するシステム（以下，本システムという）を構築することになった。本システムの構築目的は，インターネットに接続している不特定多数の消費者に対して，個人の興味・好みに合わせた図書の推薦やポイントサービスを行って，売上の向上を図ることである。

〔本システムの概要〕

・本システムを利用するには，会員登録を行う必要がある。会員登録を行う際には，パスワード，氏名，住所，メールアドレスを入力する。
　なお，会員番号は本システムが自動的に採番する。また，パスワード，メールアドレスの入力は必須とする。
・本システムに会員登録を行った利用者（以下，会員という）は，会員番号，パスワードを入力して本システムにログインすることで，本システムのサービスを受けることが可能になる。
・会員は興味のあるジャンルをお気に入りとして登録することで，本システムから興味のあるジャンルの推薦図書情報を受け取ることが可能になる。興味のあるジャンルは，複数登録することができる。
・会員が図書を購入した際には，その履歴として購入番号，購入年月日，会員番号，購入金額を本システムが記録する。また，その明細として ISBN と数量を記録する。
　なお，購入番号は本システムが自動的に採番する。
・図書ごとに在庫数を管理する。
・図書の購入金額（円）の 1%がポイントとして会員に付与される。
・付与されたポイントは，会員が図書購入時に 1 ポイント 1 円として使用することができる。
・累積したポイントの有効期限は，最後に図書を購入した日の 1 年後とする。

本システムの E-R 図を図 1 に示す。



注　属性名の実線の下線 ＿＿ は主キー，破線の下線 ＿ ＿ ＿ は外部キーを示す。
主キーの実線が付いている属性名には，外部キーの破線を付けない。

図１　本システムのＥ-Ｒ図

本システムでは，E-R 図のエンティティ名を表名，属性名を列名にして，適切なデータ型で表定義した関係データベースによって，データを管理することにする。

設問１　図１の E-R 図中の [a] ～ [c] に入れる適切な属性名を答えよ。主キーの場合は実線の下線を，外部キーの場合は破線の下線を引いて示せ。

設問２　表定義について，(1)，(2) に答えよ。

(1) 会員テーブルを定義する次の SQL 文中の [d]，[e] に入れる適切な字句を答えよ。

なお，[a] には図 1 中の [a] と同一のものが入る。

```
CREATE TABLE 会員
(会員番号 CHAR(9) [ d ] ,
パスワード CHAR(15) [ e ] ,
氏名 VARCHAR(20) ,
```

住所 VARCHAR(60) ,
[a] VARCHAR(30) [e])

(2)図書テーブルを定義する次の SQL 文中の [f] ～ [h] に入れる適切な字句を答えよ。

なお，列名以外の列定義は省略してあり，[d] には (1) の SQL 文中の [d] と同一のものが入る。

CREATE TABLE 図書
(ISBN, 図書名, 著者名, ジャンル番号, 定価, 販売価格, 発行日,
[d] ([f]),
FOREIGN KEY ([g]) REFERENCES [h])

設問３　本システムの図書購入処理におけるデータの整合性確保に関する次の記述中の [i], [j] に入れる適切な記号を，図 2 中のア～クの中から選べ。

本システムにおいて，会員が図書を購入するには，購入したい図書の情報を確認した後，図書の送り先の情報を確認し，購入指示を出す。会員が本システムへ図書の購入を指示すると，図 2 に示す一連の処理が実行される。この処理において，データの不整合を回避するために，DBMS のトランザクション機能を利用する場合，設定すべき最も狭いトランザクション範囲は，[i] ～ [j] になる。

なお，図書購入処理中に，図書の属性情報が変更されることはないものとする。

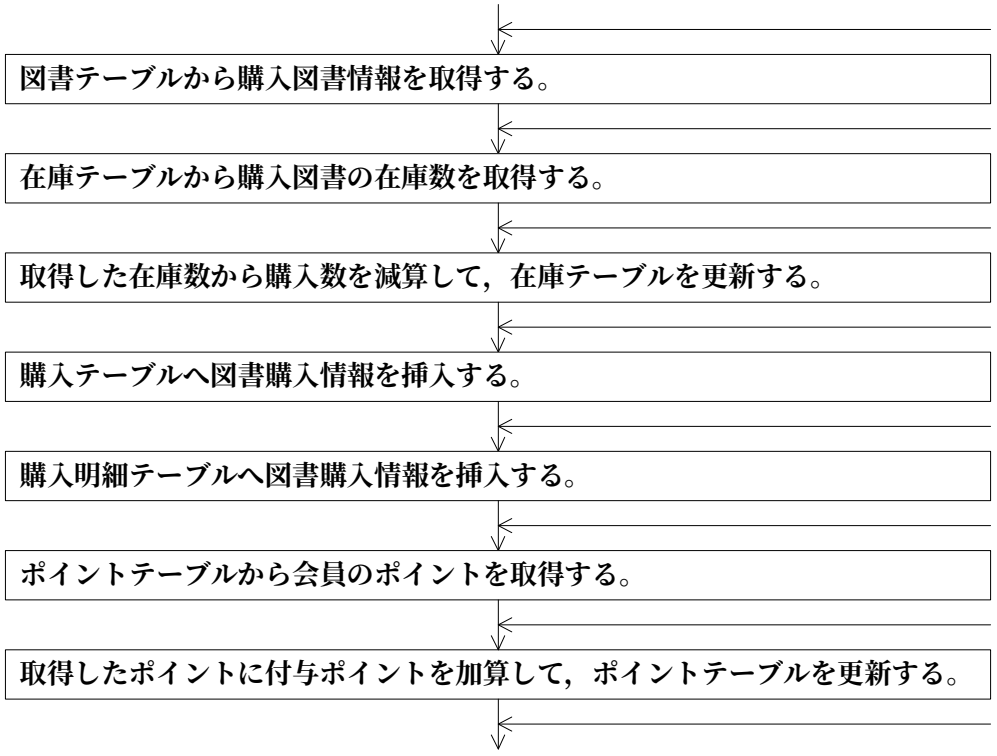


図２　図書購入処理

設問４　本システムへログインした会員が，お気に入りとして登録しているジャンルの中で，先月の初日以降に発行され，在庫があり，その会員がまだ購入していない図書を抽出する SQL 文を次に示す。SQL 文中の [k] ～ [n] に入れる適切な字句又は式を答えよ。ただし，“：先月初日”，“：会員番号”は，それぞれ先月の初日，ログインした会員の会員番号を表すホスト変数である。

なお，列を表す場合，表名を省略してはならない。

```
SELECT 図書.ISBN,図書.図書名
FROM 図書,在庫
WHERE [ k ]
  AND 図書.発行日 >= :先月初日
  AND 在庫.在庫数 > 0
  AND 図書.ジャンル番号 IN
  (SELECT [ l ]
    FROM お気に入り
    WHERE [ m ])
  AND [ n ]
  (SELECT 購入明細.ISBN
    FROM 購入,購入明細
    WHERE 購入.購入番号 = 購入明細.購入番号
      AND 購入.会員番号 = :会員番号)
```