問1　百貨店における Web サイトの統合に関する次の記述を読んで，設問 1～5 に答えよ。

C 社は，半年ほど前に旧 A 社と旧 B 社が合併してできた会社である。旧 B 社が存続会社となり，旧 A 社の事業を承継した上で，C 社に改称した。C 社は，旧 A 社の 10 店舗の百貨店（以下，A 百貨店という）と，旧 B 社の 5 店舗の百貨店（以下，B 百貨店という）を運営している。

C 社は，旧 A 社が発行していたクレジットカードの会員向け Web サイト（以下，サイト P という），A 百貨店の取扱商品を販売するオンラインストア Web サイト（以下，サイト Q という），及び旧 B 社のポイントカードを保有する会員向け Web サイト（以下，サイト R という）を運営している。

合併後，旧 A 社クレジットカードの B 百貨店での利用を促進したり，旧 B 社ポイントカードの A 百貨店での利用を可能にしたりするなど，両社の顧客サービスの融合に力を入れている。表 1 は，サイト P，Q，R の概要である。

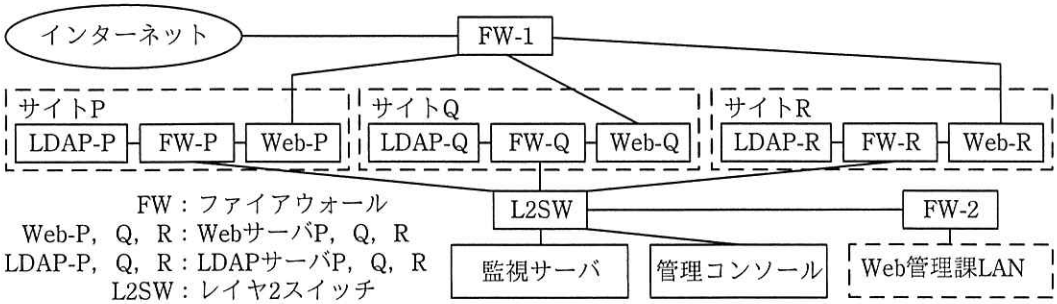
表 1　各サイトの概要

サイト名	機能	利用者 ID
サイト P	・利用明細の表示 ・クレジットカード利用ポイントの残高確認，商品又は他社のポイントとの交換申請 ・各種申請書類の送付依頼 ・アカウント管理（新規登録，パスワード変更，登録事項変更など）	英数字 8～16 字の文字列を利用者が設定する。
サイト Q	・A 百貨店の商品の購入 ・購入履歴の表示 ・アカウント管理（新規登録，パスワード変更，決済用クレジットカード登録など）	英数字 8～16 字の文字列を利用者が設定する。
サイト R	・ポイント獲得履歴，利用履歴の表示 ・ポイント獲得履歴から購買傾向を分析し，傾向に基づいたお買い得情報の表示 ・キャンペーンへの応募，応募履歴の表示 ・アカウント管理（新規登録，パスワード変更，パスワード失念時の処理など）	ポイントカードに記載されている数字 8 桁の会員番号が割り当てられる。

〔機器の集約と運用作業の効率化〕

C 社では，運用作業を効率化するために，別々の場所に設置されていた各サイトを構成する機器を 1 か所のデータセンタに集約するとともに，これらの機器を管理する

Web 管理課を新設した。機器集約後のネットワーク構成は図１のとおりである。



FW：ファイアウォール
Web-P，Q，R：WebサーバP，Q，R
LDAP-P，Q，R：LDAPサーバP，Q，R
L2SW：レイヤ2スイッチ

注記　FW-1 では，サイト P，Q，R 相互間の通信を禁止している。

図１　機器集約後のネットワーク構成（抜粋）

サイト P，Q，R は，それぞれ独立して運営されている。いずれのサイトも，アカウント情報はサイトごとに設置した LDAP サーバのユーザエントリとして管理しており，ログイン時の認証には，各 LDAP サーバのユーザエントリ内の利用者 ID とパスワードを利用している。

〔サイト間でのアカウントの共通利用〕

C 社では経営戦略の一環として，三つのサイトで収集した情報から顧客の購買傾向を分析することにした。そこで，サイト P，Q，R でクロス分析などの手法を用いて購買傾向を分析する上で，各サイト間で同一の顧客を特定するために，サイト P，Q，R 相互間でのアカウントの共通利用を実現することにした。

アカウントの共通利用では，次の三つの利用方法のいずれかを各顧客に選択してもらうことにした。

(1) サイト P のアカウントを親アカウントとし，サイト Q，R のアカウントを子アカウントとして，子アカウントを親アカウントに紐付（ひも）ける。主に旧 A 社の顧客向けである。

(2) サイト R のアカウントを親アカウントとし，サイト P，Q のアカウントを子アカウントとして，子アカウントを親アカウントに紐付ける。主に旧 B 社の顧客向けである。

(3) アカウントの共通利用をしない。

顧客がアカウントの紐付けを設定すれば，子アカウントの代わりに親アカウントを用いて各サイトにログインできる。

〔アカウントの共通利用の設計〕

Web 管理課の J 主任は，アカウントの共通利用の設計を任された。J 主任は，アカウントの共通利用を実現するために，次の四つを行うことにした。

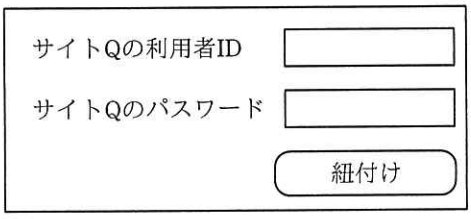
(1) サイト P，Q，R のログイン処理を変更する。

(2) LDAP-P，LDAP-R で管理するアカウント情報に，紐付け情報を保存する。具体的には，LDAP-P のユーザエントリに siteQid，siteRid 属性を追加して，当該アカウントに紐付けた子アカウントの利用者 ID を保存する。LDAP-R のユーザエントリには，sitePid，siteQid 属性を追加する。

なお，紐付け前の sitePid，siteQid，siteRid には，空文字列が設定される。

(3) Web-P，Web-R の Web アプリケーションプログラム（以下，Web アプリケーションプログラムを Web アプリという）に，アカウントの紐付け機能を追加する。

サイト P のアカウントを親アカウントとし，サイト Q のアカウントを子アカウントとして紐付けるときのサイト P の画面と処理内容は図 2 のとおりである。



［紐付け］ボタンが押された場合の処理

(i) Web-PのWebアプリが，LDAP-Qに問合せて，入力されたサイトQの利用者IDとパスワードを用いて認証を行う。

(ii) 認証に失敗したら，エラー画面を表示する。

(iii) Web-PのWebアプリが，LDAP-Pの該当するユーザエントリのsiteQid属性に，サイトQの利用者IDを書き込んで完了画面を表示する。

注記　サイト P にログイン済みである。

図 2　サイト P におけるサイト Q のアカウントの紐付けの画面と処理内容

(4) FW-P，Q，R のルールに対して必要な変更を行う。例えば，変更後の FW-P のルールは，表 2 のようにする。

表 2　変更後の FW-P のルール

項番	送信元	宛先	プロトコル	動作
1	監視サーバ，管理コンソール，Web 管理課 LAN	Web-P，LDAP-P	管理用プロトコル	許可
2	Web-Q，Web-R	LDAP-P	LDAP	許可
3	Web-P	a	b	許可
⋮	⋮	⋮	⋮	⋮
15	全て	全て	全て	拒否

注記 1　FW-P は，ステートフルパケットインスペクション型である。
注記 2　項番が小さいルールから順に，最初に合致したルールが適用される。
注記 3　項番 4～14 には，LDAP に関するルールは記述されていない。

〔個人情報の取扱い〕

J 主任は，C 社の法務担当の M さんに，アカウントの共通利用について説明し，個人情報の取扱いの観点から問題がないかどうか相談した。M さんは，合併前後の個人情報の利用目的の内容について確認した。

確認後，M さんは，アカウントの共通利用には，顧客に対して利用目的の変更を通知して，同意を得る必要があると指摘した。

指摘を受け，J 主任は，顧客がアカウントの共通利用を選択したときに，個人情報の利用目的の変更を通知するとともに，変更後の利用目的を明示して同意を得る機能を加えることにした。

〔コードレビュー〕

アカウントの共通利用の設計を終えた J 主任は，サイト P の Web アプリの改修に着手した。

図 3 及び図 4 は，サイト P のアカウントにサイト Q のアカウントを紐付ける場合のサイト P 上での紐付け処理の Java ソースコードである。

利用者が，サイト P にログイン後，図 2 の画面を操作し，“紐付け”ボタンをクリックした場合，図 4 の行番号 101 の siteID にサイトの識別文字列として“siteQ”が代入された状態で図 4 のコードの実行が開始される。図 4 のコードが正常に終了したら，紐付けが完了したことを表示する。

```
（省略）　// パッケージ宣言，インポート宣言
          // インポート宣言には，javax.naming.NamingException を含む。
public class AccountLink {
    boolean childChecked;   // 子アカウントの利用者 ID のチェック完了フラグ
    String childSite;       // 子アカウントのサイトの識別文字列
    String childID;         // 子アカウントの利用者 ID
    String childPW;         // 子アカウントのパスワード
    String parentID;        // 親アカウントの利用者 ID
    public static final int NO_ERROR = 0;                  // 正常終了コード
    public static final int ERROR_SITE_UNAVAILABLE = -101; // エラーコード
    public static final int ERROR_ID_OR_PW = -102;         // エラーコード
    public static final int ERROR_LINK_FAILED = -103;      // エラーコード

    public AccountLink(String site, String id, String pw, String loginID) {
        childChecked = false;
        childSite = site;
        childID = id;
        childPW = pw;
        parentID = loginID;
    }

    public int checkChild() throws NamingException {
        // サイトの識別文字列のチェック
        childChecked = childSite.equals("siteQ") || childSite.equals("siteR");
        if (!childChecked) {     // サイトの識別文字列が正当でなかった場合
            return ERROR_SITE_UNAVAILABLE;
        }
        switch (childSite) {
        case "siteQ":
            try {
                // 認証情報チェック
                if (siteQAuth(childID, childPW) == NO_ERROR) {
                    childChecked = true;    // 認証成功
                } else {
                    childChecked = false;   // 認証失敗
                }
            } catch (NamingException e) {   // 認証が実行できなかった場合
                throw e;
            }
            if (!childChecked) {
                return ERROR_ID_OR_PW;
            } else {
                return NO_ERROR;
            }
```

図 3　Web-P の Web アプリにおける AccountLink のクラス定義

```
39:         case "siteR":
                (省略) // サイトRに対して認証を行い，結果に応じてERROR_ID_OR_PW又は
                       // NO_ERROR を返して終了する。
40:         default:
                (省略) // ERROR_SITE_UNAVAILABLEを返して終了する。
41:         }
42:     }

43:     public int makeLink() throws NamingException {
            (省略) // LDAP-P上のparentIDのユーザエントリの属性siteQid又はsiteRid
                   // にchildIDを書き込む。
44:     }

45:     private int siteQAuth(String qID, String qPW) throws NamingException {
            (省略) // 認証情報を確認し，認証成功ならNO_ERRORを，認証失敗ならNO_ERROR
                   // 以外の値を返す。必要な通信ができないなど，認証そのものが実行
                   // できない場合，例外NamingExceptionを投げる。
46:     }
        (省略) // その他のメソッドなどの定義
47: }
```

図3　Web-PのWebアプリにおけるAccountLinkのクラス定義（続き）

```
101:    AccountLink idPair = new AccountLink(siteID, userID, userPassword, loginId);
          // 各変数には次の内容が代入されている。
          //   siteIDには，紐付けるアカウントのサイトの識別文字列
          //   userIDには，紐付けるアカウントの利用者ID
          //   userPasswordには，紐付けるアカウントのパスワード
          //   loginIdには，現在サイトPにログインしている利用者ID
102:    int result = AccountLink.NO_ERROR;
103:    for (int retryCount = 1; retryCount < 4; retryCount++) {
104:        try {
                // サイトの識別文字列，利用者ID/パスワードの組みを確認する。
105:            result = idPair.checkChild();
106:            if (result == AccountLink.NO_ERROR) {
107:                break;
108:            } else {
                    (省略) // resultの値に従った適切な処理を実施する。
109:            }
110:        } catch (NamingException e) {
                (省略) // 再試行のために，一定時間待つ。
111:        }
112:    }
113:    if (!idPair.childChecked) {
114:        return AccountLink.ERROR_LINK_FAILED;
115:    }
```

図4　Web-PのWebアプリにおけるAccountLinkのメソッドを呼んでいる部分

```
116:     try {
117:         idPair.makeLink();          // アカウントの紐付けを実施する。
118:     } catch (NamingException e) {
             (省略) // 例外処理
119:     }
```

図4　Web-P の Web アプリにおける AccountLink のメソッドを呼んでいる部分（続き）

J 主任は，情報処理安全確保支援士（登録セキスペ）の K 主任とともにコードレビューを実施した。K 主任は，次のような特定の状況では，サイト P のある利用者のアカウントに，サイト Q の他人のアカウントが紐付いてしまうと指摘した。

(1)　図 2 で，サイト Q の利用者 ID 欄に誤った利用者 ID を入力する。

(2)　図 4 のコードが呼び出され，[c] に誤った利用者 ID が代入されたまま，[d] が呼び出される。

(3)　何らかの理由で，[d] 中の図 3 中の行番号 [e] で [f] が発生し，結果として [d] が再試行される。

(4)　[d] が 3 回試行され，全て同じく [f] が発生すると，[g] の値は true なので，図 4 中の行番号 [h] に進み，紐付けが行われる。

K 主任は，図 3 の 32 行目前後に着目し，[i] という修正案を提示した。

〔サイト R でのインシデント〕

リリースに向けて準備を進めていたところ，インシデント発生の報告があった。発端は，ある顧客からの指摘で，その内容は，“サイト R のキャンペーン応募履歴を見たら，3 月のキャンペーンに応募したことになっているが，身に覚えがない。3 月にはサイト R に一度もアクセスしていないはずだ。”というものであった。

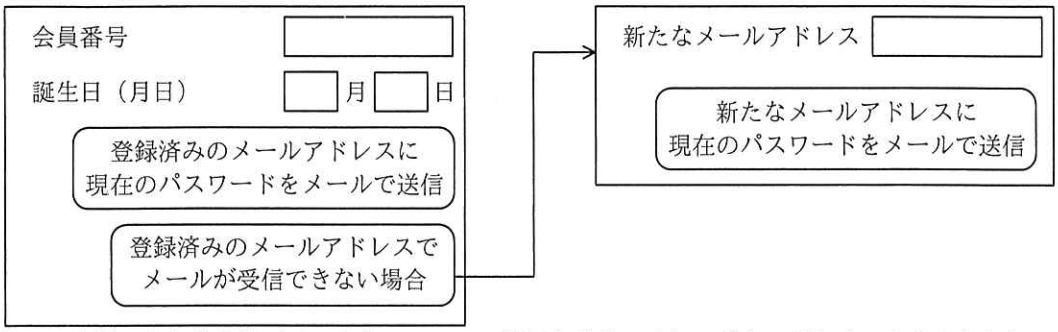
J 主任が，サイト R のアクセスログを確認したところ，次のことが分かった。

・3 月 30 日に，当該顧客のアカウントでキャンペーンに応募していた。

・応募の直前にパスワード失念時の処理を実行し，パスワードを電子メール（以下，メールという）で未登録のメールアドレスに送信した記録がある。

・3 月 25 日から 3 月 31 日に掛けて，サイト R への通信量が増加傾向にあった。

図５は，サイトＲのパスワード失念時の操作画面である。



注記　実線の矢印は当該画面の入力フォームに適切な値を入力してボタンがクリックされたときの遷移を示す。

図５　サイトＲのパスワード失念時の操作画面

Ｊ主任は，攻撃者がパスワード失念時の処理を悪用して，会員番号及び誕生日を総当たりで入力し，たまたま合致した当該顧客のアカウントを乗っ取ったものと判断した。Ｊ主任は，このインシデントについてWeb管理課のＬ課長に報告した。Ｌ課長は旧Ａ社出身で，この報告でサイトＲのパスワード失念時の操作を初めて知った。次は，その報告の時のＬ課長とＪ主任の会話である。

Ｌ課長：サイトＲのパスワード失念時の処理には，三つの問題があるね。一つ目は，本人であることを確認するための情報が少なすぎるという問題だ。そこは後で解決するとしよう。二つ目は，パスワードそのものをメールで送るという問題だ。三つ目は，［　j　］という問題だ。二つ目と三つ目の問題の解決には，［　k　］ように改修すべきだ。この方法では，一部の利用者はパスワード失念時にログインできなくなるが，その場合はコールセンタで対応することにしよう。

Ｊ主任　：はい。分かりました。

Ｌ課長：サイトＲでは，攻撃者がアカウントを乗っ取ったとしても，あまり経済的利益を得られないので，今回のような被害で済んだと考えられるが，直ちに改修を完了させてほしい。もしもこれらの問題に気付かずにアカウントの共通利用を提供していたら，①利用者に更に大きな被害が発生するところ

だった。アカウントの共通利用の設計，及びリリースまでのスケジュールも見直してほしい。

〔アカウントの共通利用の設計の見直し〕

J主任は，次のように全面的に設計を見直し，承認を得た。

・サイトSを立ち上げ，新たにサイトSのアカウントを発行して管理する。

・アカウントの共通利用に当たり，アカウントの紐付け時とサイト P，Q，R へのログイン時には，SAML プロトコルの Web Browser SSO Profile を用いる。サイト S が，IdP（Identity Provider）となり，サイト P，Q，R は，SP（Service Provider）となる。

・サイトQには，サイトQのアカウントでも，サイトQのアカウントと紐付けたサイトSのアカウントでもログインできる。サイトP，Rも同様である。

図6は，Web Browser SSO Profile の基本的な通信の流れである。

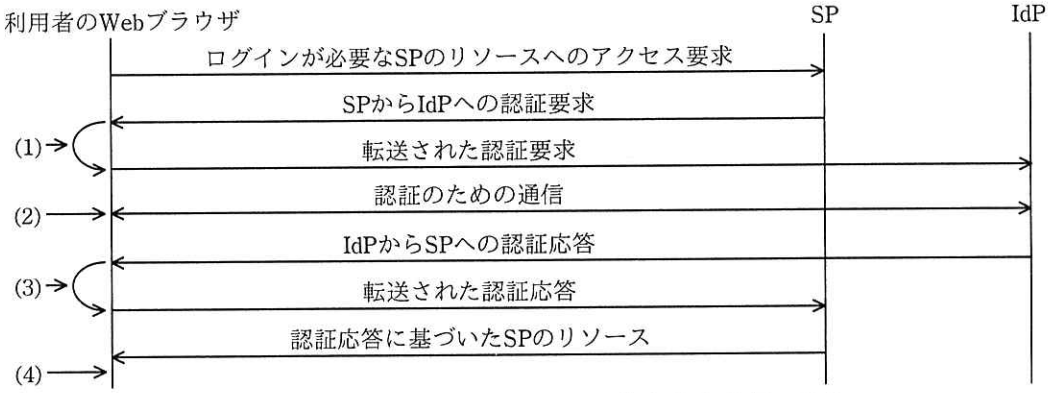
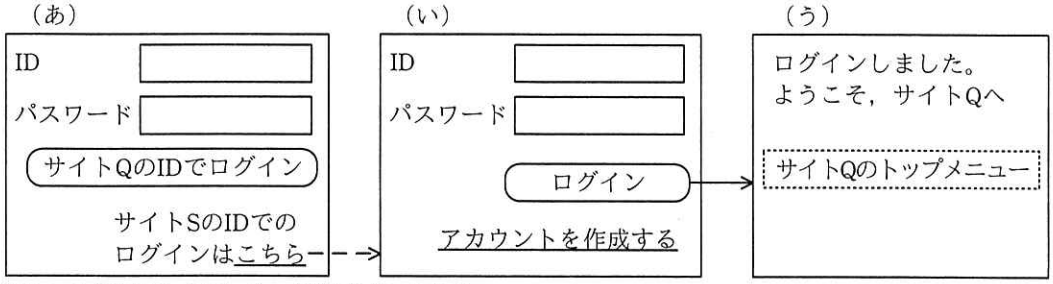


図6　Web Browser SSO Profile の基本的な通信の流れ

図7は，紐付け済みのサイトSのアカウントでサイトQにログインするときの画面遷移図である。

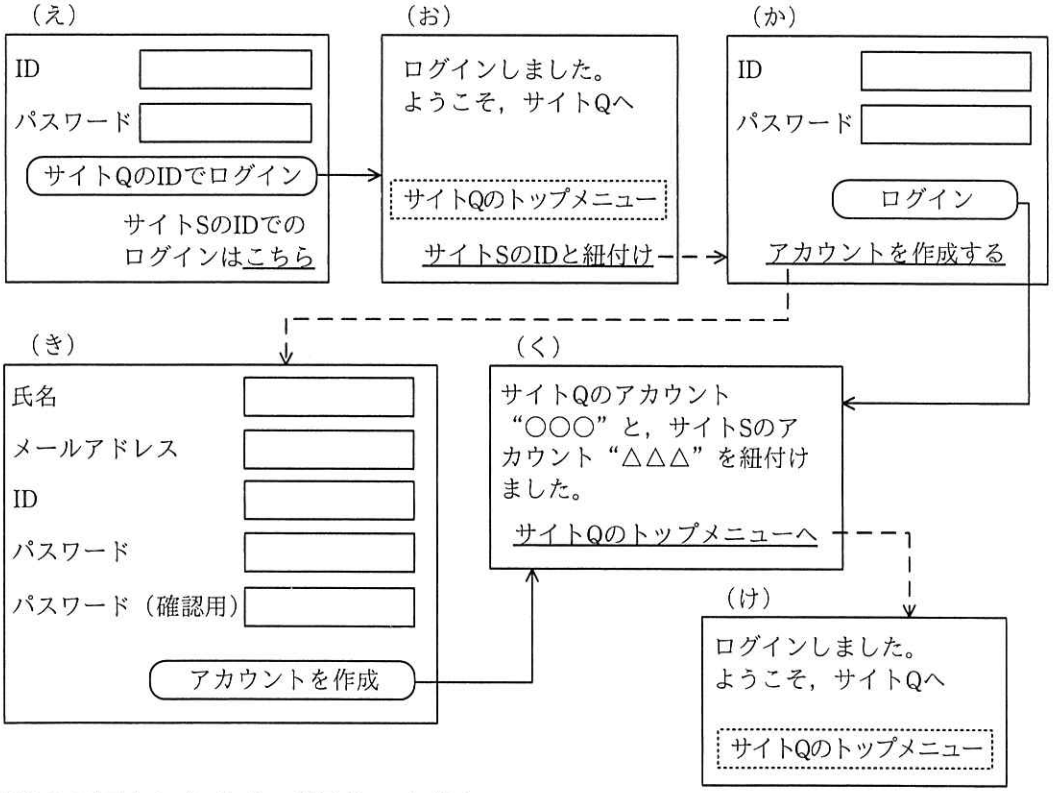


注記 1 画面内の“ID”は，利用者 ID を指す。

注記 2 実線の矢印は当該画面の入力フォームに適切な値を入力してボタンがクリックされたときの遷移を示す。破線の矢印はリンクがクリックされたときの遷移を示す。

図 7　サイト S のアカウントでサイト Q にログインするときの画面遷移図

図 8 は，サイト Q のアカウントとサイト S のアカウントを紐付けるときの画面遷移図である。



注記 1 画面内の“ID”は，利用者 ID を指す。

注記 2 実線の矢印は当該画面の入力フォームに適切な値を入力してボタンがクリックされたときの遷移を示す。破線の矢印はリンクがクリックされたときの遷移を示す。

図 8　サイト Q のアカウントとサイト S のアカウントを紐付けるときの画面遷移図

〔見直し後のアカウント共通利用の提供開始〕

C 社は，サイト P，Q，R の改修とサイト S の開発を完了して，アカウント共通利用の提供を開始し，順調に運用を続けた。サイト P，Q，R のアカウントの廃止，及びサイト S のアカウントへの統合を目指している。この統合が実現すれば，図 7 の（あ）や，図 8 の（え），（お）などの画面は不要となり，より使いやすいサイトを実現できる。

設問 1　表 2 中の [a]，[b] に入れる適切な字句を答えよ。

設問 2　〔個人情報の取扱い〕について，旧 A 社と旧 B 社の合併による C 社への事業承継に伴って取得した個人情報の取扱いに関し，個人情報保護法に定められている禁止事項は何か。70 字以内で述べよ。

設問 3　〔コードレビュー〕について，(1)～(3)に答えよ。

(1)　本文中の [c]，[d]，[f]，[g] 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　idPair.checkChild()	イ　idPair.childChecked
ウ　idPair.childID	エ　idPair.childPW
オ　idPair.childSite	カ　idPair.makeLink()
キ　idPair.parentID	ク　idPair.siteQAuth()
ケ　NamingException	コ　return ERROR_ID_OR_PW
サ　return NO_ERROR	

(2)　本文中の [e]，[h] に入れる適切な図 3 又は図 4 中の行番号を，解答群の中から選び，記号で答えよ。

解答群

ア　20	イ　23	ウ　26	エ　34
オ　105	カ　106	キ　117	ク　118

(3)　本文中の [i] に入れる適切な処理内容を 50 字以内で具体的に述べよ。

設問 4　〔サイト R でのインシデント〕について，(1)～(3)に答えよ。

(1)　本文中の [j] に入れる適切な内容を 40 字以内で述べよ。

(2) 本文中の [k] に入れる適切な内容を 40 字以内で述べよ。

(3) 本文中の下線①について，更に大きな被害とは何か。具体的な被害を二つ挙げ，それぞれ 30 字以内で述べよ。

設問5 〔アカウントの共通利用の設計の見直し〕について，(1)～(3)に答えよ。

(1) 利用者の操作によって，図7のとおりに画面が遷移した場合，（い）（う）の画面は，図 6 の(1)～(4)のどの時点で表示されるか。それぞれ，(1)～(4)の記号で答えよ。

(2) 利用者の操作によって，図 7 のとおりに画面が遷移した場合，（あ）（い）（う）の各画面では，どのサーバから送られた HTML を表示するか。それぞれ，答案用紙の“SP”，“IdP”のいずれかを○印で囲んで示せ。

(3) 利用者の操作によって，図 8 の（え）（お）（か）（き）（く）（け）の順に画面が遷移した場合，（え）（か）（き）（く）の各画面では，どのサーバから送られた HTML を表示するか。それぞれ，答案用紙の“SP”，“IdP”のいずれかを○印で囲んで示せ。