問1　脆弱性検査の結果とその後の対策に関する次の記述を読んで，設問 1～3 に答えよ。

A 社は，衣料品を取り扱う会員制インターネット通販会社である。A 社では，セキュリティ専門会社の B 社に依頼して，Web アプリケーションの脆弱性検査を，年に 4 回定期的に実施している。検査を実施した結果，会員の属性情報の登録，変更などを行う会員情報管理システム（以下，P システムという）において，指摘事項が 2 件報告された。

A 社情報システム部の X 部長は，指摘事項の確認と対策の検討を，Y 主任に指示した。

〔脆弱性検査の結果〕

Y 主任は，早速，B 社の検査担当者から，指摘事項 2 件についての詳細な説明を受けた。

指摘事項 A：画面の遷移の中で，暗号化通信と非暗号化通信が混在しているが，暗号化通信でだけ使用されるべきクッキーに，［　a　］属性が設定されていないページが存在する。

指摘事項 B：任意のスクリプトが実行可能であるページが存在する。

説明に当たって B 社からは，指摘事項 A を検出したときのログイン操作直後の HTTP レスポンス（図 1），指摘事項 B を検出したときに使用された HTTP リクエスト（図 2）と，図 2 の HTTP リクエストに対する HTTP レスポンス（図 3）が示された。検査においてフォーム認証に使用した会員 ID とパスワードは，検査のために用意されたものである。

```
HTTP/1.1 200 OK
Date: Sun, 01 Apr 2012 12:00:00 GMT
Set-Cookie: JSESSIONID=0C3FE1B62D5B5A5DE5A06E5D5C23F6F9
（以下，省略）
```

図 1　指摘事項 A を検出したときのログイン操作直後の HTTP レスポンス

```
GET /mypage/progA?PID=xx&QTY=yy"><script>alert("script")</script> HTTP/1.1
Host: www.a-sha.co.jp
Cookie: JSESSIONID=0C3FE1B62D5B5A5DE5A06E5D5C23F6F9
Referer: http://www.a-sha.co.jp/prog0
（以下，省略）
```

図２　指摘事項Ｂを検出したときに使用されたHTTPリクエスト

```
HTTP/1.1 200 OK
（中略）
<form name="form" method="post" action="/mypage/progB">
   <table>
   <tr>
     <th>商品ID</th>
     <td><input type="text" name="PID" value="xx" ></td>
     <th>個数</th>
     <td>
       <input type="text" name="QTY" value="yy"><script>alert("script")</script>">
     </td>
   </tr>
   </table>
（以下，省略）
```

図３　図２のHTTPリクエストに対するHTTPレスポンス

Ｙ主任は，提示された図1～3の内容を手掛かりに，対象のプログラムのソースコードを調査し，指摘事項Ａと指摘事項ＢがＰシステムにおいて，脆弱性といえるものであることを確認した。

〔脆弱性の影響の検討〕

続いてＹ主任は，脆弱性の影響について検討を行った。まず，今回の指摘事項Ａと指摘事項Ｂに対して，どのような攻撃があり得るか，また，その攻撃を受けた場合，どのような被害が予想されるかについて検討を行った。検討結果の抜粋は，表１のとおりである。

表１　予想される攻撃と被害（抜粋）

指摘事項	予想される攻撃	予想される被害
A	クッキーに含まれるセッション ID が，非暗号化通信時に盗聴される。	盗聴されたセッション ID が使用され，セッションが乗っ取られる。
B	HTTP リクエストに挿入されたスクリプトが，ブラウザ上で実行されることで，クッキー内のセッション ID が窃取される。	窃取されたセッション ID が使用され，セッションが乗っ取られる。

さらに，Y 主任は，これまでに表 1 の攻撃及び被害が実際に起きているかどうかを，ログから確認することにした。

指摘事項 A で予想される攻撃である，セッション ID の盗聴については，ログから検出することは困難である。次善の策として，盗聴されたセッション ID が不正に使用されたことを検出するために，同一のセッション ID が，複数の送信元 IP アドレスから送信されているリクエストをログから抽出することにした。この方法では，①正当なアクセスを誤って検出してしまう場合と，②不正なアクセスを検出できない場合があるが，それらの場合については容認することにした。

指摘事項 B に対する攻撃は，③HTTP アクセスログから検出することにした。ただし，この方法では［　b　］メソッドが使用された場合にしか検出できない。

以上の方法で過去のログを確認したところ，不正使用や攻撃は検出されなかった。

〔改修方法の検討〕

Y 主任は，脆弱性への対応を行うために，まず，指摘事項 B に該当する複数のプログラムのソースコードを再度確認した。該当するプログラムの一つを図 4 に示す。

```
01: import java.io.*;
02: import java.util.*;
03: import javax.servlet.*;
04: import javax.servlet.http.*;
05:
06: public class SearchWord extends HttpServlet {
07:    public void doGet(HttpServletRequest request, HttpServletResponse response)
08:    throws IOException, ServletException {
09:
10:       response.setContentType("text/html;charset=UTF-8");
11:       PrintWriter out = response.getWriter();
12:
13:       String word = request.getParameter("word");
14:       String category = request.getParameter("category");
```

図４　指摘事項Ｂに該当するプログラム

```
15:
16:       out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML4.01 Transitional//EN\">");
17:       out.println("<html>");
18:       out.println("<head>");
19:       out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset= utf
    -8\">");
20:       out.println("<title>Search</title>");
21:       out.println("</head>");
22:       out.println("<body>");
23:       out.println("<h3>Search</h3>");
24:       out.println("<form action=\"searchWord\" method=\"GET\">");
25:       out.println("word<br>");
26:       out.println("<input type=\"text\" size=\"10\" name=\"word\">");
27:       out.println("<br>");
28:       out.println("category(option)<br>");
29:       out.println("<input type=\"text\" size=\"10\" name=\"category\">");
30:       out.println("<br>");
31:       out.println("<input type=\"submit\">");
32:       out.println("</form>");
33:
34:       out.println("<br>");
35:       out.println("- - results - - category: " + escape(category) + "<br>");
36:
37:       out.println("<a name=\"#\" onclick= \"alert('" + escape(word) + "')\">");
38:       out.println("Previous search word" );
39:       out.println("</a>");
40:
41:       if (word != null) {
42:         searchResult(out, word, category);
43:       }
44:       out.println("</body>");
45:       out.println("</html>");
46:    }
47:
48:    public void doPost(HttpServletRequest request, HttpServletResponse response)
49:    throws IOException, ServletException {
50:       doGet(request, response);
51:    }
52:
53:    private static String escape(String message) {
54:       //  この関数は，message に含まれる文字を次のように置換した String 型の文字列を返す。
55:       //  「&」→「&amp;」  「<」→「&lt;」  「>」→「&gt;」  「"」→「&quot;」
56:       (省略)
57:    }
58:
59:    private void searchResult(PrintWriter out, String word, String category) {
60:       //  検索を実行し，結果を表示する。
61:       (省略)
62:    }
63: }
```

図４　指摘事項Ｂに該当するプログラム（続き）

このプログラムは，利用者が入力した文字列をダイアログに表示するために，受け取ったパラメタの値をスクリプトに埋め込み，動的にスクリプトを生成する。図 4 の [c] 行目では，通常の HTML にパラメタの値を埋め込むときと同じ方法で，エスケープ処理を行っていたことから，生成されるスクリプトに問題が生じてしまうことが分かった。

Y 主任は，以上の検討結果から，指摘事項 A と指摘事項 B に対する改修方針を，表 2 のとおりにまとめた。

表 2　指摘事項に対する改修方針

指摘事項	改修方針
A	暗号化通信でだけ使用するクッキーに [a] 属性を設定する。
B	出力するときの文脈に合わせたエスケープ処理を行う。

特に，表 2 中の指摘事項 B の改修方針を実現するために，図 4 の [c] 行目の escape 関数呼出しを，新たに作成する escape2 関数（図 5）呼出しに変更することにした。escape2 関数では，意図したスクリプトが生成されるように，まず，スクリプト言語の文法上必要なエスケープ処理を行った後，更に escape 関数でエスケープ処理を行う。

```
private static String escape2(String message) {
  //  この関数は，message に含まれる文字を次のように置換した String 型の文字列を生成し，
  //  その文字列を escape 関数に渡し，その結果の String 型の文字列を返す。
  //  「 d 」→「 e 」  「 f 」→「 g 」
  (省略)
}
```

図 5　新たに作成する escape2 関数

その後，Y 主任は，X 部長の承認を得た上で，改修作業を行い，脆弱性検査結果についての対策を完了した。

設問1　本文中と表2中の［　a　］と，本文中の［　b　］に入れる適切な字句を，それぞれ8字以内で答えよ。

設問2　〔脆弱性の影響の検討〕について，(1)～(3)に答えよ。

(1)　本文中の下線①について，正当なアクセスを誤って検出してしまう場合とはどのような場合か。40字以内で具体的に述べよ。

(2)　本文中の下線②について，不正なアクセスを検出できない場合とはどのような場合か。40字以内で具体的に述べよ。

(3)　本文中の下線③について，どのような条件のログを検出すればよいか。35字以内で具体的に述べよ。

設問3　〔改修方法の検討〕について，(1)，(2)に答えよ。

(1)　本文中の［　c　］に入れる適切な行番号を一つ数字で答えよ。

(2)　図5中の［　d　］～［　g　］について，［　d　］，［　f　］には置換の対象文字を，［　e　］，［　g　］には置換後の文字列を，それぞれ答えよ。