平成20年度　春期
テクニカルエンジニア（情報セキュリティ）
午後Ⅱ　問題

試験時間	14:10 ～ 16:10（2時間）

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. この注意事項は，問題冊子の**裏表紙**に続きます。必ず読んでください。
4. **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。**
5. 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
選択方法	1問選択

6. 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) B又はHBの黒鉛筆又はシャープペンシルを使用してください。
   (2) **受験番号欄**に，**受験番号**を記入してください。正しく記入されていない場合は，採点されません。
   (3) **生年月日欄**に，受験票に印字されているとおりの**生年月日**を記入してください。正しく記入されていない場合は，採点されないことがあります。
   (4) **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**を○印で囲んでください。
   なお，○印がない場合は，採点の対象になりません。2問とも○印で囲んだ場合は，はじめの1問について採点します。
   (5) 解答は，問題番号ごとに指定された枠内に記入してください。
   (6) 解答は，丁寧な字ではっきりと書いてください。読みにくい場合は，減点の対象になります。

〔問2を選択した場合の例〕

選択欄
問1
(問2)

注意事項は問題冊子の裏表紙に続きます。
こちら側から裏返して，必ず読んでください。

問1　Webアプリケーションシステムの脆弱性対策に関する次の記述を読んで，設問1～3に答えよ。

A社は従業員数500名の卸売会社で，メーカから事務用品を仕入れ，小売店向けに販売している。A社の情報システム部は，企画開発課と運用課で構成されている。企画開発課は受発注にかかる時間と経費を削減するために，商品の受発注処理を電子化したXシステムを3年前に自社開発し，運用課がXシステムの運用を担当している。

〔Xシステムの概要〕

図1にXシステムの構成を示す。Xシステムの利用者は，メーカの受注担当者，小売店の発注担当者及びA社の受発注担当者で，インターネットからアクセスし，事務用品の受発注を行う。ログインの際は，利用者IDとパスワードを入力する。入力されたパスワードからはそのハッシュ値（以下，パスワードハッシュという）が算出される。利用者の認証は，利用者IDとパスワードハッシュを，それぞれ表1の利用者テーブル（以下，利用者TBLという）のUSR_ID，PASSWORDと照合して行われる。利用者に実行が許可されている機能は，利用者IDごとに定義されており，［　a　］制御されている。

なお，Xシステムの各サーバのOSはUNIX，通信プロトコルはHTTP over SSLである。

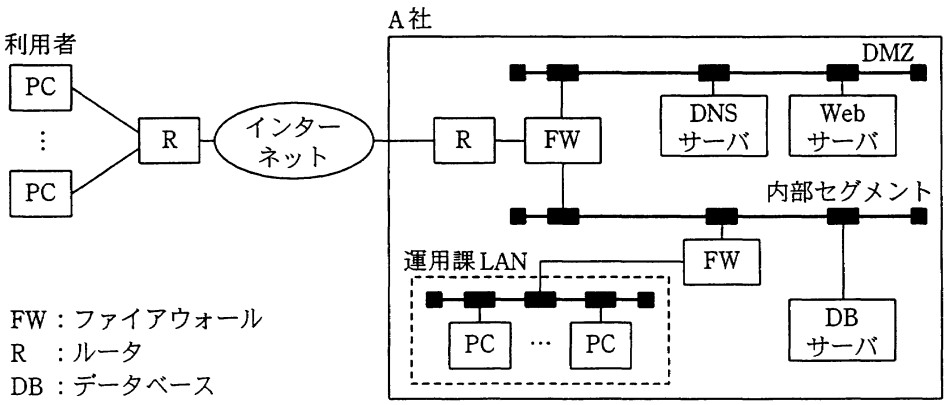

図1　Xシステムの構成

表1　利用者 TBL の仕様

テーブル名：M_USER

列名	データ型（バイト数）	内容	備考
USR_ID	CHAR（12）	利用者 ID	主キー
PASSWORD	CHAR（32）	パスワードハッシュ	
USR_NM	VARCHAR（40）	利用者氏名	
USR_ADDR	VARCHAR（60）	メールアドレス	
COM_ID	CHAR（8）	会社 ID	業種区分＋連番で構成
⋮	⋮	⋮	⋮

〔インシデント発覚〕

休日明けの 2007 年 10 月 15 日（月），X システムの多数の利用者から，これまで利用できていたパスワードでログインできなくなったという苦情が利用者問合せ窓口に殺到した。運用課の窓口担当者である D 君は重大なトラブルの可能性を考慮し，運用課の C 課長に状況を報告した。その後，C 課長は企画開発課の F 課長に原因の調査と対策の検討を依頼した。F 課長は，部下の E 君に調査を命じた。E 君はまず，認証情報が格納されている利用者 TBL に異常がないかどうかを調査した。

次は，E 君が F 課長に調査結果を報告した際の会話である。

E 君　：それでは今回の調査結果を報告します。結論から申しますと，利用者 TBL のすべての行の PASSWORD が，同じ値に変更されていました。これは推測ですが，インターネットからの不正アクセスか，運用担当者による SQL コマンドの発行ミスが原因だと思われます。念のため，C 課長に問い合わせたところ，ここ数日間は利用者 TBL を変更するような処理を実施していないとのことでした。一方，管理者の認証情報は，利用者 TBL ではなく別のテーブルに格納されていたので，管理者機能は現在でも利用できます。

F 課長：君の推測はログから確認できるのかな。

E 君　：いいえ。DB サーバのアクセスログ（以下，DB ログという）が収集されていないので確認はできません。さらに，運用担当者であれば C 課長の許可なく利用者 TBL を含むすべての DB にアクセスできる状況であることと，DB にログインするための DBMS の利用者 ID（以下，DB アカウントという）を運用課の 4 名の担当者で共用していることも分かりました。これらの運用状況

は改善する必要がありますね。

F課長：そのとおりだ。しかし，DB ログを収集していなくても，DB のトランザクションログ（以下，TRN ログという）を調べれば，更新時刻やどの DB アカウントによる更新なのかが分かるはずだ。

E君　：当社には TRN ログを本格的に解析できる者がいません。

F課長：そうか。では，TRN ログの解析や，不正アクセスの可能性についての調査を，セキュリティ専門会社に依頼することにしよう。

F課長は，セキュリティ専門会社の B 社に調査を依頼した。B 社では G 氏をリーダとして調査することにした。

〔調査前の B 社による X システムの概要確認〕

本格的な調査を開始する前に，F課長と G 氏によって，図 1～3，表 1，2 などの資料を使って，X システムの概要確認が行われた。次は，そのときの会話である。

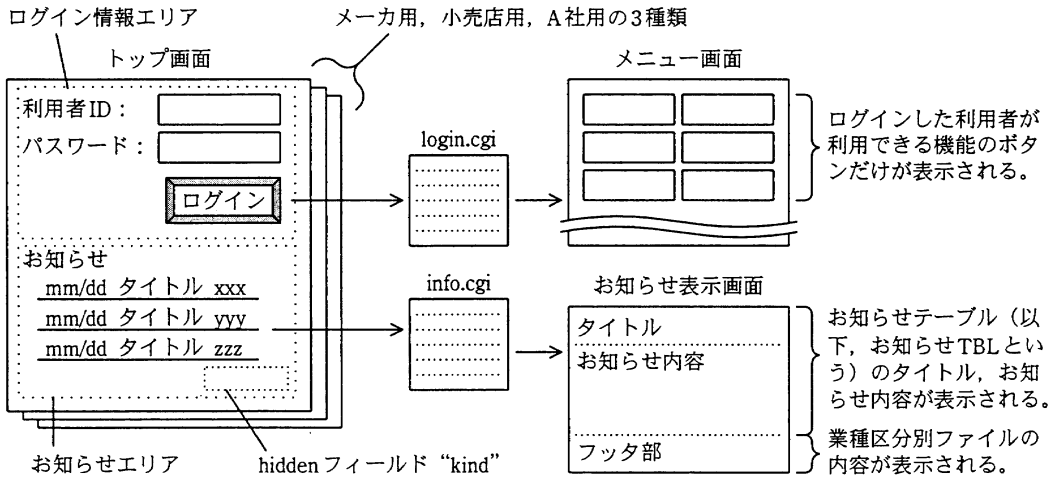

図2　X システムの入出力画面イメージ（抜粋）

F課長：まず，X システムの入出力画面について説明します。トップ画面は，メーカ用，小売店用，A 社用の業種区分別に 3 種類あり，それぞれ別々の URL で閲覧するようにしてあります。しかし，それらのトップ画面から呼び出される CGI プログラム login.cgi，info.cgi は共通です。

G氏　：お知らせ表示の機能（以下，お知らせ機能という）は，ログインしていなくても利用できるようにしているのですね。

F課長：はい。年に数回実施するDBのメンテナンスやそのほかの作業で，Xシステムにログインできないときに，利用者に通知する必要があるからです。

G氏　：なるほど。それから，お知らせ表示画面では，タイトルとお知らせ内容はお知らせTBLから取得し，フッタ部は業種区分別ファイルの内容を表示しているのですね。

表2　お知らせTBLの仕様

テーブル名：M_NEWS

列名	データ型（バイト数）	内容	備考
NWS_ID	CHAR（4）	お知らせID	主キー
NWS_NM	VARCHAR（100）	タイトル	
NWS_BODY	VARCHAR（4,000）	お知らせ内容	
⋮	⋮	⋮	⋮

F課長：はい。そのために，それぞれのトップ画面のお知らせエリアのhiddenフィールド“kind”（以下，kindという）には，業種区分を表す文字列がセットされており，お知らせ機能は，ログインしていなくても利用者の業種区分を識別できます。

（中略）

G氏　：利用しているDBMSに対して特殊な設定をしていませんか。

F課長：標準のDBMS初期設定値のままで，特殊な設定はしていません。

DBMS仕様

・DB管理テーブル（以下，DB管理TBLという）の仕様
0）共通事項
・DB管理TBLは，DBMSによって管理され，管理対象の属性が変更された場合は自動的に更新される。
・DB管理TBLに対するアクセス権限は，DBアカウント別に付与，削除が可能である。
1）DBアカウント情報管理テーブル（テーブル名はADMIN_ALL_USERS）
・DBMS内のすべてのDBアカウントのアカウント名（USER_NAME），パスワード文字列のハッシュ値（USER_PASSWORD）などのDBアカウント情報が管理されている。
2）テーブル情報管理テーブル（テーブル名はADMIN_ALL_TABLES）
・DBMS内のすべてのテーブルのテーブル名（TABLE_NAME），スキーマ（SCHEMA）などのテーブル情報が管理されている。

図3　XシステムでDBMSの仕様（抜粋）

G氏　：DB サーバにアクセスできるのは，Web サーバと運用課の専用 PC だけですか。

F課長：はい。DB サーバ側でアクセス元の IP アドレスを制限しています。

G氏　：運用課の C 課長は，利用者 TBL を変更するような処理はしていないと言っているのですね。社内の者が PASSWORD を改ざんしてもメリットはないので，まずはインターネットからの不正アクセスの可能性について調査します。10 月 13 日（土）の Web サーバのアクセスログ（以下，Web ログという）を見せていただけますか。

F課長：はい，すぐに用意します。

〔原因の判明とインシデント対応〕

G氏は，F課長が用意した Web ログを調査した。次は，そのときの G氏と F課長の会話である。

G氏　：思ったとおり，SQL インジェクションによる攻撃を受けていました。図 4 をご覧ください。左から順に，空白で区切って，リクエスト日，リクエスト時刻，クライアントの IP アドレス，HTTP のアクセスメソッド，パス名，クエリストリング，ステータスコードの各項目が並んでいます。どうやら PASSWORD の改ざんだけでなく，USR_NM と USR_ADDR からも大量にデータが盗まれた可能性があります。

このような不正アクセス時の対応手順は策定されていますか。

```
#Fields: date time client-ip method pathname query-string status
（省略）
2007-10-13 02:24 aa.cc.kk.hh GET /mlogin.html - 200
2007-10-13 02:25 aa.cc.kk.hh GET /scripts/info.cgi "?nid=0701&kind=m" 200
2007-10-13 02:29 aa.cc.kk.hh GET /scripts/info.cgi "?nid=0701&kind=../etc/passwd"
 200
（省略）
2007-10-13 03:00 aa.cc.kk.hh GET /scripts/info.cgi "?nid=0701&kind=ls |" 200
2007-10-13 03:13 ff.aa.rr.pp GET /slogin.html - 200
2007-10-13 03:14 dd.ee.ss.tt GET /slogin.html - 200
2007-10-13 03:15 dd.ee.ss.tt GET /scripts/info.cgi "?nid=0701&kind=m" 200
2007-10-13 03:16 dd.ee.ss.tt GET /scripts/info.cgi "?nid=gf34' union select USER_
NAME,USER_PASSWORD from ADMIN_ALL_USERS where（省略） &kind=m" 200
（省略）
2007-10-13 03:17 dd.ee.ss.ll GET /scripts/info.cgi "?nid=as23' union select TABLE
_NAME,SCHEMA from ADMIN_ALL_TABLES where（省略） &kind=m" 200
2007-10-13 03:18 dd.ee.ii.jj GET /scripts/info.cgi "?nid=0701';update M_NEWS set
PASSWORD='fas7wgw4&kind=m" 200
2007-10-13 04:01 aa.cc.kk.hh GET /scripts/info.cgi "?nid=0701&kind=../ls |" 200
2007-10-13 04:17 dd.ee.ii.jj GET /scripts/info.cgi "?nid=0703' union（省略） &kind=
m" 200
2007-10-13 04:28 dd.ee.ii.jj GET /scripts/info.cgi "?nid=0701';update M_USER set
PASSWORD='fas7wgw4&kind=m" 200
```

注　Fields:に続く各項目は，ログの項目名を示す。
　　time の秒の部分は省略されている。
　　（省略）は，ログ出力内容の省略を意味する。
　　query-string は，二重引用符で囲まれ，URL デコード済の値である。

図 4　2007 年 10 月 13 日（土）の Web ログ（抜粋）

F 課長：対応手順は策定していませんが，PASSWORD が改ざんされ，利用者がログインできない状態でしたので，CIO と相談し，指示に従って X システムを停止しておきました。

G 氏　：良い判断でしたね。そのまま運用を継続していたら被害が拡大したかもしれません。

F 課長：被害拡大は防げましたが，今後はどうすればよいでしょうか。

G 氏　：被害者への連絡，捜査機関への被害届出，場合によっては情報公開や監督官庁への報告をしていただく必要がありますので，証拠保全と正確な情報の把握から始めます。Web ログ，TRN ログに加えて，ソースコードの解析，図 4 で見られた攻撃の再現検証，及び疑似侵入テストを実施し，重要な箇所については詳細な調査を行いたいのですが，よろしいですか。

F 課長：部下の E 君をサポートに付けますので，調査をお願いします。

G 氏　：承知しました。明日の夕方に報告いたします。可能であれば，CIO も同席してくださるようお願いします。

〔調査結果の報告〕

次は，翌日夕方の調査結果の報告におけるG氏とF課長の会話である。

G氏　：それでは今回の調査結果を報告します。表3をご覧ください。Webログから確認できる範囲の最初の攻撃は2007年8月24日（金）2時24分で，それ以降も41回の攻撃を受け，累計で834件，325人分の情報が盗まれていました。お知らせ機能だけが攻撃を受けており，そのほかの機能は無事でした。この機能はログインしなくても利用でき，かつ，脆弱性があったからではないかと推測されます。

表3　攻撃一覧（抜粋）

攻撃の時間帯	攻撃元IPアドレス	被害の概要
2007/08/24 02:24 ～ 2007/08/24 04:41	aa.bb.xx.yy	91件の利用者氏名，メールアドレスの流出
⋮	⋮	⋮
2007/10/13 02:24 ～ 2007/10/13 06:01	aa.cc.kk.hh	OSコマンド実行結果の不正閲覧 想定外ファイルの不正参照
2007/10/13 03:14 ～ 2007/10/13 05:17	dd.ee.ss.tt	①DB管理TBLの不正参照
[redacted]	dd.ee.ii.jj	②パスワードハッシュの改ざん 329件の利用者氏名，メールアドレスの流出
2007/10/14 01:18 ～ 2007/10/14 01:19	aa.bb.xx.zz	24件の利用者氏名，メールアドレスの流出

注　網掛けの部分は，設問の関係上，表示していない。

F課長：USR_NMとUSR_ADDRが流出した可能性があるのですね。

G氏　：はい。図4で見られた攻撃を再現させたところ，union句を利用して利用者TBLのデータを参照できることを確認しました。

G氏は，調査結果の報告を終え，今後の対策について説明を進めた。

〔SQLインジェクション対策〕

次は，お知らせ機能におけるSQLインジェクション対策に関する会話である。

F課長 ：図 2 のお知らせエリアには入力フィールドがないので，SQL インジェクション対策は必要ないと思っていました。

G氏 ：画面上の入力フィールドの有無とセキュアプログラミングの要否は関係ありません。SQL インジェクション対策について言えば，SQL 文の要素になる文字列に対しては，すべて，エスケープ処理や DB のバインド機構の利用が必要です。実際に攻撃を受けていたお知らせ機能の CGI プログラム（図 5）を基にもう少し詳しく説明します。このプログラムは Perl を用いて作成されています。

G氏は，SQL インジェクション対策について説明した。

```
 1 #!/usr/bin/perl
 2 use CGI;                                  # 汎用 CGI 処理用モジュール CGI の使用を宣言
 3 use DBI;                                  # 汎用 DB 処理用モジュール DBI の使用を宣言
 4 $DOC_ROOT = "/home/info/";                        # お知らせファイルディレクトリ
 5 $cgi = new CGI;                                   # CGI クラスのオブジェクトを生成
 6 $nws_id = $cgi->param('nid');                     # nid パラメタを取得
 7 $kind = $cgi->param('kind');                      # kind パラメタを取得
 8 print "Content-Type: text/html¥n¥n";              # HTTP リプライヘッダを出力
 9 print "<HTML><BODY>";                            # HTML 開始部分を出力
10
11 $dbh = DBI->connect('DBI（省略）') or die;       # DB に接続
12 $sql = "select NWS_NM,NWS_BODY from M_NEWS ";# SQL 文組立てを開始
13 $whr = "where NWS_ID = '";
14 $qry = $sql.$whr.$nws_id."';";                   # SQL 文組立てを終了
15 $sth = $dbh ->prepare($qry) or die;              # SQL 文を準備
16 $sth->execute;                                   # SQL 文を実行
17
18 @ary = $sth ->fetchrow_array;                    # 先頭行を取得
19 print xss($ary[0])."¥n<HR>¥n";                   # お知らせタイトル，仕切り線を出力
20 print xss($ary[1])."¥n<HR>¥n<PRE>";              # お知らせ内容，仕切り線を出力
21 $sth->finish or die;                             # SQL 文使用を終了
22 $dbh->disconnect or die;                         # DB から切断
23
24 $filepath = $DOC_ROOT.$kind;                     # 業種区分別ファイルへのパスを決定
25 open(FILE, $filepath) or die;                    # ファイルを開く
26 while($line=<FILE>) {
27     print xss($line);                            # ファイル内容をエスケープして出力
28 }
29 close(FILE);                                    # ファイルを閉じる
30
31 print "</PRE></BODY></HTML>";                   # HTML 終了部分を出力
（省略）
42 sub xss {（省略）}    # 引数の文字列中の&, <, >, ", 'に対するクロスサイトスクリプティング
43                       # 対策処理をして返す
```

注 （省略）は，記述を省略していることを意味する。

図5 お知らせ機能の CGI プログラム“info.cgi”（抜粋）

〔そのほかの脆弱性とその対策〕

G氏は，テスト環境でXシステムに対して疑似侵入テストを実施した際に，SQLインジェクション以外にも幾つかの脆弱性を発見した。次は，それらの脆弱性についての会話である。

G氏　：まず，表3中の下線①について再現できるか検証したところ，各プログラムからDBへのアクセスに利用されているDBアカウントで，“ADMIN_ALL_USERS”や“ADMIN_ALL_TABLES”のデータを参照できました。DB管理TBLに対する参照権限は本当に必要でしょうか。

F課長：いいえ，必要ではありません。それが何か問題なのでしょうか。

G氏　：プログラムの修正漏れなどによって，万一脆弱性が残ってしまった場合に備えて，被害を最小にするためにDB側で，［　b　］するのが定石です。
次に，そのほかの脆弱性とその対策に移ります。Xシステムでは利用者ごとに許可されている機能が定義され，［　a　］制御されているはずですが，一部の機能では，実行用のURLを直接入力することによって，テスト用アカウントには許可されていないはずの機能を実行できました。

F課長：ログイン後のメニュー画面には，利用者に許可されている機能のボタンだけを表示していたのですが，実行用のURLを直接入力することで実行できてしまったということですね。

G氏　：はい。ログインした利用者が各機能を実行してよいか否かをサーバ側で確認する［　a　］制御を，すべての機能に対して必ず実装してください。

F課長：分かりました。全機能を確認し，修正します。

G氏　：また，ログアウト後にブラウザの“戻る”ボタンを押すことで，再びログインしなくても，それまで利用していたページを表示させ，Xシステムの機能を実行することができました。これを防ぐために，ログアウト処理で［　c　］をしてください。

F課長：分かりました。ログアウト処理の部分を修正します。

引き続き，G氏はそのほかの脆弱性についても報告した。

〔インシデント対応策の検討〕

検出されたすべての脆弱性についての報告が終了した。次は，インシデント対応策についての会話である。

F課長：お蔭様でインシデント対応策のめどがつきました。あとは当社でプログラムの修正を実施すれば再開できますね。

G氏　：いいえ。全プログラムを対象に調査を実施したわけではないので，まだすべての脆弱性を発見できたわけではありません。費用や期間の問題もありますが，できれば全プログラムを対象にセキュリティ検査を実施した方がよいので，ご検討ください。

F課長：分かりました。ただ，X システムの場合，ほとんどの機能はログインしてから利用できるようになっており，利用者は取引先の受注，発注担当者と当社の受発注担当者だけですので，今回脆弱性を指摘された部分を中心に再検査をすればよいと思っています。それにしても今回は，Web ログをきちんと取っておいたことは不幸中の幸いでした。

G氏　：そうですね。しかし，攻撃者が GET ではなく POST で攻撃してきたら，図4の項目の［　d　］に有益な情報が残らないので，漏えいした情報を確認することはできなかったでしょう。Web アプリケーションの場合，ログの種類としては，主に Web ログ，TRN ログ，DB ログがありますが，③各ログ単独では，いつ，だれが，何を，どうしたかの情報が不足しています。

G 氏は，その理由について説明した。続いて，運用課におけるシステム運用の問題点と対策についても説明した。

G氏　：最後に，長期的な観点から，X システムにおける同様の不正アクセスに対する防止策の検討と，万一発生した場合の対応策を用意しておく必要があります。防止策については，先ほど申し上げたような対策を実施する必要があります。発生時に備える対応策としては，不正アクセスをできる限り早く検知するための対策や，④検知後の対応が迅速かつ正確に行えるようにする対策を実施しておく必要があります。

F課長 ：今回はたまたま初動対応がうまくいきましたが，だれが対応しても同じようにできなくてはいけませんからね。

G氏 ：そのとおりです。

G氏は，インシデント対応策について更に具体的に説明した。

その後，捜査機関によって攻撃者が逮捕された。幸いにも，盗まれた情報が二次利用されたとの報告はなかった。また，流出した個人情報にかかわる利用者に対する事情説明や情報公開なども適切に行われた。

脆弱性が発見されたプログラムは修正され，B社によるセキュリティ検査を実施した上でリリースされた。そして，各利用者に新たなパスワードを発行して，Xシステムは，安全にサービスを再開することができた。

設問1 脆弱性対策について，(1)～(3)に答えよ。

(1) 本文中の［ a ］，［ c ］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 エラーログの採取　　イ サニタイジング　　ウ 識別

エ セッション情報の破棄　　オ 認可　　カ 認証

(2) 図5のプログラムに対して，コマンドifconfigの実行結果を表示させる攻撃があった。攻撃時にkindに指定された文字列を，20字以内で答えよ。ここで，コマンドifconfigの絶対パスは，/bin/ifconfigであるものとする。

(3) 図5のプログラムにおいて，25行目のopen関数をsysopen関数に変更した場合，A社が受けていたどの攻撃が防げるか。20字以内で答えよ。また，open関数に起因する問題点のうち，sysopen関数に変更しただけでは解決できない問題点を，30字以内で述べよ。

設問2 SQLインジェクション対策について，(1)～(4)に答えよ。

(1) 表3中の下線②が実行された日付及び時刻を答えよ。

(2) 本文中の［ d ］に入れる適切な字句を，12字以内で答えよ。

(3) 図5のプログラムにおいて，DBMSの特殊文字をエスケープ処理するSQLイ

ンジェクション対策用サブルーチンを利用するとした場合，このサブルーチンを適用すべき最も適切な変数はどれか。該当するものを解答群の中から選び，記号で答えよ。

解答群

ア　`$cgi`　　イ　`$nws_id`　　ウ　`$qry`　　エ　`$ary[0]`

(4) 本文中の［　b　］に入れる対策内容を，50字以内で述べよ。

設問3　DBサーバに対する運用上の問題とインシデント対応策について，(1)～(3)に答えよ。

(1) DBサーバに対する運用課における運用上の問題点を二つ挙げ，それぞれ30字以内で述べよ。

(2) 本文中の下線③でG氏が言及している，TRNログ単独及びDBログ単独では不足している情報は何か。各ログについて該当するものを解答群の中からすべて選び，記号で答えよ。また，DBログに不足している情報を補うためにはXシステムをどのように改修すべきか。50字以内で述べよ。

解答群

ア　いつ（実行日時）　　イ　だれが（利用者ID）

ウ　何を（どのテーブル）　　エ　どうした（参照，更新，削除）

(3) 本文中の下線④でG氏が言及している，実施すべき対策を二つ挙げ，それぞれ25字以内で述べよ。

問2　認証システムの企画・設計に関する次の記述を読んで，設問1～5に答えよ。

L 学園は，傘下に情報専門学校，ビジネス専門学校，建築・インテリア専門学校の三つの専門学校をもつ学校法人である。3校の学生数は合わせて3,000名，学校法人本部（以下，本部という）の職員を含めて教職員数は150名である。表1に示すL学園の情報システムは，本部の情報システム課で開発管理を行い，本部の人事課と各校の学生課などの各主管部署で運用を行っている。このほか，本部にはWebサーバがあり，各校の一般向けWebサイトと学生向けWebサイトをインターネット経由で公開している。

表1　L学園の情報システム

種別	システム名	クライアントソフト	利用者	サーバ名（設置場所）	現行の認証方式
事務系	人事給与システム	Webブラウザ	教職員	人事給与サーバ（本部）	Webサーバによるフォーム認証
	財務会計システム	専用クライアントソフト	職員	財務会計サーバ（本部）	専用サーバソフトによる認証
教務系	学籍管理システム	Webブラウザ	教職員	教務系サーバ（本部）	Webサーバによるフォーム認証
	成績管理システム	Webブラウザ	教職員，学生		
	就職情報システム	専用クライアントソフト	教職員	就職情報サーバ（各校）	専用サーバソフトによる認証
	学生実習環境（Web閲覧を含む）	市販ソフト，Webブラウザ	教職員，学生	PC実習サーバ（各校）	ディレクトリサーバによる認証
				UNIX実習サーバ（各校）	UNIX OSによる認証
共通	ファイルサーバ	（OSの標準機能）	教職員	ファイルサーバ（本部，各校）	ディレクトリサーバによる認証
	グループウェア	専用クライアントソフト	教職員	グループウェアサーバ（本部，各校）	グループウェアサーバによる認証
	電子メール	メールソフト	教職員，学生	メールサーバ（本部，各校）	IMAP4による認証

学園全体でのシステム化が進むに従って，学生や教職員が利用する利用者IDとパスワードの数は増加する一方である。事務系，教務系の各情報システムでは主管部署が個別に利用者IDとパスワードを発行しており，学生の入学や卒業，教職員の異動などのたびにアカウントの追加や削除を行っている。各校では授業で利用するノートPCを一括購入して新入学生に配布していることもあって，年度末から新年度にかけては各部署での業務上の負担が特に大きくなってきている。

また，各利用者に対して情報システムごとに利用者 ID が付与されることが多くなったことから，複数校の授業を兼任する教員などからは，複数の利用者 ID とパスワードを覚えるのが難しいという声が頻繁に聞かれるようになってきている。一方，一部の情報システムでは職員個人には利用者 ID が割り当てられておらず，数人の職員が一つの利用者 ID を共用しているケースもある。

こうした状態はパスワードや情報の漏えいにつながりかねないとの経営陣からの指摘や，将来の想定される用途を踏まえ，情報システム課は，安全な利用者認証を実現する手段や，利用者認証の新たな用途への応用について検討することにした。

次は，情報システム課の P 課長とその部下の Q さんの会話である。

〔学生向け Web サイトの認証方式〕

P 課長：まず，公開している Web サーバについて考えていくことにしよう。現在，一般向けに公開している各校の Web サイトには利用者認証が必要なページはないが，各校の学生向け Web サイトには認証が必要なページがあるね。

Q さん：各校の学生向け Web サイトでは，入学の際に利用者 ID と初期パスワードを学生に通知し，最初のアクセス時に初期パスワードを変更してもらっています。Web サーバでは HTTP でベーシック認証を行い，学生はシラバス（講義要領）などの資料や休講連絡のページにアクセスしています。

P 課長：今は SSL を使用していないから，HTTP 上でのベーシック認証ではパスワードが漏えいする可能性があるのが一番の問題だね。これは一般的に Web サイトでよく使われるフォーム認証でも同様だ。

ベーシック認証よりも安全な方法として知られているものには，［ a ］認証がある。この方法では，ハッシュ関数を用いたチャレンジ・レスポンス方式を採用することでパスワードの漏えいを防ぐことができる。しかし，古いブラウザなどでは対応していないという問題がある。

Q さん：学生は携帯電話からアクセスすることが多いのですが，携帯電話でも［ a ］認証には対応していないものがあると聞いたことがあります。

P 課長：今はあまり重要な情報を提供してはいないが，将来は学生向けポータルサイトとして履修登録や成績の確認，学生の出欠管理もできるようにしたいという要求が各校から出ている。

Q さん：そうなると，Web サイト上で重要な個人情報を取り扱うことになりますから，やはり①［ a ］認証を採用するだけでは不十分ですね。

P 課長：そうだね。学生向けポータルサイトには SSL の導入が必要になりそうだ。

Q さん：SSL を導入する場合には外部のサーバ証明書発行サービスを利用することになると思いますが，携帯電話への対応はどうしましょうか。

P 課長：利用者側で SSL に対応した携帯電話を利用するのはもちろんだが，携帯電話会社や機種の違いも考慮する必要がある。多くの機種に対応したサイトを構築するには，サーバ側で適切な配慮が必要になりそうだね。

〔情報システムの認証方式〕

P 課長：次に，現状での情報システムの問題点について見ていこう。

Q さん：教職員からは，個別の利用者 ID とパスワードが情報システムごとに割り振られているので使いにくいという話をよく聞きます。

P 課長：各情報システムによって構築された時期が違うし，利用形態や稼働している OS もそれぞれ違っているからね。そもそも財務会計システムなどでは，職員一人一人ではなく係単位でしか利用者 ID が割り振られていないから，セキュリティ上大きな問題がある。

Q さん：どういったところが問題なのでしょうか。

P 課長：ID は英語で identifier（識別子）というくらいだから利用者の識別が主な目的ではあるが，もう一つ大事な目的として［ b ］制御がある。つまり，どの利用者に，どのリソースに対して，どのようなアクセスを許可するのかということだ。個別に利用者 ID を割り当てないと，権限の管理が適切に行えないおそれがある。また，最近は内部統制の観点から［ c ］性という要素も重視されるようになってきている。

Q さん：事務系も教務系も，利用者 ID とパスワードの管理が煩雑になっているのが一番の問題ですね。シングルサインオン（以下，SSO という）を実現して，一度の利用者認証で複数の情報システムを利用できるようにするのがよいのではないでしょうか。

P 課長：いや，それよりも現状では各情報システムの利用者の登録や削除に伴う運用コストの方が問題だ。まずは，個々の利用者に対して学園内の各情報システ

ム共通に利用できる利用者 ID を発行したい。これを仮に共通 ID と呼ぶことにしよう。その上で，この共通 ID を含む認証情報と，利用者の氏名や所属などの属性情報を併せて一元管理するシステムを作るのがよさそうだ。これは仮に共通 ID システムと呼ぶことにしよう。②個々の情報システムがこの共通 ID システムの提供する認証情報を利用して認証を行えるようにすれば，最初に利用者 ID の移行に伴う作業が発生するが，各情報システムの主管部署での運用コストは低減できる。その次の段階ですべての情報システムを SSO に統合するのがよいだろう。その際，すべての情報システムを一度に SSO に対応させることが難しいなら，一部の情報システムから対応させていってもいい。

Q さん：すべての情報システムが SSO に統合されると，共通 ID システムももちろんですが，SSO 機能を提供するサーバの可用性も重要になってきますね。

P 課長：それに加えて，利用者側での利用者 ID やパスワードの管理の問題もある。先日ビジネス専門学校に打合せに行った時，利用者 ID とパスワードが書かれた付せん紙が PC のディスプレイにはってあったのを見つけたよ。このような実態では SSO を導入するとかえって危険だから，パスワードに代わる認証方式を導入したいところだ。

Q さん：生体認証や IC カードなどの認証トークンを検討してはどうでしょうか。どんなデバイスを選ぶかにもよりますが，学生の出席確認や入退室管理など，将来いろいろな用途が想定されます。

P 課長：そうだね。ただ，当学園の情報システムには，クライアントとして Web ブラウザを使うものと専用クライアントソフトを使うものが混在しているから，認証トークンを導入すると SSO への対応が難しくなるかもしれない。更に言うと，認証トークンを使うかどうかにかかわらず，現行の情報システムはできるだけ改修せずに SSO への対応を図りたいところだね。

Q さん：そうですね。ちょっと欲張りな要件のような気もしますが，ベンダの提案を聞いてみることにしましょう。

〔無線 LAN と利用者認証〕

P 課長：ネットワークの構成に関しても，様々な要望があるね。学生からは，インタ

ーネットに接続可能な無線 LAN アクセスポイント（以下，AP という）を食堂やラウンジに設置してほしいという要望が出ている。実習スペースのデスクトップ PC や校内の情報コンセントからでもインターネットにアクセスできるが，数年前から学生に配布しているノート PC には無線 LAN の機能があるからね。

Q さん：複数校を兼任してノート PC を持ち歩いている教員からは，どの学校からでも同じように無線 LAN を使って成績管理システムにアクセスしたいという声を聞きました。個人情報を扱うことになるので十分な対策が必要だと思いますが，無線 LAN でよく使われる［ d ］フィルタリングによる接続の制限や，WEP による暗号化だけで十分なのでしょうか。

P 課長：無線 LAN のセキュリティ対策には，利用者認証と暗号化方式の二つの側面がある。利用者認証の面から言うと，知識のある人なら［ d ］は偽装することができるから十分な対策とはいえない。さらに，各校の AP に利用者のノート PC の［ d ］をすべて登録する必要があるから，管理上の手間もかかる。また，暗号化方式の面では，WEP では利用者が同じ暗号鍵を共用する上，解読が比較的容易だという脆弱性が知られている。

Q さん：利用者認証と暗号化方式の強化を考えなければいけないのですね。利用者認証の強化にはどのような方法があるのですか。

P 課長：利用者認証には，IEEE 802.1X という規格を採用するのがよさそうだ。これと暗号化方式の強化を組み合わせることでセキュリティの強化を図ることができるだろう。

Q さん：そういえば，私が駅やファストフード店でときどき使う公衆無線 LAN の AP でも IEEE 802.1X を利用していたと思います。IEEE 802.1X では，利用者の認証はどのように行うのですか。

P 課長：IEEE 802.1X では，利用者の認証情報は AP ではなく認証サーバと呼ばれる別のサーバに格納されており，AP はこの認証サーバにアクセスの可否を問い合わせる。そのプロトコルとしては［ e ］を利用する実装が多いようだ。

Q さん：以前はダイヤルアップのアクセスサーバなどでよく使われていたプロトコルですね。各校に認証サーバを設置して共通 ID システムと連携させれば，各

校の間で共通に無線 LAN が利用できるわけですね。

P 課長：そういうことだ。IEEE 802.1X で使われる EAP（Extensible Authentication Protocol）というプロトコルは PPP を拡張したものだから，［ e ］とは親和性が高い。EAP では，まず端末と AP の間で認証を行い，認証が完了した時点で端末に個別のセッション鍵を配送する。セッション鍵をもっていない未認証の端末は，AP を超えて LAN 側にパケットを送出することができない仕組みになっている。これに加え，EAP では各種の認証方式を利用することができる。代表的な認証方式を挙げると，表 2 のようになる。

表 2　EAP の代表的な認証方式の比較

認証方式	クライアント認証	サーバ認証	セッション鍵の自動生成	ノート PC の OS での対応
EAP-MD5	利用者 ID/パスワード	なし	なし	現在は実装なし
EAP-TLS	ディジタル証明書	あり	あり	標準対応
EAP-TTLS	利用者 ID/パスワード	あり	あり	追加ソフトが必要
PEAP	利用者 ID/パスワード	あり	あり	標準対応

Q さん：サーバ認証というのは，クライアント認証とは逆に，クライアントが認証サーバを認証するためのものですね。

P 課長：そのとおり。EAP-MD5 は③サーバ認証の仕組みをもたないのでリスクがあるし，学生に配布しているノート PC の OS でのサポートも打ち切られている。クライアント認証について言うと，EAP-TTLS では OS で標準的にサポートされていないので，追加ソフトが必要になる。したがって，共通 ID システムに公開鍵基盤（PKI）を導入するなら EAP-TLS を，そうでなければ PEAP を使うことになるだろう。

Q さん：セッション鍵の自動生成というのは，暗号化方式の強化に関係するものですね。

P 課長：そうそう，暗号化方式を強化するという話がまだだったね。IEEE 802.1X では，利用者ごとに異なった WEP 鍵が使用されることになっている。これに加え，セッション中も自動的に WEP 鍵を更新することで，暗号化された通信の解読をより困難にするというわけだ。

無線 LAN の暗号化方式に関しては，近年になってより安全な規格が提案されている。WEP の後継規格である WPA と IEEE 802.11i（WPA2）という規格では，利用者認証に IEEE 802.1X を採用した上で，更に暗号化方式の強化を図っている。WPA では，暗号化には WEP と同じ［ f ］を使うものの，TKIP という鍵交換方式の採用によって暗号鍵を動的に変更することができる。また，WPA2 では，暗号化に［ g ］を用いて暗号自体の強度を高めることができる。

Q さん：なるほど，暗号化方式の強化にもいろいろな方法があるのですね。

〔共通認証システム〕

以上の検討を基に，P 課長と Q さんは共通 ID システム及びそれを利用して利用者認証を行うシステム（以下，両システムを併せて共通認証システムという）の要件をまとめ，図に示す提案依頼書をベンダ各社に提示した。

この提案依頼書に対し，数社のベンダから提案書が寄せられた。その採否を検討する過程で，L 学園は Z 社に対してヒアリングを行うことになった。

Z 社の提案によると，利用者である個々の学生や教職員の認証情報と属性情報は，L 学園本部に新たに設置する LDAP サーバにすべて格納される。このサーバが個々の情報システムに利用者の認証情報や属性情報を配信することで，情報システムが利用者認証を行うことができるようになっている。また，利用者認証には PKI を利用することとし，公開鍵証明書と秘密鍵は，学生証又は教職員証を兼ねた IC カードに格納して利用者本人に配布することとしている。

共通認証システムに関する提案依頼書

1. L学園の情報システムの現状とネットワーク構成
（省略）
2. 共通認証システムの目的
当学園に在籍する学生と教職員を対象とした共通認証システムを新たに構築することによって，セキュリティを保ちつつ［ア］ことと［イ］ことを目的とする。また，教務系の各情報システムでの学生情報の利用などへの活用も目的とする。
3. 共通認証システムの概要及びその要求仕様
3.1 本部及び各校のすべての情報システム利用者に対する共通的な認証基盤である共通IDシステムの構築
(1) 共通IDシステムでは，利用者に対して本部及び各校で共通の利用者IDを付与すること
(2) 共通IDシステムでは，利用者の認証情報（共通IDを含む）及び属性情報（氏名，所属など）を一元管理し，当学園の情報システムに対して配布できること
(3) 共通IDシステムを利用した利用者認証には④二要素認証を採用すること。ただし，学生向けWebサイトでの認証を除く
(4) 共通IDシステムは，業務のサービスレベルの観点から可用性に配慮すること
3.2 主要な情報システムにおけるシングルサインオン（SSO）機能の導入
(1) 利用者が複数の業務アプリケーションにアクセスする際，それぞれにログインし直す必要がないよう，表1の情報システムに対してSSO機能を提供すること
(2) SSO機能には，共通IDシステムが提供する認証情報を用いること
(3) SSO機能を提供するサーバは，業務のサービスレベルの観点から⑤可用性に配慮すること
(4) SSO機能の提供に当たっては，既存の情報システムの改修を極力少なくすること
3.3 学生向けWebサイトのSSL化
(1) 各校の学生向けWebサイトをSSLに対応させること
(2) 携帯電話からのSSLによるアクセスに対応させるために，サーバ証明書はすべての携帯電話会社の主要な機種に組み込まれたルート証明書によって検証可能であること
3.4 各校における無線LANアクセスポイントの設置
(1) 各校に無線LANアクセスポイント及び認証サーバを設置し，IEEE 802.1Xによる利用者認証を行うこと
(2) 各校に設置されたIEEE 802.1Xの認証サーバと共通IDシステムの連携を図り，利用者が3校間で同様に無線LANを利用できるようにすること
(3) EAPの認証には，クライアントとサーバ間の相互認証が可能な方式を利用すること
(4) 通信の暗号化方式としてWPA又はWPA2を採用すること
（以下省略）

図　提案依頼書

次は，P課長，Qさんと Z社の担当者X氏との会話である。

〔SSO機能の実現方式〕

P課長：SSO機能の実現方式についてお聞かせください。

X氏　：今回はWebのSSO機能と専用クライアントソフトのSSO機能を両立できるシステムを提案しています。

Webにおいて SSOを実現させる場合には，クッキーにセッションIDなどの情報を格納することで，認証された利用者に対するセッションを維持するのが一般的です。しかし，貴学園の場合は⑥本部と各校がそれぞれ別個のドメ

イン名を使用しているので，本部と各校をまたがって，クッキーを利用してセッションを維持することができません。

P課長：そうですか。ドメイン名のことまでは考えが及びませんでした。

X氏　：そこで，現在のドメイン名を変更することなく複数のドメインにまたがってSSOを実現するために，今回の提案ではリバースプロキシという方式を採用した製品を選択しました。

この方式では，利用者とWebサーバの間にリバースプロキシサーバと呼ばれるサーバを設置します。利用者からのアクセスは，このリバースプロキシサーバが認証を行った上で各Webサーバに中継することによってSSOが実現できます。新たに機器を設置することになるので，ネットワークの構成やWebアプリケーションへのアクセスの方法は変わりますが，Webアプリケーション自体の改修は基本的には不要です。

Qさん：専用クライアントソフトでのSSO機能についてはどのように考えていますか。

X氏　：利用者のPCにエージェントモジュールをインストールすることで，SSO機能を導入したいと思います。専用クライアントソフトの起動時に，エージェントモジュールが認証情報の入力を自動的に代行することでSSOを実現することができます。今回提案するSSOシステムのエージェントモジュールは，今お使いになっている専用クライアントソフトにすべて対応しています。

〔PKIによる利用者認証〕

P課長：次に，共通IDシステムを利用した利用者認証の方式についてお聞かせください。

X氏　：利用者認証には幾つかの方式が考えられます。ワンタイムパスワードや生体認証も考えられますが，PKIであれば公開鍵暗号を利用してお互いの公開鍵証明書を検証することができるので，より確実な認証方法となります。この方式であれば，共通IDシステムとIEEE 802.1Xの認証サーバをLDAPで連携させることで，無線LANのクライアント認証も実現可能です。

P課長：SSOや無線LANの認証にPKIを利用するとなると，ディジタル証明書と秘密鍵を格納するデバイスをどうするかという問題が出てきますね。御社でIC

カードを提案されたのは，どのような理由からでしょうか。

X氏　：認証用デバイスとしては，USB トークンや IC カードを利用することが考えられます。どちらもデバイスの機能によってディジタル証明書と秘密鍵が保護されており，PIN を入力することで二要素認証が実現できます。

一般的には，それぞれ表 3 のようなメリットとデメリットがあります。これらの特徴に加え，⑦将来の想定される用途を踏まえて両者を比較した結果，IC カードを採用すべきであると判断しました。

表3　認証用デバイスのメリットとデメリット

デバイス / 評価	USB トークン	IC カード
メリット	・機器の USB ポートに挿すだけで使える。 ・デバイス自体に鍵のようなイメージがあり，利用者にとってなじみやすい。	・配布方法が既に確立されている学生証や教職員証を兼ねることができる。 ・USB トークンよりも安価である。 ・用途に応じて接触方式又は非接触方式を選択できる。
デメリット	・IC カードよりも高価である。 ・USB ポートのある機器でしか利用できない。	・機器に IC カード読取り機が必要である。

Q さん：新学期が始まる前には IC カードを大量に発行する必要がありますが，どのような方法で対応されますか。

X氏　：IC カードの発行は外部に委託し，各校で本人確認を行った上で配布していただくことを提案しています。

Q さん：IC カードの発行を外部に委託できるのであれば，年度末の負担もそれほど増えずに済みますね。

L 学園は更に検討を進めた結果，Z 社の提案を採用し，共通認証システムの構築を完了することができた。

設問1　学生向けWebサイトの認証の手法について，(1)，(2)に答えよ。

(1) 本文中の［　a　］に入れる適切な字句を答えよ。

(2) 本文中の下線①について，Qさんが不十分としている理由を，25字以内で述べよ。

設問2　情報システムの認証方式について，(1)～(3)に答えよ。

(1) 本文中の［　b　］，［　c　］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　効率	イ　真正	ウ　信頼
エ　責任追跡	オ　認可	カ　認証

(2) 本文中の下線②について，個々の情報システムで行っていた利用者IDの発行を共通IDシステムに移行するとき，個々の情報システムの主管部署と情報システム課の間で必要となる調整事項を，30字以内で述べよ。

(3) 図中の下線④は，共通IDシステムを利用してSSOを実現する際に考慮すべき，どのようなリスクの低減を意図したものか。60字以内で述べよ。

設問3　SSOの実現方式について，(1)，(2)に答えよ。

(1) 図中の下線⑤は，SSO機能を提供するサーバにおいてどのようなリスクの低減を意図したものか。60字以内で述べよ。

(2) 本文中の下線⑥について，クッキーを利用したセッションの維持ができない理由を，クッキーの動作原理に基づいて40字以内で述べよ。

設問4　無線LANと利用者認証の方式について，(1)，(2)に答えよ。

(1) 本文中の［　d　］～［　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　AES	イ　CHAP	ウ　ESSID	エ　IPアドレス
オ　MACアドレス	カ　RADIUS	キ　RC4	ク　RC5
ケ　SASL	コ　URL	サ　トリプルDES	シ　パケット

(2) 本文中の下線③について，無線LANのサーバ認証によってどのような脅威を防ぐことができるか。また，その脅威が防げない場合には，どのような直接の被害が発生する可能性があるか。それぞれ30字以内で述べよ。

設問5　共通認証システムの設計について，(1)，(2) に答えよ。

(1) 現状での情報システムの問題点に関するP課長とQさんの会話を踏まえて，図中の［　ア　］，［　イ　］に入れる共通認証システムの目的を，それぞれ15字以内で述べよ。

(2) 本文中の下線⑦の観点から，USB トークンと IC カードを比較した結果，X氏がICカードを採用すべきであると判断した理由を，30字以内で述べよ。

プログラム言語 Perl の用例・解説

Perl を使用した問題では，各問題文中に注記がない限り，次に示す用例に従って記述する。

なお，用例は，解答で使用する演算子，関数，予約語などを制限するものではない。

種類	用例 解説

1. 注釈

#	#ここにコメントを書く 行末までが注釈となる。

2. リテラル

スカラ	123 10 進数 123 である。
	12.3 10 進数 12.3 である。
	4E-5 10 進数 4×10^{-5} である。
	0x9f 16 進数 9F である。
	0147 8 進数 147 である。
	0b010111 2 進数 010111 である。
	$var = "hello"; print '$var ', "$var ", `echo world`; 変数 var に文字列“hello”を代入する。文字列のスカラ '$var ', "$var ", `echo world` を出力する。"$var " は変数を展開し，`echo world` はコマンドの出力を展開するので，出力は“$var hello world”となる。
	\n 制御文字（改行）である。
	\r 制御文字（復帰）である。
	\t 制御文字（水平タブ）である。

リストリテラル	('a', 'b', 'c')
	リスト ('a', 'b', 'c') である。
	('a', 'b', 'c')[0]
	リスト ('a', 'b', 'c') の 1 番目の要素 'a' である。
	()
	空リストである。
	('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie')
	キー a, b, c に，それぞれ値 alpha, bravo, charlie を結び付けたハッシュである。
ファイルハンドル	STDIN
	標準入力である。
	STDOUT
	標準出力である。
	STDERR
	標準エラー出力である。
	ARGV
	コマンドラインから指定されたファイル名のリストを順に読み込むためのファイルハンドルである。

3. 変数

スカラ変数	$var
	スカラ変数 var である。
配列変数	@ary
	配列変数 ary である。
配列要素	$ary[6]
	配列変数 ary の 7 番目の要素である。
ハッシュ変数	%hash
	ハッシュ変数 hash である。
ハッシュ要素	$hash{'a'}
	ハッシュ変数 hash の要素のうち，キー a に結び付けられた値である。
局所的な変数	{my $var;}
	{} 内を有効範囲とする変数 var の宣言である。
$_	$_ = "abc"; if (/b/) print "match";
	パターンマッチの演算子が省略されたとき，$_ の文字列“abc”が // 内のパターン b と一致するかどうかを判定し，“match”が出力される。
@ARGV	@ARGV
	コマンドライン引数のリストを格納する配列変数である。
@_	@_
	サブルーチンに渡す引数のリストを格納する配列変数である。

4. 演算子

->	$object->method1
	オブジェクト object のメソッド method1 を呼び出す。
	Class->method2
	クラス Class のメソッド method2 を呼び出す。
++, --	$a++
	変数 a を評価した後に 1 を加算する。
	--$b
	変数 b から 1 を減算した後に評価する。
!, +（単項), -（単項）	!$a
	変数 a の論理否定である。
	+123
	正の数 123 である。
	-123
	負の数 123 である。
=~, !~	$html_contents =~ /<b>/
	変数 html_contents の値に，文字列“<b>”が含まれているときに真を返す。
	$html_contents !~ / /
	変数 html_contents の値に，文字列“ ”が含まれていないときに真を返す。
*, /, %	314 * 34
	314 と 34 の乗算である。
	6 / 469
	6 を 469 で割る除算である。
	34 % 6
	34 を 6 で割る剰余演算である。
+, -, .	3.14 + 2.72
	3.14 と 2.72 の加算である。
	220 - 8125
	220 から 8125 を引く減算である。
	"IPA"."JITEC"
	文字列“IPA”と“JITEC”の連結である。
<, >, <=, >=, lt, gt, le, ge	1 < 2
	数値 1 と 2 を比較し，演算子の左側が右側より小さいので真を返す。数値の関係演算子には，ほかに >，<=，>= がある。
	"b" lt "a"
	文字列“b”と“a”を比較し，演算子の左側が右側より小さくないので偽を返す。文字列の関係演算子には，ほかに gt，le，ge がある。

==, !=, <=>, eq, ne, cmp	1 <=> 2 数値 1 と 2 を比較し，演算子の左側が右側より大きければ 1，等しければ 0，小さければ −1 を返すので，この場合は −1 を返す。数値の比較演算子には，ほかに ==, != がある。
	"b" cmp "a" 文字列 “b” と “a” を比較し，演算子の左側が右側より大きければ 1，等しければ 0，小さければ −1 を返すので，この場合は 1 を返す。文字列の比較演算子には，ほかに eq，ne がある。
&&	$x >= 0 && $x < 10 変数 x の値が 0 以上かつ 10 未満なら真を返す。
||	$x < 0 || $x >= 10 変数 x の値が 0 未満又は 10 以上なら真を返す。
..	@card = (1 .. 52) 1 から 52 までの連続する整数を配列変数 card に代入する。
=, +=, -=, *=, /=, %=	$a = 1 変数 a に 1 を代入する。
	$a += 10 変数 a の値に 10 を加算して a に代入する。 代入演算子には，ほかに -=, *=, /=, %= がある。
=>, ,	%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie') a に alpha，b に bravo，c に charlie を結び付けたハッシュをハッシュ変数 hash に代入する。
not	not $a 変数 a の論理否定である。
and	$a < 0 and $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理積である。
or, xor	$a < 0 or $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理和である。
	$a < 0 xor $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の排他的論理和である。

注 演算の優先順位は，上表の枠の順である。

5. 文

if	if ($var == 1) {
	  print "a";
	} elsif ($var == 2) {
	  print "b";
	} else {
	  print "c";
	}
	変数 var の値が 1 なら “a” を，2 なら “b” を，それ以外なら “c” を出力する。

while	$i = 1;
	while($i <= 10) {
	  print $i++, "\n";
	}
	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

for	for($i = 1; $i <= 10; $i++){
	  print "$i\n";
	}
	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

foreach	foreach $i (1, 3, 5) {
	  print "$i\n";
	}
	変数 i にリストの各要素 1，3，5 を順に代入し，3 回出力する。

next	for ($i = 1; $i <= 10; $i++) {
	  next if $i % 2;
	  print "$i\n";
	}
	変数 i が 2 で割り切れないとき，ループ本体の next 行より後を実行しないので，偶数を出力する。

6. 正規表現

\	/\.\^\$\[|\+*\?\{\(\)\/\\/ 次の 1 文字そのものを表す。“.^$[|+*?{()/\” と一致する。
.	/www.ipa.go.jp/ 改行文字以外の任意の 1 文字と一致する。“wwwdipa,go@jp” と一致する。
^	/^ab/ 先頭が “ab” である文字列と一致する。“abc” と一致するが，“cab” とは一致しない。
$	/yz$/ 末尾が “yz” である文字列と一致する。“xyz” と一致するが，“yza” とは一致しない。
+	/go+d/ 直前の 1 文字 o の 1 回以上の繰返しと一致する。“god” や “gooood” と一致するが，“gd” とは一致しない。

*	/go*d/
	直前の 1 文字 o の 0 回以上の繰返しと一致する。“gd”, “god” や “gooood” と一致する。
?	/colou?r/
	直前の 1 文字 u の 0 回又は 1 回の出現と一致する。“color” 又は “colour” と一致する。
{m}, {m,n}	/co{2}l/
	直前の 1 文字 o の 2 回の繰返しと一致する。“cool” と一致するが, “col” や “coool” とは一致しない。
	/go{1,3}d/
	直前の 1 文字 o の 1 ～ 3 回の繰返しと一致する。“god” や “good” と一致するが, “gd” や “gooood” とは一致しない。
(…)	/<(h.)>/
	() 内の文字列と一致するパターンを部分パターンとしてまとめる。“<h1>” と一致した場合は “h1” が, “<hr>” と一致した場合は “hr” が, まとめられる。
\1, \2, …	/<(..)><([bp])>JITEC<\/\2><\/\1>/
	左から順に () 内のパターンと一致した文字列が \1, \2, … に割り当てられる。“<h1>JITEC</h1>” と一致するが, “<td>JITEC</p> </td>” とは一致しない。
[…]	/<h[12r]>/
	[] 内で指定した文字 1, 2 又は r のどれか一つと一致する。“<h1>”, “<hr>” と一致するが, “<h3>” や “<HR>” とは一致しない。
	/[^0-9]/
	[] 内で指定した 0 ～ 9 以外の 1 文字と一致する。“a” と一致するが, “3” とは一致しない。
…|…	/<(a href|img src)=/
	| で区切られた “a href” 又は “img src” のどちらか一方と一致する。“<a href=” や “<img src=” と一致するが, “<A HREF=” や “<img height=” とは一致しない。

7. サブルーチン

定義	sub greeting {
	  print "hello Perl\n";
	}

	“hello Perl” を出力するサブルーチン greeting を定義する。
呼出し	subroutine ($arg1, $arg2);
	サブルーチン subroutine を引数 arg1 と arg2 で呼び出す。() を省略して “subroutine $arg1, $arg2;” とする表記もある。
戻り	return -1;
	サブルーチンから抜け出し, 値 −1 を返す。

8. モジュール

use	use CGI; モジュール CGI を 1 度だけ読み込み，利用可能にする。

9. メソッド呼出し

->	$object->method1(arg1); 演算子 -> を使って，オブジェクト object のメソッド method1 を引数 arg1 で実行する。
	Class->method2(arg1, arg2); 演算子 -> を使って，クラス Class のメソッド method2 を引数 arg1 及び arg2 で実行する。

10. 文字列操作関数

chomp	chomp @lines; 配列変数 lines の各要素の末尾にある改行文字を削除する。
eval	eval $exp_str; 変数 exp_str の内容を Perl プログラムとして解釈し実行する。
length	length $long_str; 変数 long_str に格納される文字列の文字数を返す。

11. 配列・ハッシュ操作関数

keys	%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie');
	foreach $key (keys %hash) {
	    print "$key\n";
	}
	ハッシュ変数 hash のキーのリストを取り出し，各キーを出力する。この場合は，“a”，“b”，“c” を順不同に出力する。

shift	$next = shift @queue; 配列変数 queue の先頭要素を取り除いて詰め，取り除いた値を変数 next に代入する。
sort	@pile = sort @jumble; 配列変数 jumble の値を文字列の大小比較によって昇順に整列し，配列変数 pile に代入する。
	@pile = sort {$b <=> $a} @jumble; 配列変数 jumble の値を数値の大小比較に従って降順に整列し，配列変数 pile に代入する。
split	@fields = split ',', $csv; 変数 csv の値をコンマで区切って分割したリストを配列変数 fields に代入する。

12. 検索・置換関数

m/…/ 又は /…/	`$html_contents =~ /<b>/i;` 変数 html_contents の値が，文字列“<b>”又は“<B>”を含んでいるかどうかを判定する。i は，大文字，小文字の区別をしないオプションである。
s/…/…/	`$html_contents =~ s/ /\n/gi;` 変数 html_contents の中の文字列“ ”，“ ”，“ ”又は“ ”を改行文字に置換する。g は，一致したすべての文字列を置換するオプションである。
$`，$&，$'，$1，$2，…	`'The date is 1970-01-23.' =~ /([0-9]{4})-([0-9]{2})-([0-9]{2})/;` `print "String before the date: $`\n";` `print "Date: $&\n";` `print "String after the date: $'\n";` `print "Year: $1\n", "Month: $2\n", "Day: $3\n";` 文字列“The date is 1970-01-23.”に対して，一致した部分の前の文字列，一致した文字列，一致した部分の後ろの文字列をそれぞれ変数 `，&，' に代入する。また，() で囲まれた部分パターンと一致した文字列を，1 番目から順に変数 1，2，3 に代入する。これらを利用し，“String before the date: The date is ”，“Date: 1970-01-23”，“String after the date: .”，“Year: 1970”，“Month: 01”，“Day: 23”の 6 行を出力する。

13. 入出力操作関数

open	`open LOG, '>>cgi.log';` ファイル cgi.log を追記モードで開き，ファイルハンドル LOG に対応付ける。
<*filehandle*>	`$line = <USER_FILE>;` ファイルハンドル USER_FILE から 1 行を読み込んで変数 line に代入する。
<>	`@records = <>;` 標準入力（コマンドライン引数があるときは，コマンドライン引数で指定されたファイル）から順にデータを読み込み，すべての行を配列変数 records に代入する。
print	`print LOG "sync.\n";` ファイルハンドル LOG に対応するファイルに文字列を出力する。
close	`close LOG;` ファイルハンドル LOG に対応するファイルを閉じる。

14. システムインタフェース

die	`open(FILE, 'a_file') or die 'cannot open a_file';` ファイル a_file を開く。開くのに失敗したとき，“cannot open a_file”というメッセージを出力して実行を終了する。
system	`system 'a.out';` コマンド a.out を実行し，コマンドが終了するまで待機する。

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

7. 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	14:50 ～ 16:00

8. **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。
9. 問題冊子の余白などは，適宜利用して構いません。
10. プログラム言語 Perl の用例・解説は，この冊子の末尾を参照してください。
11. 試験中，机上に置けるもの及び使用できるものは，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，黒鉛筆又はシャープペンシル，鉛筆削り，消しゴム，定規，時計（アラームなど時計以外の機能が付いているものは不可），ハンカチ，ティッシュ

    これら以外は机上に置けません。使用もできません。
12. 試験終了後，この問題冊子は持ち帰ることができます。
13. 答案用紙は，いかなる場合でも，すべて提出してください。回収時に提出しない場合は，採点されません。
14. 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。