THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP300X
Internet Computing

Spring 2000

Final Examination
Friday, 26 May 2000
8:30am - 10:40am (2 hours 10 minutes)

This is version 1 of the exam

Instructions

- This is a open book, open notes examination
- Simple calculators are permitted; PDAs and computers are not
- Questions 1-10 are each worth 2 marks
- Questions 11-30 are each worth 4 marks
- There is a further question, question 31, to identify which exam version you are using
- Read each question CAREFULLY before answering
- Answer all the questions on the multiple choice answer sheet, which you need to return for marking
- Clearly write your name, student id and lab section on the answer sheet, and indicate your student id on the sheet using the marking system

Unless stated otherwise,

- Assume that questions apply to the latest versions of software stored on the ITSC machines.
- Assume that the latest version of Netscape Navigator is being used.
- Assume that Perl questions concern Perl version 5 or higher (same as we have used).

[Questions 1 to 10 have three possible answers. Each question is worth 2 marks.]

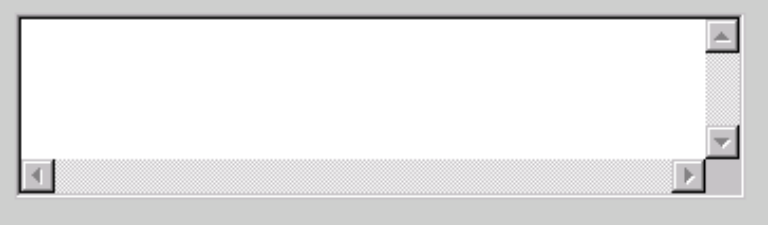
Q1)

Here is some HTML code.

```
<textarea name="specify" rows=3 cols=32></textarea>
```

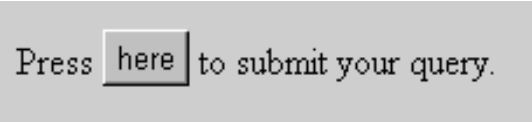
When viewed in a browser, which of the following is most likely to be seen?

a)

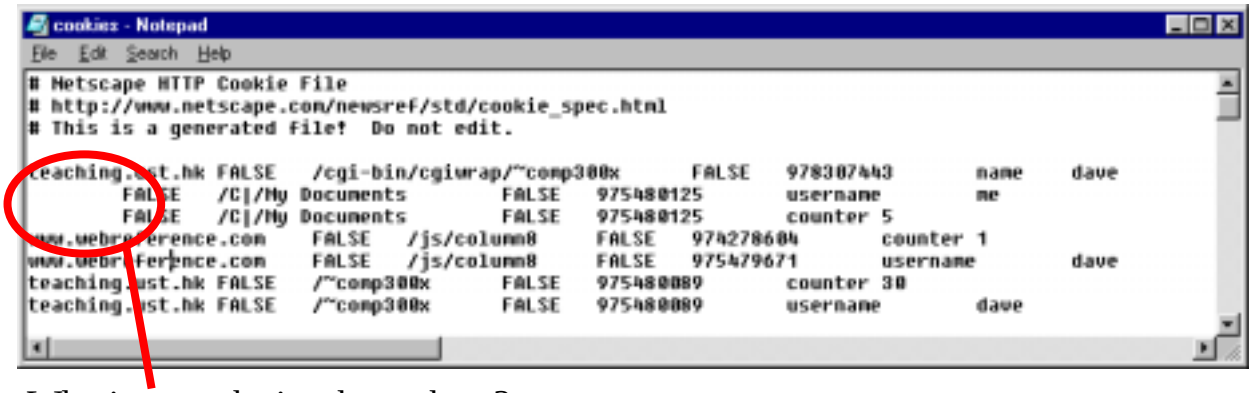


b)

c)



Q2)



Why is no web site shown here?

a) Because the cookie was made by a file run from the local hard disk
b) Because the web site which made the cookie has used encryption
c) Because the user de-selected ‘Enable JavaScript’ in the Preferences section of the browser so that JavaScript cannot be executed

Q3)
Which one of the following ways can be used to show the current time/date on a web page when the web page is loaded by the browser? (The time/date does not have to be continually updated. Assume that JavaScript and SSI work correctly).

- Use JavaScript
- Use an SSI instruction

a) None of the above
b) One of the above
c) Two of the above

Q4)
Which one of the following statements about proxy servers is false?

a) If the file being requested is stored in the proxy server, the user will usually get a faster response than if there was no proxy server
b) If the file being requested is not in the proxy server, the user will get a slower response than if there was no proxy server
c) At UST, some PC machines are configured so that they *have* to use a proxy server, and the user is unable to change this

Q5)
A web site gets 100,000 ‘hits’ in a day. Approximately how big will the log file be?

a) 100,000 bytes
b) 1,000,000 bytes
c) 10,000,000 bytes

Q6)
What is the most popular domain type?

a) .com
b) .org
c) .edu

Q7)
How many of the following statements are *false*?

- On average, more men than women use the internet
- A T1 line is faster than a T3 line
- Internet traffic sent from one Hong Kong ISP to another Hong Kong ISP often goes through the HKIX
- In Hong Kong, there is typically a peak of internet traffic at 3am

a) One of the above is false
b) Two of the above are false
c) Three or four of the above are false

Q8)
How many of the following are completely *true*?

- If you send the Microsoft Word file containing the 300x exam you are currently reading through the internet to another country via email, it is possible that some or all of page 20 will arrive at the destination before any of page 3 arrives. [Assume that the Word file is 500,000 bytes and that datagrams of 1000 bytes are used].
- Someone in Hong Kong wants to see the web site *http://199.2.210.248/* (a web site in the USA) using Netscape by typing in the address and pressing enter in the usual way. The web site has not been visited before. For this to work, the browser must have access to the DNS system. [Ignore caches and proxy servers when considering this].
- An IP address uses eight bytes.

a) One of the above is true
b) Two of the above are true
c) Three of the above are true

Q9)
How many of the following can be used to create a CGI program which, when executed through the internet, creates a cookie in the users machine?

- Perl
- C
- A DOS batch file together with a text file

a) One of the above
b) Two of the above
c) Three of the above

Q10)
Which one of the following has an item which is not related to the other item shown on the same line (or is only very weakly related)?

a) Dreamweaver, DHTML
b) Flash, XHTML
c) Grins, SMIL

[Questions 11 to 30 each have five possible answers. Each question is worth 4 marks]
Q11)
A client program wishes to download a document with URL:

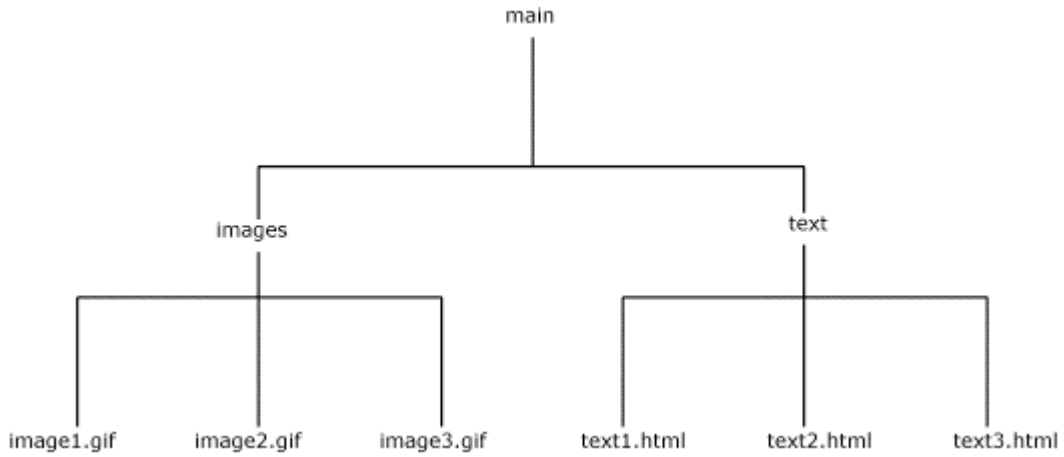
http://www.cs.ust.hk/course/comp300x/index.html

(Assume that the files exists at the appropriate location and that permissions are set appropriately). The sequence of events/ commands is shown below. C means client, S means server.

One line is incorrect. Which one?

a) C: *<establish connection to www.cs.ust.hk on port 80>*
b) C: GET /course/comp300x/index.html HTTP/1.0
c) S: HTTP/1.0 400 OK
d) S: ... *transfer of HTML ...*
e) S: *<closes the connection to the client>*

Q12)
Here is a web site. ‘main’, ‘text’, and ‘images’ are directories.



How can the html file ‘text2.html’ refer to the image file ‘image1.gif’?

a) <IMG SRC="../images/image1.gif">
b) <IMG SRC="image1.gif">
c) <IMG SRC="../image1.gif">
d) <IMG SRC="../text/images/image1.gif">
e) <IMG SRC="../image1.gif/images">

Q13)
An old version of the home page at *http://www.rsac.org/* is viewed using Netscape. The user sees the following.

As you can see, there are three choices, leading to three different web pages (after clicking the relevant area). Which one of the following methods/ techniques can *not* be used to implement this?

a)

```
<a href="howto/index.html">

<img src="images/howto.gif" alt="How to Use Explorer &amp;
RSACi together." border="0" width="195" height="115">

</a> <a href="sponsor.html"><img src="images/register.gif"
alt="Webmasters! Register with RSACi!" border="0"
width="195" height="115">

</a> <a href="faq.html">

<img src="images/faq.gif" alt="RSACi FAQ" border="0"
width="195" height="115"></a>
```

b) Create three layers using DHTML. Use three different images, one for each layer. Add code so that if the user clicks on the image the browser goes to the appropriate web page.

c) Use one single image together with a .MAP file as follows.

```
rect howto/index.html 0,0 194,115
rect sponsor.html 195,0 389,115
rect faq.html 390,0 584,115
```

(assume that 0,0 is the top left hand corner pixel)

d) Use Flash.

e) Use nph.

Q14)
There is a new web format called Scaleable Vector Graphics (SVG). Example code from *http://www.irt.org/articles/js176/* and other pages are shown below. One of the extracts has had a single character (a character that you can actually see - not a space or something similar) removed, and as a result the code would not work at all. Use your knowledge of XML to identify which answer has been modified.

a)
```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG July 1999//EN"
  "http://www.w3.org/Graphics/SVG/svg-19990706.dtd">
<svg width="4in" height="3in">
  <desc>This is a blue circle with a red outline
  </desc>
  <g>
  <circle style="fill: blue; stroke: red"
    cx="200" cy="200" r="100"/>
  </g>
</svg>
```

b)
```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG July 1999//EN"
  "http://www.w3.org/Graphics/SVG/svg-19990706.dtd">
<svg width="4in" height="3in">
  <desc>This is a blue circle with a red outline</desc>
  <g>
  <circle style="fill: blue; stroke: red" cx="200" cy="200" r="100"/>
  <text x=".5in" y="2in">Hello World</text>
  </g>
</svg>
```

c)
```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 03December 1999//EN"
"http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg xml:space="preserve" width="5.5in" height="2in">
<rect style="fill:blue;" width="250" height="100"/>
</svg>
```

d)

```
<?xml version = "1.0" standalone = "yes"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG April 1999//EN"
 "http://www.w3.org/Graphics/SVG/svg-19990412.dtd">
<svg width = "268px" height="207px">
<g style = "stroke: #000000" >
   <path  d = " M 29 28 "/>
   <path  d = " L 19 74 "/>
</g>
<g style = "stroke: #800040" >
   <polyline verts = " 32,100 72,50 90,82 73,16 120,64 152,9 177,107"/>
</g>
<g style = "stroke: #000000" >
</g>
<g style = "stroke: #0000ff" >
   <rect x = "30" y = "101" width  = "51" height = "33"/>
</g>
<g style = "fill: #0000ff" >
   <ellipse cx = "182" cy = "127" major  = "37" minor = "31" angle = "90"/>
</g>
<g style = "fill: #ff0000" >
   <polyline verts = " 78,180 76,151 131,149 136,182 135,182 134,183 127,185
117,186 109,192 104,194 98,199 96,200 95,201 94,202 92,202 85,202 70,200 54,199
47,198 46,197 45,197 37,195 26,193 17,187 9,181 8,181 7,176 6,175 6,173 6,172
6,170 8,164 8,163 8,162 9,162 10,162 11,162 13,162 20,162 26,162 27,162 28,162
30,162 30,163 31,163 32,164 34,166 35,166 36,167 36,168 37,169 38,169 39,169
41,170 43,170 45,170 47,170 49,170 50,168 50,161 50,160 50,159 47,162 78,180"/>
   <g>
     <desc> Java Font definition:Dialog 0</desc>
   </g>
   <g>
     <desc> Java Font definition:Helvetica 0</desc>
   </g>
</g>
<g style = "stroke: #000000" >
   <text x = "188" y = "36" >this is text</text>
</g>
<g style = "stroke: #000000" >
   <g>
     <desc> Java Font definition:Dialog 0</desc>
   </g>
   <g>
     <desc> Java Font definition:Helvetica 700</desc>
   </g>
</g>
<g style = "stroke: #008080" >
   <text x = "176" y = "85" >sadfsadfsad</text>
</g>
<g style = "stroke: #000000" >
</g>
<g style = "fill: #800040" >
   <ellipse cx = "208" cy = "180" major  = "45" minor = "31" angle = "0"/>
</g>
<g style = "stroke: #000000" >
</g>
<g style = "fill: #ffffff" >
   <g>
     <desc> Java Font definition:Dialog 700</desc>
   </g>
   <g>
     <desc> Java Font definition:Dialog 700</desc>
   </g>
</g></svg>
```

e)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 12August 1999//EN"
"http://www.w3.org/Graphics/SVG/SVG-19990812.dtd">

<svg xml:space="preserve" width="1000" height="1000" >
<style type="text/css">
.redbox{fill:#FF0000;}
.whitewords{font-family:Times-Bold;font-size:36;fill:#FFFFFF;}

</style>
<g>
<rect class="redbox" x="10" y="0" width="460" height="50" />

<text class="whitewords" x="20" y="40" >This site is powered by
SVG.</text>
<g>
</svg>
```

15)

One of sections (a) to (d) may have been altered to make it partly or totally incorrect. If you believe one of the sections has been altered, state which one. If you believe all none of (a) to (d) have been altered (all are correct) select (e).

a)

[From *http://www.csclub.uwaterloo.ca/u/mlvanbie/cgisec/callprog.html*]

We will assume that the CGI intends to call *grep* on a text database and that a form provides the regular expression. The naïve approach

```
system("grep $exp database");
```

has a number of problems. Consider *exp* with the value "`root /etc/passwd;rm`". Not only does it read the wrong file, it deletes the real database!

b)

```
#!/usr/local/bin/perl

use LWP::RobotUA;
use HTTP::Request;
use HTTP::Response;

my $ua = new LWP::RobotUA('hcat_RobotUA', 'examples@ora.com');

$ua->proxy('http', 'http://proxy.ora.com:8080/');
$ua->no_proxy('ora.com');

my $request = new HTTP::Request('GET', $ARGV[0]);
my $response = $ua->request($request);
if ($response->is_success) {
     print $response->content;
} else {
```

```
     print $response->error_as_HTML;
}
```

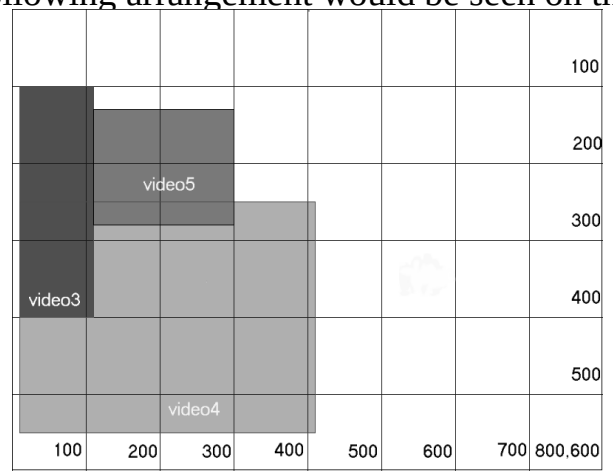
The program code shown above can be easily altered to run faster by doing one or two simple ‘search and replace’ operations in any basic text editor (such as WordPad).

c)

When the following SMIL code is run:

```
<smil>
 <head>
  <layout type="text/smil-basic">
   <channel id="chan1" left="10" top="40" width="100"  height="100" z-index="7" />
   <channel id="chan2" left="10" top="70" width="200"  scale="slice"  z-index="5"
/>
   <channel id="chan3" left="10" top="100" width="100"  height="300"
            scale="fill" z-index="3" />
   <channel id="chan4" left="10" top="250" scale="visible" z-index="1" />
   <channel id="chan5" left="100" top="130" scale="visible" z-index="2" />
 </layout>
 </head>
 <body>
  <seq>
   <par>
    <video id="video1" src="video/m3_208x160.mpg"  channel="chan1" dur="10s" />
    <video id="video2" src="video/m3_416x320.mpg"  channel="chan2" dur="10s" />
   </par>
   <par>
    <video id="video3" src="video/m4_208x160.mpg" channel="chan3" dur="8s" />
    <video id="video4" src="video/m4_416x320.mpg" channel="chan4" begin="2s"
dur="6s" />
    <video id="video5" src="video/m5_416x320.mpg" channel="chan5" begin="5s"
dur="5s" />
   </par>
  </seq>
 </body>
</smil>
```

the following arrangement would be seen on the screen between 15 – 18 sec:



d)

Examples of new standards built on top of XML are SMIL, XHTML, and MathML.

e) None of (a) to (d) above have any problems. They are all correct.

16)
Which one of the following (all correctly working) examples of SSI will give the server the most work? Assume that the server is the version of Apache we have used on the Windows platform, and that it is correctly configured to handle SSI.

a) That's the end of this web page. <!--#include file="footer.html" ->
b) This page has been accessed <!--#exec cgi="/cgi-bin/counter.pl"--> times.
c) Welcome to my web site <!--echo var="REMOTE_USER"-->
d) The next page was last updated <!--#flastmod virtual="page2.html"-->
e) Before you load the next web page I should warn you it has an image on it which is <!--#fsize virtual="/images/all.jpg"--> bytes long.

17)
The following code is loaded by Netscape.

```
<HTML>
<BODY>

<SCRIPT>
input = "1h111e23_1e23";
esc = "abcdefghijklmnopqrstuvwxyz";
splitter = " ";

function init() {
   splitter = "_";
   var esc = "0123456789" + splitter;
}

function process(input) {
      var immed="";
      ret = "";

      for (var i=0; i<input.length; i++) {
            temp = "" + input.substring(i, i+1);
            if (esc.indexOf(temp) == "-1") {
                  immed += temp;
            }
      }

      splitstring = immed.split(splitter);
      for(i = 0; i < splitstring.length; i++)
            ret += splitstring[i];

    return ret;
}
```

```
init();


document.write(eval( 'process' + '(input)' + ';' ));

</SCRIPT>
</BODY>
</HTML>
```

What can you see written in the output document?

a) hee
b) process(input);
c) 111123123
d) 111246
e) andydave

Q18)
The following code is loaded by Netscape.

```
<HTML>
<HEAD>

<SCRIPT>

the_message="";

function wakeMeIn3()
{
    var the_message = "Study";
    setTimeout("alert_and_go(" + the_message + ");", 3000);
    return false;
}


function alert_and_go(mesg)
{
    if (mesg!="") alert(mesg);
    window.open('http://www.ust.hk/','HKUST','location = yes ,
menubar = yes');
}

</SCRIPT>
</HEAD>
<BODY>

<a href="http://www.technologypost.com" onClick="return
wakeMeIn3();">He He</a>

</BODY>
</HTML>
```

After loading a link is shown anchored to the text ‘He He’, as shown below.

After the user clicks on the link, which one of the following is true?

a) The browser goes to the web site *www.technologypost.com*.
b) After 3 seconds, an alert is shown with the message “Study”.
c) Netscape encounters a JavaScript error.
d) The browser goes to the web site *www.ust.hk*.
e) Nothing happens at all.

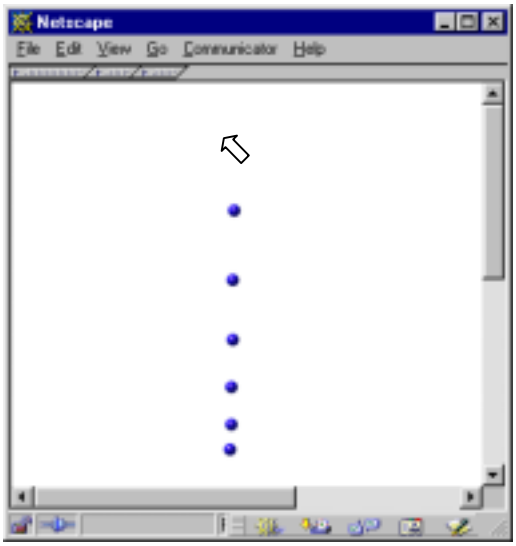
Q19)
The following code is loaded by Netscape.

```
<html>
<body>
<BODY>

<div id="dot0" style="position: absolute; visibility: hidden; height: 11; width:
11;"><img src="bullet.gif" height=11 width=11></div>
<div id="dot1" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot2" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot3" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot4" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot5" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot6" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>

<SCRIPT LANGUAGE="JavaScript">

<!-- Begin
var nDots = 7;
var Xpos = 0;
var Ypos = 0;

var DELTAT = .02;
var SEGLEN = 10;
var SPRINGK = 10;
var MASS = 1.5;
var GRAVITY = 50;
var RESISTANCE = 10;
var STOPVEL = 0.1;
var STOPACC = 0.1;
var DOTSIZE = 11;
var BOUNCE = 0.75;
var isNetscape = navigator.appName=="Netscape";
var followmouse = true;
var dots = new Array();
init();
function init() {
        var i = 0;
```

```
        for (i = 0; i < nDots; i++) {
                dots[i] = new dot(i);
        }
        for (i = 0; i < nDots; i++) {
                dots[i].obj.left = dots[i].X;
                dots[i].obj.top = dots[i].Y;
        }
        if (isNetscape) {
                startanimate();
        } else {
                setTimeout("startanimate()", 3000);
        }
}

function dot(i) {
        this.X = Xpos;
        this.Y = Ypos;
        this.dx = 0;
        this.dy = 0;
        if (isNetscape) {
                this.obj = eval("document.dot" + i);
        } else {
                this.obj = eval("dot" + i + ".style");
        }
}

function startanimate() {
        setInterval("animate()", 20);
}

function setInitPositions(dots) {
        var startloc = document.all.tags("LI");
        var i = 0;
        for (i = 0; i < startloc.length && i < (nDots - 1); i++) {
                dots[i+1].X = startloc[i].offsetLeft
                startloc[i].offsetParent.offsetLeft - DOTSIZE;
                dots[i+1].Y = startloc[i].offsetTop +
                startloc[i].offsetParent.offsetTop + 2*DOTSIZE;
        }
        dots[0].X = dots[1].X;
        dots[0].Y = dots[1].Y - SEGLEN;
}

function MoveHandler(e) {
        Xpos = e.pageX;
        Ypos = e.pageY;
        return true;
}

function MoveHandlerIE() {
        Xpos = window.event.x + document.body.scrollLeft;
        Ypos = window.event.y + document.body.scrollTop;
}

if (isNetscape) {
        document.captureEvents(Event.MOUSEMOVE);
        document.onMouseMove = MoveHandler;
} else {
        document.onmousemove = MoveHandlerIE;
}

function vec(X, Y)
{
        this.X = X;
        this.Y = Y;
}

// adds force in X and Y to spring for dot[i] on dot[j]
function springForce(i, j, spring)
{
        var dx = (dots[i].X - dots[j].X);
        var dy = (dots[i].Y - dots[j].Y);
        var len = Math.sqrt(dx*dx + dy*dy);
        if (len > SEGLEN) {
                var springF = SPRINGK * (len - SEGLEN);
                spring.X += (dx / len) * springF;
```

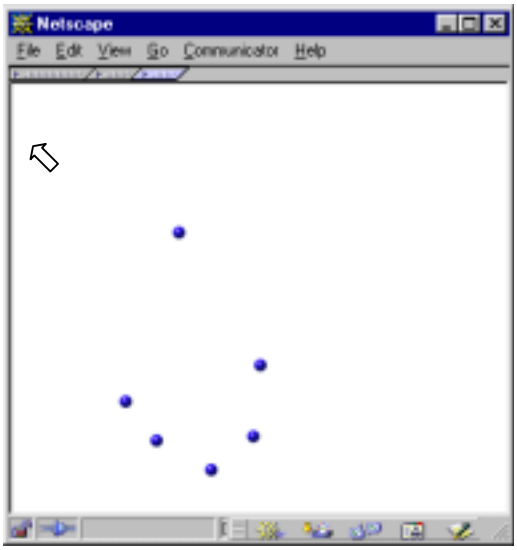
```
                spring.Y += (dy / len) * springF;
        }
}

function animate() {
        var start = 0;
        if (followmouse) {
                dots[0].X = Xpos;
                dots[0].Y = Ypos;
                start = 1;
        }

        for (i = start ; i < nDots; i++ ) {
                var spring = new vec(0, 0);
                if (i > 0) {
                        springForce(i-1, i, spring);
                }
                if (i < (nDots - 1)) {
                        springForce(i+1, i, spring);
                }
                var resist = new vec(-dots[i].dx * RESISTANCE, -dots[i].dy *
RESISTANCE);
                var accel = new vec((spring.X + resist.X)/ MASS, (spring.Y +
resist.Y)/ MASS + GRAVITY);
                dots[i].dx += (DELTAT * accel.X);
                dots[i].dy += (DELTAT * accel.Y);
                if (Math.abs(dots[i].dx) < STOPVEL &&
                        Math.abs(dots[i].dy) < STOPVEL &&
                        Math.abs(accel.X) < STOPACC &&
                        Math.abs(accel.Y) < STOPACC) {
                        dots[i].dx = 0;
                        dots[i].dy = 0;
                }
                dots[i].X += dots[i].dx;
                dots[i].Y += dots[i].dy;
                var height, width;
                if (isNetscape) {
                        height = window.innerHeight + document.scrollTop;
                        width = window.innerWidth + document.scrollLeft;
                } else {
                        height = document.body.clientHeight +
document.body.scrollTop;
                        width = document.body.clientWidth + document.body.scrollLeft;
                }
                if (dots[i].Y >=  height - DOTSIZE - 1) {
                        if (dots[i].dy > 0) {
                                dots[i].dy = BOUNCE * -dots[i].dy;
                        }
                        dots[i].Y = height - DOTSIZE - 1;
                }
                if (dots[i].X >= width - DOTSIZE) {
                        if (dots[i].dx > 0) {
                                dots[i].dx = BOUNCE * -dots[i].dx;
                        }
                        dots[i].X = width - DOTSIZE - 1;
                }
                if (dots[i].X < 0) {
                        if (dots[i].dx < 0) {
                                dots[i].dx = BOUNCE * -dots[i].dx;
                        }
                        dots[i].X = 0;
                }
                dots[i].obj.left = dots[i].X;
                dots[i].obj.top =  dots[i].Y;
        }
}
// End -->

</script>
</body>
</html>
```

An example screen display from one point in time is shown here. *The arrow shown represents the mouse position at that point in time.*



Which one of the following is *false*?

a) Six balls are usually visible in the window
b) If the mouse moves inside the window, all the balls move
c) The balls can bounce off the top of the window
d) The balls can bounce off the left side of the window
e) The following display/ status is possible. *The arrow shown represents the mouse position at that point in time. Ignore any presence/ absence of sliders on the Netscape window, these are irrelevant to the question.*



Q20)
A table is created using the following SQL statement:

create table HTML (id int primary key, URL varchar(50), Last_modified char(50), Size int)

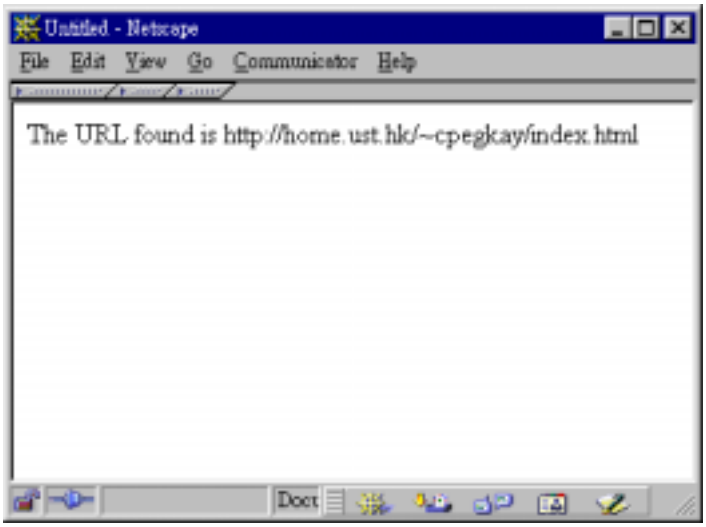
The table is then filled with data as follows:

Id	URL	Last_modified	Size
1	http://home.ust.hk/~cpegkay/index.html	10 March 2000 15:08 HKT	5
2	http://www.cs.ust.hk/index.html	12 April 2000 10:16 HKT	4
3	http://www.tom.com/index.html	25 May 2000 16:58 HKT	20

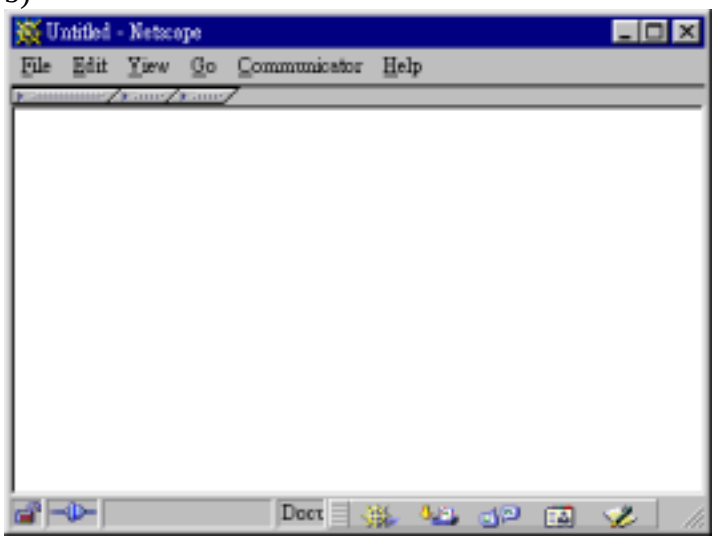
What will be the output of the following program when the program is called using CGI (assume the user has typed the correct address, etc)?

```
#!/usr/local/bin/perl

# The 'use', 'require', etc, statements are the same as used for the
# course work and have not been repeated here

$d = &dblogin();  # login to the database and return an database
                  # object to access the database

print "Content-type: text/html\n\n";
print "<html>\n";
print "<title>Untitled</title>\n";
print "<body>\n";
@result = &sql($d, "select URL from HTML where Size < 10");
foreach $a(@result) {
      if ($a =~ m/cs\.ust\.hk/i) {
            print "The URL found is $a<br>\n";
      }
}
print "</body>\n"
print "</html>\n";

      # The code for dblogin() is the same as used
      # for the course work and has not been repeated here
```

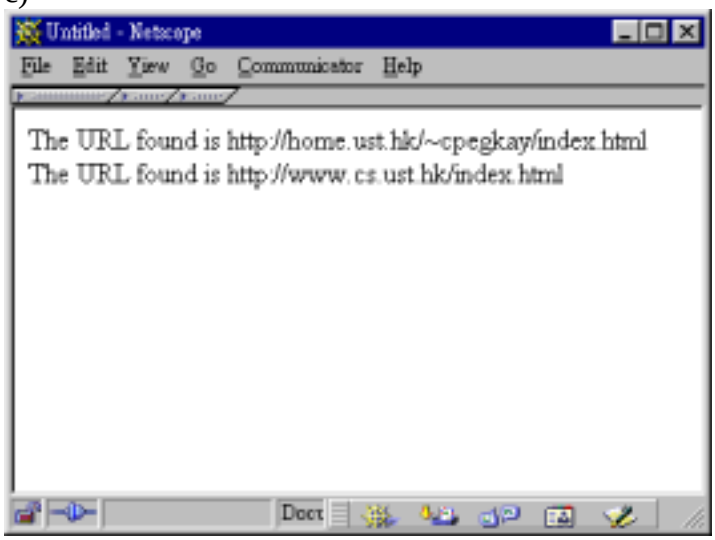
a)



b)


Untitled - Netscape
File Edit View Go Communicator Help
Document Done

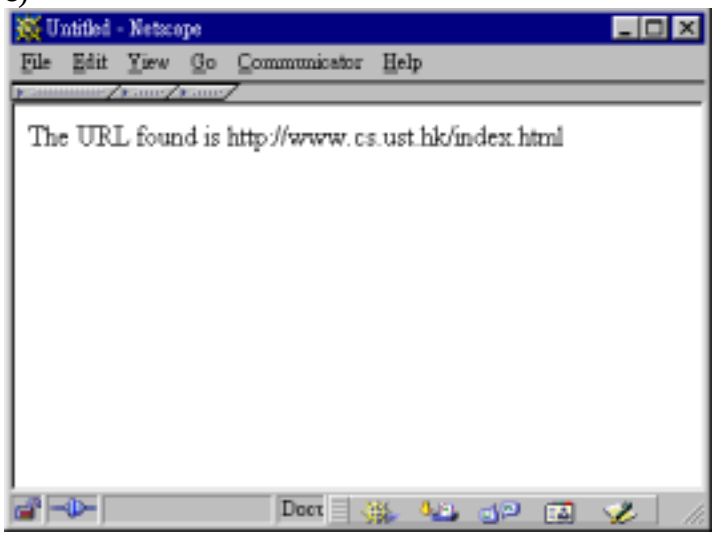
c)


Untitled - Netscape
File Edit View Go Communicator Help
The URL found is http://home.ust.hk/~cpegkay/index.html
The URL found is http://www.cs.ust.hk/index.html
Document Done

d)


Netscape
File Edit View Go Communicator Help
CGI Programming Error
The server encountered an error while processing your CGI program.
Please check your program for any bugs or syntax error.
For reference on CGI programming, please refer to
http://hoohoo.ncsa.uiuc.edu/cgi
Hints: In most common cases, the web server detected "an
Document Done

e)



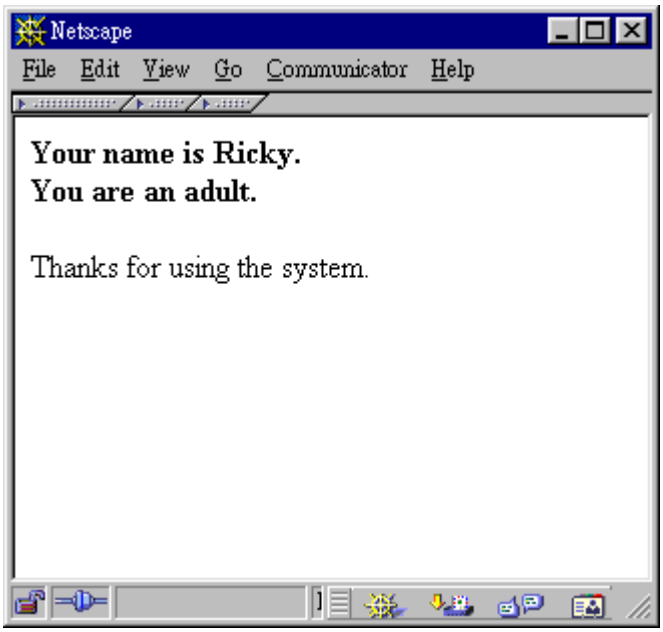
Q21)

We have written the following form in HTML as shown below:

The actual HTML code is given below:

```
<html>
<form action="URL of the CGI program" method="GET">
Please type in your name and age below <br>
Name : <input type="text" name="Name" size="20"><br>
Age : <br>
<select name="Age" size="3">
<option value="0">0-10
<option value="11">11-20
<option value="21">21 or above
</select><br>
<input type="submit">
</html>
```

Using the form, a user types in the name 'Ricky', and chooses '21 or above' (as shown in the previous screen dump) and then submits the data. The CGI program then outputs the following.



Which one of the following is the possible code for the CGI program?

a)

```
#!/usr/local/bin/perl

read(STDIN,$cgivar,$ENV{'CONTENT_LENGTH'});
@cgipair=split(/&/,$cgivar);
foreach $pair (@cgipair){
# Get the name and the value for a name and value pair
       ($name,$val)=split (/=/,$pair,2);
       # Restore the special characters
       $val=~tr/+/ /;
       $val=~ s/%(..)/pack("c",hex($1))/ge;

       # Store the result with a hash (associative array)
       if ($name eq "Name") {
              $username = $val;
       }
       elsif ($name eq "Age") {
              $userage = $val;
       }
}

print "Content-type: text/html\n\n";
print "<html>\n";
print "<b>Your name is $username.<br>\n";
if ($userage == 21) {
       print "You are an adult.<br></b>\n";
}
elsif ($userage == 11) {
       print "You are a teenager.<br></b>/n";
}
else {
       print "You are a child.<br></b>/n";
}
print "<br>Thanks for using the system.<br>\n";
print "</html>\n";
```

b)

```
#!/usr/local/bin/perl

read(STDIN,$cgivar,$ENV{'CONTENT_LENGTH'});
@cgipair=split(/&/,$cgivar);
foreach $pair (@cgipair){
# Get the name and the value for a name and value pair
       ($name,$val)=split (/=/,$pair,2);
       # Restore the special characters
       $val=~tr/+/ /;
       $val=~ s/%(..)/pack("c",hex($1))/ge;
}

print "Content-type: text/html\n\n";
print "<html>\n";
print "<b>Your name is $name.<br>\n";
if ($val == 21) {
       print "You are an adult.<br></b>\n";
}
elsif ($val == 11) {
       print "You are a teenager.<br></b>/n";
}
else {
       print "You are a child.<br></b>/n";
}
print "<br>Thanks for using the system.<br>/n";
print "</html>\n";
```

c)

```
#!/usr/local/bin/perl

$cgivar = $ENV{'QUERY_STRING'};

@cgipair=split(/&/,$cgivar);
foreach $pair (@cgipair){
       # Get the name and the value for a name and value pair
       ($name,$val)=split (/=/,$pair,2);
       # Restore the special characters
       $val=~tr/+/ /;
       $val=~ s/%(..)/pack("c",hex($1))/ge;

       # Store the result
       if ($name eq "Name") {
              $username = $val;
       }
       elsif ($name eq "Age") {
              $userage = $val;
       }
}

print "Content-type: text/plain\n\n";
print "<html>\n";
print "<b>Your name is $username.<br>\n";
if ($userage == 21) {
       print "You are an adult.<br></b>\n";
}
elsif ($userage == 11) {
       print "You are a teenager.<br></b>/n";
}
else {
       print "You are a child.<br></b>/n";
}
print "<br>Thanks for using the system.<br>/n";
print "</html>\n";
```

d)

```
#!/usr/local/bin/perl

$cgivar = $ENV{'QUERY_STRING'};

$cgivar =~ /Name=(.*)\&Age=(.*)/;
$username = $1;
$userage = $2;

# Restore the special characters
$username=~tr/+/ /;
$username=~ s/%(..)/pack("c",hex($1))/ge;
$userage=~tr/+/ /;
$userage=~ s/%(..)/pack("c",hex($1))/ge;

print "Content-type: text/html\n\n";
print "<html>\n";
print "<b>Your name is $username.<br>\n";
if ($userage == 21) {
       print "You are an adult.<br></b>\n";
}
elsif ($userage == 11) {
       print "You are a teenager.<br></b>/n";
}
else {
       print "You are a child.<br></b>/n";
}
print "<br>Thanks for using the system.<br>\n";
print "</html>\n";
```

e)

```
#!/usr/local/bin/perl

read(STDIN,$cgivar,$ENV{'CONTENT_LENGTH'});
@cgipair=split(/&/,$cgivar);
foreach $pair (@cgipair){
# Get the name and the value for a name and value pair
       ($name,$val)=split (/=/,$pair,2);
       # Restore the special characters
       $val=~tr/+/ /;
       $val=~ s/%(..)/pack("c",hex($1))/ge;

       # Store the result
       if ($name == "Name") {
              $username = $val;
       }
       elsif ($name == "Age") {
              $userage = $val;
       }
}

print "Your name is $username.\n";
if ($userage == 21) {
       print "You are an adult.\n";
}
elsif ($userage == 11) {
       print "You are a teenager./n";
}
else {
       print "You are a child./n";
}
print "/nThanks for using the system./n";
```

Q22)
Someone is writing a Visual Basic 6.0 program that uses a Web browser, which is implemented using the Internet Control. The web browser is being used to view some information on a particular web page several times. However, the person finds that the latest version of the web page is sometimes not shown. It is found that a cache problem occurs after the page has been visited once. How many of the following are true?

1) To solve the cache problem, a cgi program can be used to display the information, with time/date information attached to the end of the URL, even though this information is not actually used by the CGI program.
2) To solve the cache problem, a cgi program can be used to display the information, with an integer counter attached to the end of the URL. The counter is initialised to zero when the VB program starts. Once a cgi program is visited, the counter will be incremented.
3) Assuming that the name of browser in the VB program is "WB", to solve the cache problem, we can execute the command ***"WB.Refresh"*** at the end of the ***"WB_DownloadComplete"*** event code.
4) To avoid the cache problem, use a Common Dialog Control

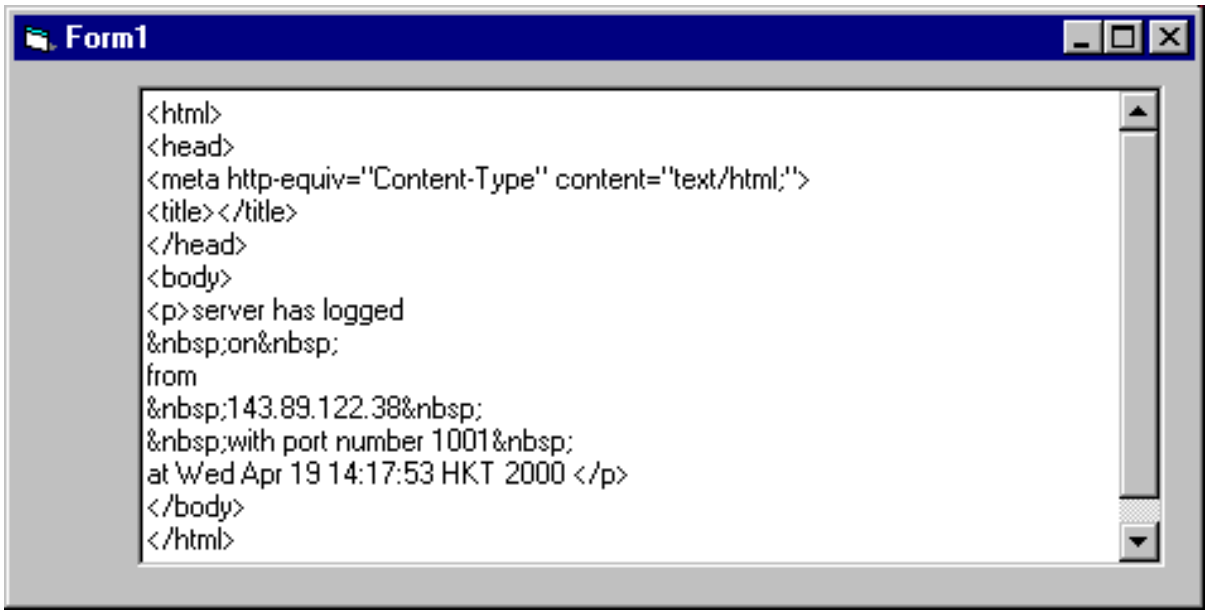
a) None of them
b) One of them
c) Two of them
d) Three of them
e) Four of them

Q23)
In Microsoft Visual Basic 6.0, there is a component called **Microsoft Internet Transfer Control 6.0**, which is able to extract the content of an HTML page. The usage of this control is described below.

Assume that two elements are added to a form of a VB project:

- A RichTextBox, named RTB1
- An Internet Transfer Control, named Inet1

If the following command is executed in VB 6.0:
RTB1.text = Inet1.OpenURL("www.cs.ust.hk/~cslcw/cgi-bin/mystatus.html")
the result will be like this:



Assume that we implement a client/ server application in VB 6.0 using the Winsock Control. Obviously, if the client wants to connect to the server, the client needs to know the IP address (or hostname) and port number of the server.

Now, if the server generates the web page as shown above when it starts and the client obtains the web page, then conceptually it works like a server sending an encoded message to the client, if the client obtains the content of this HTML Page using Internet Transfer Control. Thus, the client can connect to server without manual input.

Your task is to complete the server side code and client side code so that the client can connect to the server successfully using the mechanism described above. The server code is shown below. The client code is shown following. Possible answers are then listed.

```
‘ Server's program
Dim intMax as Integer ‘ The default value is 0

Private Sub Form_Load()
‘ getPortNum returns the port number and assign it to WsServer(0)
        WsServer(0).LocalPort = getPortNum()
        ‘ Fill in the code using Either A1 or B1

        Call updatemystatus ‘ To generate a page specifying the server's information
End sub

Private Sub WsServer_ConnectionRequest(index as Integer, ByVal requestID as _ Long)
        If index = 0 then
                ‘ Fill in the code using Either A2 or B2

        End if
End Sub
```

```
' Client program
Private Sub Form_Load()
        RTB1.Text = Inet1.OpenURL("www.cs.ust.hk/~cslcw/cgi-bin/mystatus.html")
    ‘ Fill in the code using Either A3 or B3


    ‘ Fill in the code using Either A4 or B4


End sub
```

Part 1

A1	WsServer(0).Listen
B1	WsServer(0).Accept intMax

Part 2

A2	intMax = intMax + 1 Load WsServer(intMax) WsServer(intMax).LocalPort = 0 WsServer(intMax).Accept requestID
B2	WsServer(0).Accept requestID

Part 3

A3	Result = split(RTB1.Text,”;”,-1) IP = split(Result(3),”&”,-1) WsClient1.RemoteHostIP = IP(0)
B3	Result = split(RTB1.Text,” ”,-1) WsClient1.RemoteHostIP = Result(3)

Part 4

A4	WsClient1.RemotePort = CInt(Val(Right(Result(5),4))) WsClient.Connect
B4	Port = split(RTB1.Text,”number “,-1) WsClient.RemotePort = CInt(Val(Left(Port(1),4))) WsClient.Connect

For each part, you have to choose one of the options provided. Which of the combinations below will make the two programs work correctly?

a) A1, A2, A3, A4
b) A1, B2, A3, B4
c) A1, B1, A3, A4
d) A1, A2, B3, A4
e) None of the above

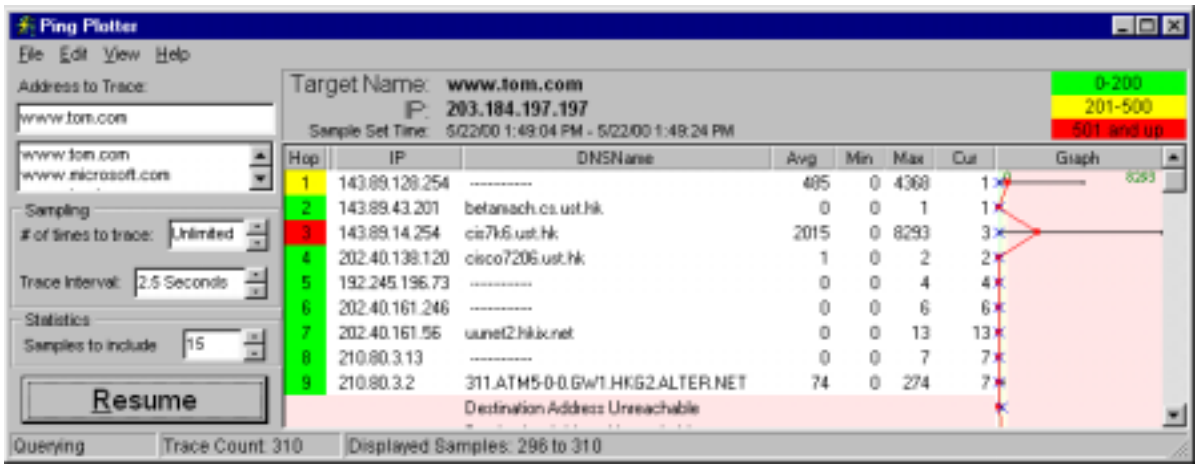
Q24)

How many of the following types of internet files/ data can be created using a simple text editor such as Microsoft Notepad?

- XML
- HTML
- Flash
- SMIL
- Apache httpd.conf configuration
- Cascading Style Sheets
- JavaScript code for animating some images
- Perl code to use the GD library (PNG version)
- A web page containing a Cold Fusion instruction to list the contents of a database table

Select your answer from the following
a) 1 or 2 of the above
b) 3 or 4 of the above
c) 5 or 6 of the above
d) 7 or 8 of the above
e) 9 of the above

Q25)
Consider the following screen dump.



Assume that the information shown in the screen dump shows the situation right now. Which one of the following is false?

a) If you type ‘ping 203.184.197.197’ (in a Unix machine) it would have the same result as typing ‘ping www.tom.com’, assuming that there is no problem with the DNS system
b) If you type ‘ping 210.80.3.2’ (in a Unix machine) the response would indicate that that web site was alive (meaning that it was accessible through the internet)

c) If a bird was flying directly from machine 143.89.128.254 to 192.245.196.73, it would certainly have to fly further than if it flew directly from 143.89.128.254 to 202.40.138.120.
d) Some or all of the functions provided by Ping Plotter are also available by using the Neotrace program
e) The program Ping Plotter tries to send out packets to one or more computers on the internet

Q26)
A friend has borrowed a computer for a few days and wants to try out as many browsers as he can during that time, without paying any money. How many of the following browsers can the friend try out?

- Amaya
- Opera
- Internet Explorer
- Lynx
- Neoplanet
- Netscape Navigator

a) Two of the above
b) Three of the above
c) Four of the above
d) Five of the above
e) Six of the above

Q27)
If each datagram hold 500 bytes, how many datagrams will be required to send an image that has the following details?

Local cache file: M02KG1AK.PNG
Last Modified: Friday, April 14, 2000 11:49:57 PM Local time
Last Modified: Friday, April 14, 2000 3:49:57 PM GMT
Content Length: 250
Expires: No date given
Charset: Unknown

a) 1
b) 2
c) 3
d) 4
e) 5

Q28) This is a log file from a web server.

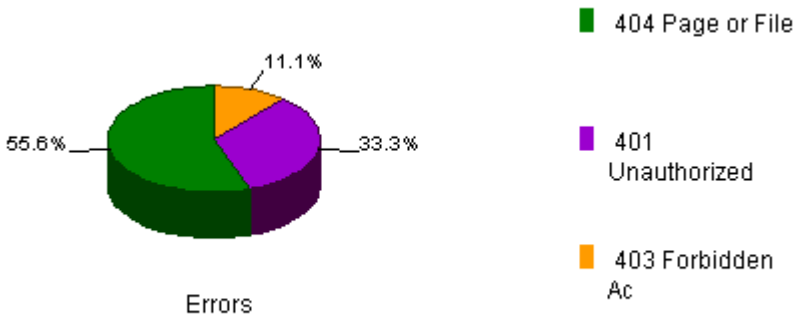
```
hkusud.hku.hk - - [22/May/2000:14:02:59 +0800] "GET /cgi-bin/cgiwrap/ismt101/genform.pl?lab=01 HTTP/1.0" 302 316
hkusud.hku.hk - - [22/May/2000:14:03:00 +0800] "GET /cgi-bin/cgiwrap/ismt101/votecount-post.cgi HTTP/1.0" 200 843
hkusud.hku.hk - - [22/May/2000:14:03:00 +0800] "GET /cgi-bin/cgiwrap/style.css HTTP/1.0" 200 877
hkusud.hku.hk - - [22/May/2000:14:03:00 +0800] "GET /~ismt101/vote.gif HTTP/1.0" 200 12283
adsl-63-202-206-2.dsl.snfc21.pacbell.net - - [22/May/2000:14:03:03 +0800] "GET /cgi-
bin/cgiwrap/~acct311/course_info.cgi HTTP/1.0" 200 2264
ccd122.ust.hk - - [22/May/2000:14:03:14 +0800] "GET /~mgto261/index.html HTTP/1.0" 200 3237
ccd122.ust.hk - - [22/May/2000:14:03:14 +0800] "GET /~mgto261/Up_line.gif HTTP/1.0" 404 285
ccd122.ust.hk - - [22/May/2000:14:03:18 +0800] "GET /~mgto261/cases.html HTTP/1.0" 200 8009
ccd122.ust.hk - - [22/May/2000:14:03:18 +0800] "GET /~mgto261/bluenew.gif HTTP/1.0" 200 9988
dmb453.resnet.ust.hk - - [22/May/2000:14:03:19 +0800] "GET /~ismt101/style.css HTTP/1.1" 304 -
dmb453.resnet.ust.hk - - [22/May/2000:14:03:19 +0800] "GET /~ismt101/bg.gif HTTP/1.1" 304 -
ctz004.ust.hk - eteval [22/May/2000:14:04:06 +0800] "GET /cosset/images/question1.gif HTTP/1.0" 200 995
ctz004.ust.hk - eteval [22/May/2000:14:04:06 +0800] "GET /cosset/images/report1.gif HTTP/1.0" 200 979
ctz004.ust.hk - eteval [22/May/2000:14:04:06 +0800] "GET /cosset/images/course1.gif HTTP/1.0" 200 969
210.176.188.125 - - [22/May/2000:14:04:32 +0800] "GET /~mark321/icon6.gif HTTP/1.0" 404 284
203.36.10.218 - - [22/May/2000:14:08:11 +0800] "GET /~mgto650a/ HTTP/1.0" 200 13125
sy5kts3-p19.ust.hk - - [22/May/2000:14:08:29 +0800] "GET /cgi-bin/auth/cgiwrap/~s_start/wisdom/getuser.cgi HTTP/1.0"
401 468
203.36.10.218 - - [22/May/2000:14:09:47 +0800] "GET /~mgto222 HTTP/1.0" 301 307
203.36.10.218 - - [22/May/2000:14:10:15 +0800] "GET /etc/cosset/ HTTP/1.0" 401 468
sy5kts4-p45.ust.hk - - [22/May/2000:14:11:36 +0800] "GET /~mgto332 HTTP/1.0" 301 307
sy5kts4-p45.ust.hk - - [22/May/2000:14:11:36 +0800] "GET /~mgto332/ HTTP/1.0" 200 502
imz085.ust.hk - - [22/May/2000:14:15:02 +0800] "GET /~ismt501/grlist.html HTTP/1.0" 200 4947
imz085.ust.hk - - [22/May/2000:14:15:04 +0800] "GET /~ismt501/FIN.doc HTTP/1.0" 200 43520
202.163.6.249 - - [22/May/2000:14:15:16 +0800] "GET /~mgto332/Exam/SecondExamAnswers.doc HTTP/1.0" 200 36864
imz034.ust.hk - - [22/May/2000:14:15:26 +0800] "GET /~ismt300j/ HTTP/1.0" 304 -
imz034.ust.hk - - [22/May/2000:14:15:26 +0800] "GET /~ismt300j/title/title.html HTTP/1.0" 304 -
imz034.ust.hk - - [22/May/2000:14:15:26 +0800] "GET /~ismt300j/toc/toc.html HTTP/1.0" 304 -
imz034.ust.hk - - [22/May/2000:14:15:26 +0800] "GET /~ismt300j/info/info.html HTTP/1.0" 304 -
imz034.ust.hk - - [22/May/2000:14:15:26 +0800] "GET /~ismt300j/css/contentFrame.css HTTP/1.0" 304 -
imz034.ust.hk - - [22/May/2000:14:15:27 +0800] "GET /~ismt300j/images/second-edition.gif HTTP/1.0" 304 -
ip-249-133-66-202.diyixian.com - - [22/May/2000:14:19:55 +0800] "GET /~civl261/ HTTP/1.1" 403 290
dma013.resnet.ust.hk - - [22/May/2000:14:20:03 +0800] "GET /~mark321/cases.ppt HTTP/1.1" 200 59392
ccb046.ust.hk - - [22/May/2000:14:20:06 +0800] "GET /~mgto332 HTTP/1.0" 301 307
ccb046.ust.hk - - [22/May/2000:14:20:06 +0800] "GET /~mgto332/ HTTP/1.0" 200 502
```

```
ccb046.ust.hk - - [22/May/2000:14:20:06 +0800] "GET /~mgto332/Index/contents.html HTTP/1.0" 200 2474
ccb046.ust.hk - - [22/May/2000:14:20:22 +0800] "GET /~mgto332/Gif/colorline.gif HTTP/1.0" 404 291
dma013.resnet.ust.hk - - [22/May/2000:14:20:35 +0800] "GET /~mark321/gillette2.doc HTTP/1.1" 200 98816
dma013.resnet.ust.hk - - [22/May/2000:14:20:38 +0800] "OPTIONS /~mark321 HTTP/1.1" 301 319
dma013.resnet.ust.hk - - [22/May/2000:14:20:41 +0800] "GET /_vti_inf.html HTTP/1.1" 404 290
dma013.resnet.ust.hk - - [22/May/2000:14:20:41 +0800] "POST /_vti_bin/shtml.exe/_vti_rpc HTTP/1.1" 404 304
dma013.resnet.ust.hk - - [22/May/2000:14:20:41 +0800] "OPTIONS /~mark321/gillette2.doc HTTP/1.1" 200 -
203.197.63.130 - - [22/May/2000:14:23:09 +0800] "GET /~mgto321/StarTV.html HTTP/1.1" 200 4508
203.197.63.130 - - [22/May/2000:14:23:14 +0800] "GET /~mgto321/clickhere.gif HTTP/1.1" 200 4376
203.197.63.130 - - [22/May/2000:14:23:19 +0800] "GET /~mgto321/ustpic3.jpg HTTP/1.1" 200 40035
203.197.63.130 - - [22/May/2000:14:23:25 +0800] "GET /~mgto321/colorhome.gif HTTP/1.1" 200 1456
sy5kts4-p02.ust.hk - - [22/May/2000:14:25:38 +0800] "GET /~mgto332/Gif/updated.gif HTTP/1.0" 200 4083
sy5kts4-p02.ust.hk - - [22/May/2000:14:25:39 +0800] "GET /~mgto332/Gif/bluenote.gif HTTP/1.0" 200 913
mez165.ust.hk - - [22/May/2000:14:34:06 +0800] "GET /cgi-
bin/cgiwrap/~s_start/course/viewcourseurl.cgi?course=mgto261 HTTP/1.1" 302 294
mez165.ust.hk - - [22/May/2000:14:34:06 +0800] "GET /~mgto261/Up_line.gif HTTP/1.1" 404 297
mez165.ust.hk - - [22/May/2000:14:34:07 +0800] "GET /~mgto261/Arrowrig.gif HTTP/1.1" 200 206
mez165.ust.hk - - [22/May/2000:14:34:10 +0800] "GET /~mgto261/robot1.JPG HTTP/1.1" 200 376053
ustsu68.ust.hk - - [22/May/2000:14:34:12 +0800] "GET /cgi-bin/auth/cgiwrap/~s_start/wisdom/getuser.cgi HTTP/1.0" 401
468
mez165.ust.hk - - [22/May/2000:14:34:15 +0800] "GET /~mgto261/syllabus.html HTTP/1.1" 200 24717
crawl1.googlebot.com - - [22/May/2000:14:35:49 +0800] "GET /~mgto121a/PhotoGroupL8a7.html HTTP/1.0" 200 3276
dmd405.resnet.ust.hk - - [22/May/2000:14:36:31 +0800] "GET /~ismt101/ HTTP/1.1" 304 -
```

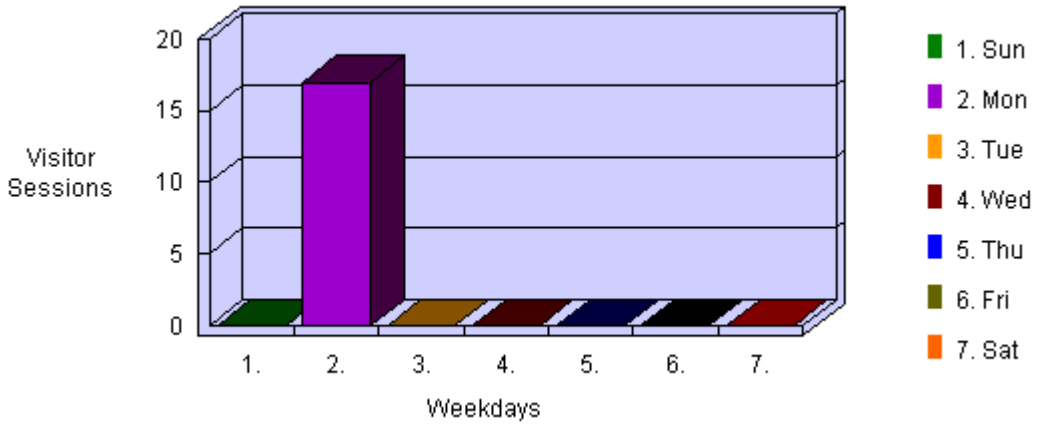
The log file shown above is analysed using the *WebTrends Log Analyzer* software, from *http://www.webtrends.com/products/log/default.htm*. The program produces a report which includes a number of graphs. Four of the following graphs have been created by the software after it analyzed this log file. One of the graphs was created from the analysis of a different log file

Which one of the following graphs has *not* been created from the analysis of the log file listed previously?

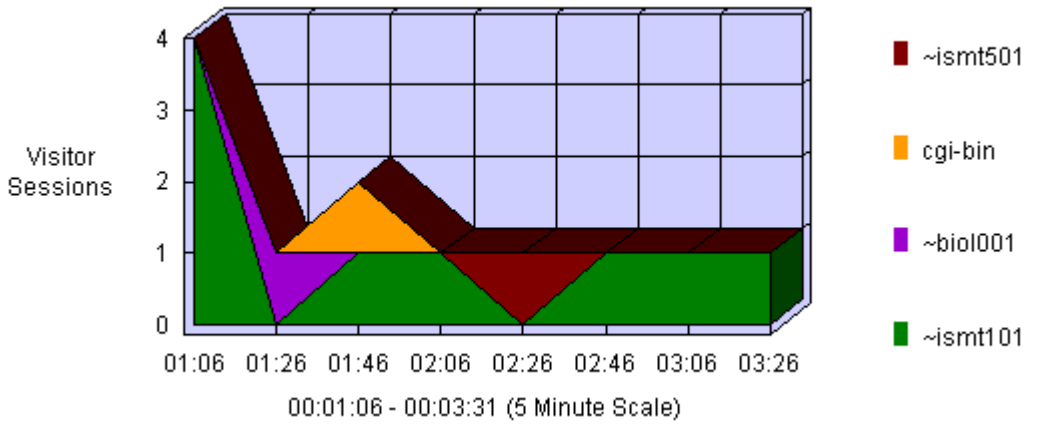
a)



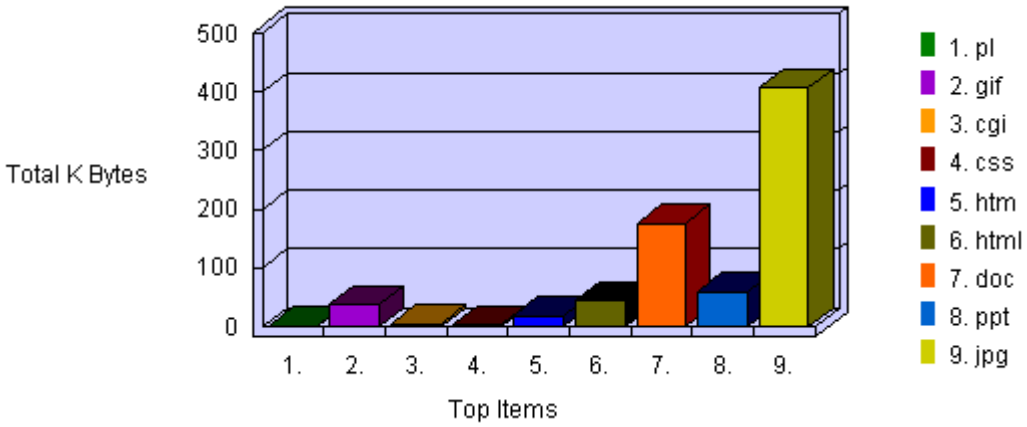
b)



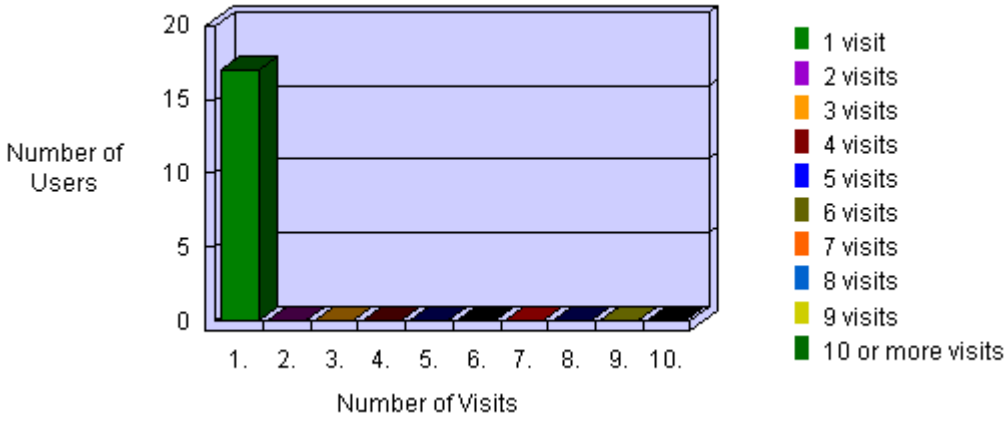
c)



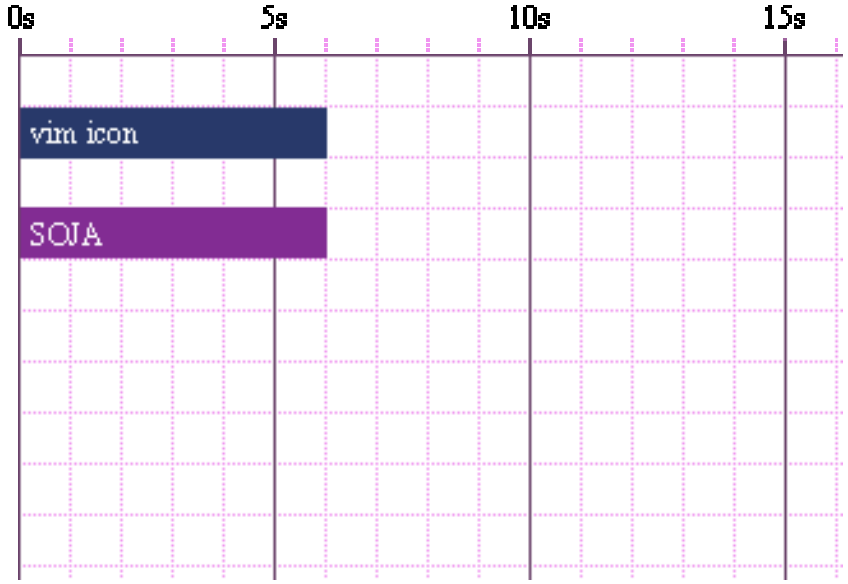
d)



e)



Q29)
Someone wishes to use SMIL to show two images on the screen for a certain period of time, as shown below.



Which one of the following would achieve this effect?

a)

```
<smil>
<head>
<layout>
<root-layout width="300" height="200" background-color="white" />
<region id="vim_icon" left="75" top="50" width="32" height="32" />
<region id="soja_icon" left="150" top="50" width="100" height="30" />
</layout>
</head>
<body>
<par>
<img src="vim32x32.gif" alt="The vim icon" region="vim_icon" dur="6s" />
<img src="madewithsoja.gif" alt="Made with SOJA" region="soja_icon"
dur="6s" />
</par>
</body>
</smil>
```

b)

```
<smil>
<head>
<layout>
<root-layout width="300" height="200" background-color="white" />
<region id="vim_icon" left="75" top="50" width="32" height="32" />
<region id="soja_icon" left="150" top="50" width="100" height="30" />
</layout>
</head>
<body>
<par>
<img src="vim32x32.gif" region="vim_icon" begin="4s" />
<img src="madewithsoja.gif" region="soja_icon" begin="6s" />
</par>
</body>
</smil>
```

c)

```
<smil>
<head>
<layout>
<root-layout width="300" height="200" background-color="white" />
<region id="vim_icon" left="75" top="50" width="32" height="32" />
<region id="soja_icon" left="150" top="50" width="100" height="30" />
</layout>
</head>
<body>
<par>
<img src="vim32x32.gif" region="vim_icon"  id="vim" begin="4s" />
<img src="madewithsoja.gif" region="soja_icon" begin="id(vim)(2s)" />
</par>
</body>
</smil>
```

d)

```
<body>
<switch>
<!-- English only -->
<par system-language="en">
<text src="you_are_english.txt" region="main_message" />
<!-- testing the screen size -->
<switch>
<text src="800_600.en.txt" region="size" system-screen-size="800X600" />
<text src="1024_768.en.txt" region="size" system-screen-size="1024X768" />
<text src="another_size.en.txt" region="size" />
</switch>
<!-- testing the bitrate -->
<switch>
<text src="14000bps.en.txt" region="bitrate" system-bitrate="14000" />
<text src="28000bps.en.txt" region="bitrate" system-bitrate="28000" />
<text src="another_bitrate.en.txt" region="bitrate" />
</switch>
</par>
<!-- French only -->
<par system-language="fr">
<text src="vous_etes_francais.txt" region="main_message" />
<!-- testing the screen size -->
<switch>
<text src="800_600.fr.txt" region="size" system-screen-size="800X600" />
<text system-screen-size="1024X768" />
<text src="another_size.fr.txt" region="size" />
</switch>
<!-- testing the bitrate -->
<switch>
<text src="14000bps.fr.txt" region="bitrate" system-bitrate="14000" />
<text src="28000bps.fr.txt" region="bitrate" system-bitrate="28000" />
<text src="another_bitrate.fr.txt" region="bitrate" />
</switch>
</par>

<!-- The following text is displayed for other languages -->
<text src="unknown_language.txt" region="main_message" />

</switch>
</body>
```

e)

```
<smil>
<head>
<layout>
<root-layout width="300" height="200" background-color="white" />
<region id="vim_icon" left="75" top="50" width="32" height="32" />
</layout>
</head>
<body>
<img src="vim32x32.gif" alt="The vim icon" region="vim_icon" dur="6s"
begin="2s" />
</body>
</smil>
```

Q30)
The people at *www.tommy.com.au* have asked for your advice. They have a single computer in Australia which hosts all the files for *www.tommy.com.au*. There is a separate machine in Hong Kong which hosts all the files for *www.tommy.com*. Previously, *www.tommy.com.au* and *www.tommy.com* had a different set of web pages. But now, to save the cost of maintaining a different set of pages, the company wants all people who type in *www.tommy.com.au* in their browsers to automatically see *www.tommy.com* instead. Which one of the following configurations (on the Australian machine) is most appropriate for this?

a)
```
Alias / “http://www.tommy.com”
```

b)
```
order deny,allow
Deny from all
Allow from tommy.com
```

c)
```
<Location /index.html>
     AuthName ByPassword
     AuthType Basic
     Require valid-user
</Location>
```

d)
```
Redirect / http://www.tommy.com/
```

e)
Include this in the appropriate configuration file:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

Then edit *index.html* so it has the following contents:

```
<META HTTP-EQUIV=REFRESH CONTENT="1; URL=index.shtml">
```

Then edit *index.shtml* so it has the following contents:

```
<!--#exec cmd="http://www.tommy.com"-->
```

Q31)
What version of the exam are you using? (This is written on the front of this exam paper).
a) Version 1
b) Version 2

This page is just a copy of the information shown at the start of question 19

This page is included only to help you answer Q19.

Because the exam pages are double sided, you may find it useful to detach this sheet from the back of the exam. Then you can put it next to the other pages to see all Q19 code at the same time.

Q19)
The following code is loaded by Netscape.

```
<html>
<body>
<BODY>

<div id="dot0" style="position: absolute; visibility: hidden; height: 11; width:
11;"><img src="bullet.gif" height=11 width=11></div>
<div id="dot1" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot2" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot3" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot4" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot5" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>
<div id="dot6" style="position: absolute; height: 11; width: 11;"><img
src="bullet.gif" height=11 width=11></div>

<SCRIPT LANGUAGE="JavaScript">

<!-- Begin
var nDots = 7;
var Xpos = 0;
var Ypos = 0;

var DELTAT = .02;
var SEGLEN = 10;
var SPRINGK = 10;
var MASS = 1.5;
var GRAVITY = 50;
var RESISTANCE = 10;
var STOPVEL = 0.1;
var STOPACC = 0.1;
var DOTSIZE = 11;
var BOUNCE = 0.75;
var isNetscape = navigator.appName=="Netscape";
var followmouse = true;
var dots = new Array();
init();
function init() {
        var i = 0;
```