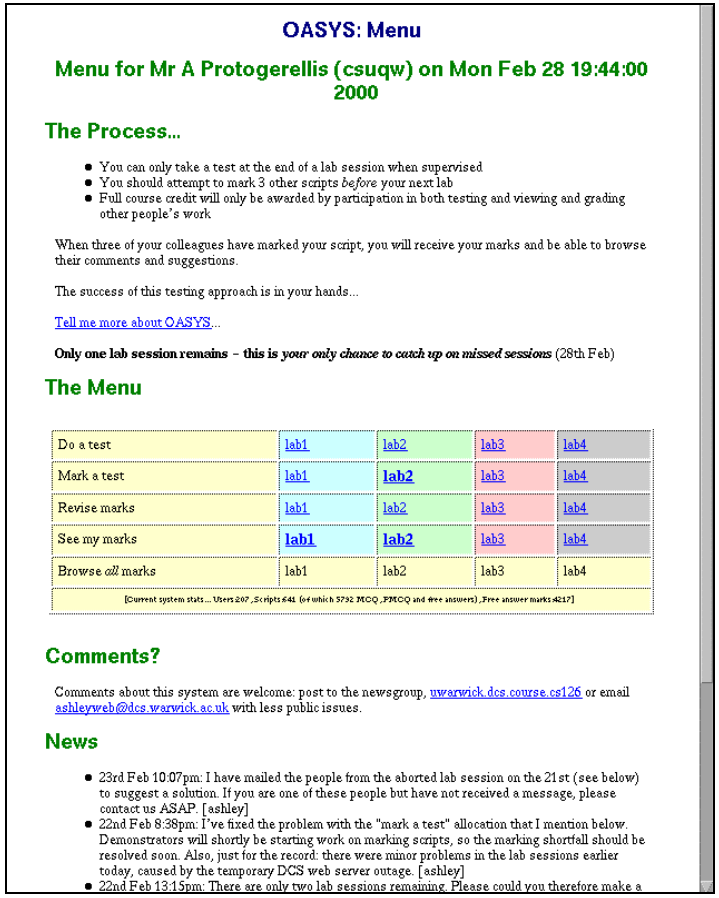


OASYS: Menu

Menu for Mr A Protogerellis (csuqw) on Mon Feb 28 19:44:00 2000

The Process...

- You can only take a test at the end of a lab session when supervised
- You should attempt to mark 3 other scripts *before* your next lab
- Full course credit will only be awarded by participation in both testing and viewing and grading other people's work

When three of your colleagues have marked your script, you will receive your marks and be able to browse their comments and suggestions.

The success of this testing approach is in your hands...

Tell me more about OASYS...

Only one lab session remains - this is *your only chance to catch up on missed sessions* (28th Feb)

The Menu

Do a test	lab1	lab2	lab3	lab4
Mark a test	lab1	**lab2**	lab3	lab4
Revise marks	lab1	lab2	lab3	lab4
See my marks	**lab1**	**lab2**	lab3	lab4
Browse *all* marks	lab1	lab2	lab3	lab4

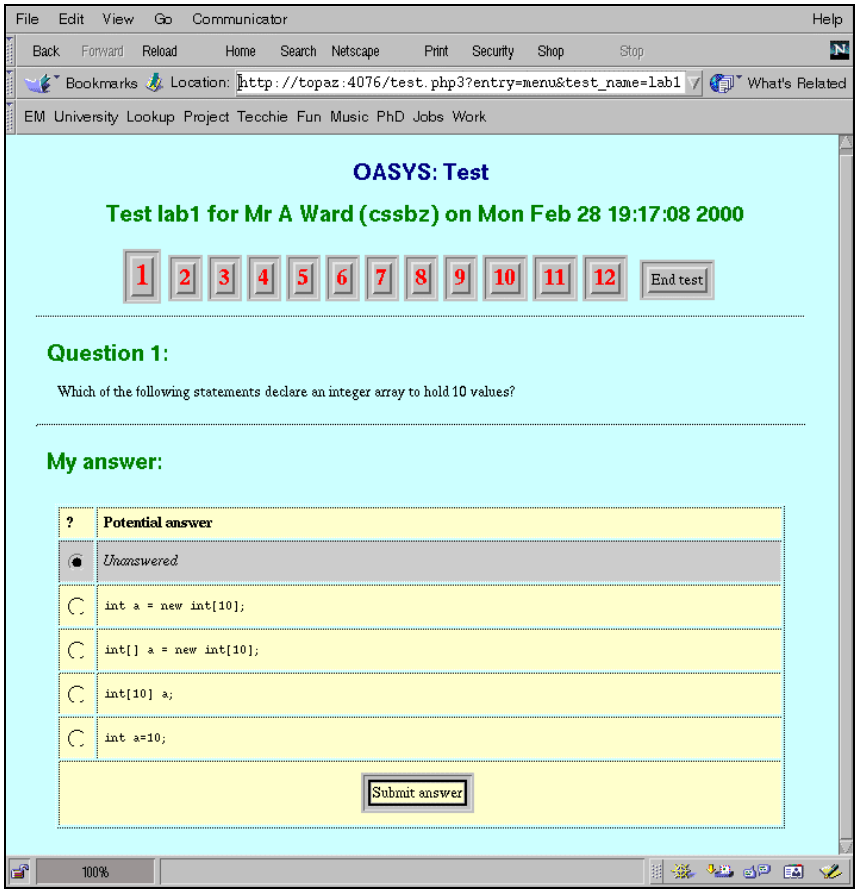
[Current system stats... Users:207 ,Scripts:641 (of which 5792 MCQ ,PMCQ and free answers) ,Free answer marks:4217]

Comments?

Comments about this system are welcome: post to the newsgroup, uwarwick.dcs.course.cs126 or email ashleyweb@dcs.warwick.ac.uk with less public issues.

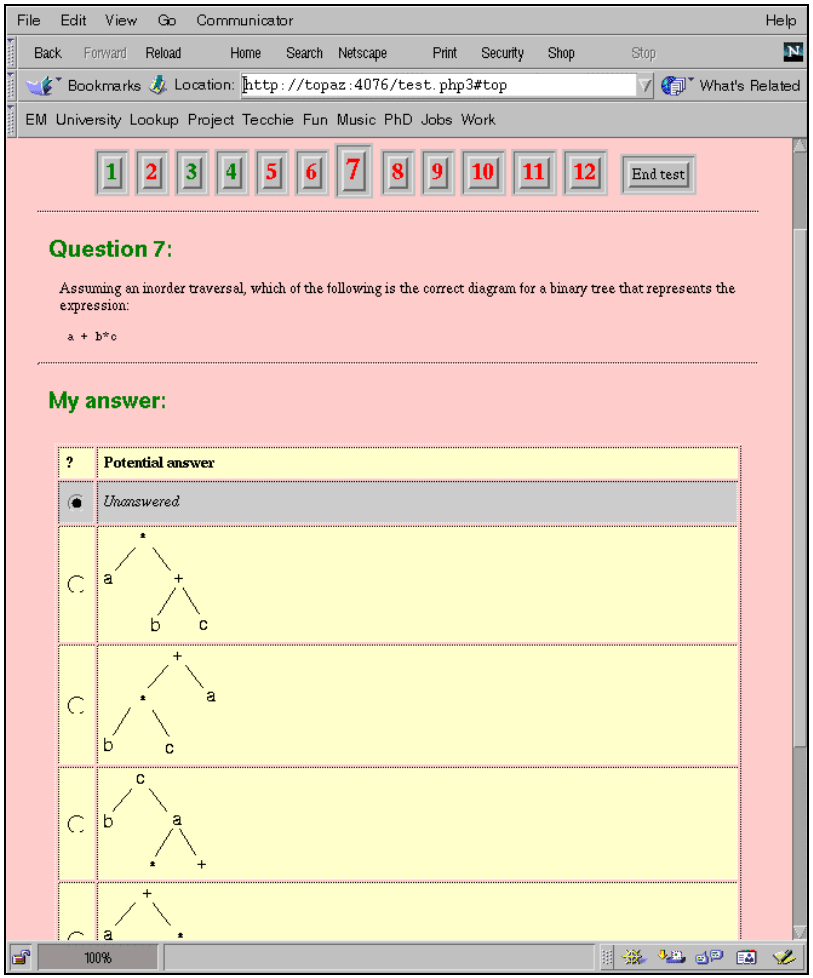
News

- 23rd Feb 10:07pm: I have mailed the people from the aborted lab session on the 21st (see below) to suggest a solution. If you are one of these people but have not received a message, please contact us ASAP. [ashley]
- 22nd Feb 8:38pm: I've fixed the problem with the "mark a test" allocation that I mention below. Demonstrators will shortly be starting work on marking scripts, so the marking shortfall should be resolved soon. Also, just for the record: there were minor problems in the lab sessions earlier today, caused by the temporary DCS web server outage. [ashley]
- 22nd Feb 13:15pm: There are only two lab sessions remaining. Please could you therefore make a

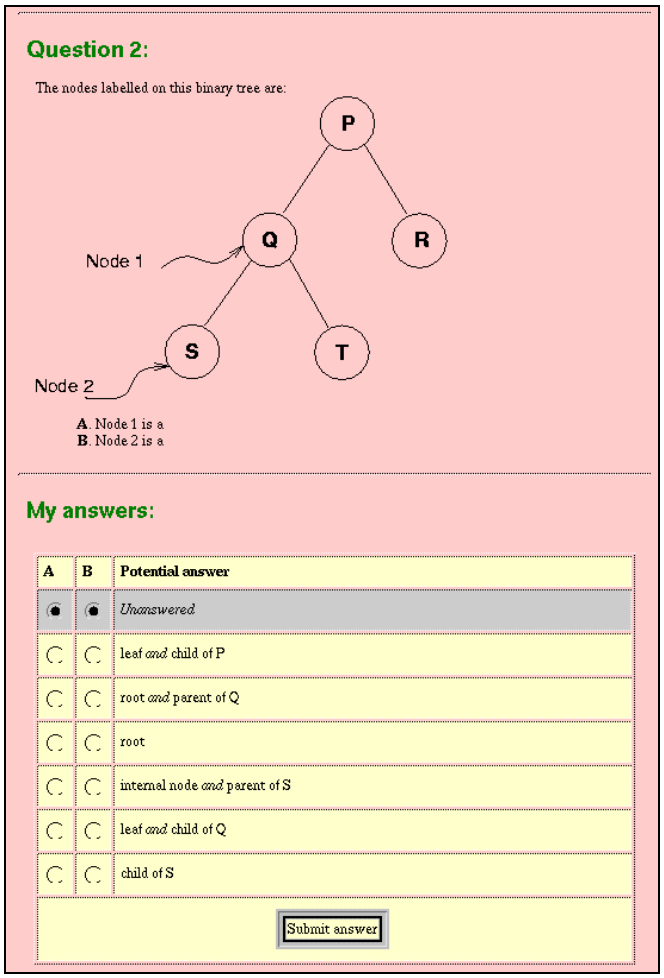


The *main menu* of OASYS (*o*n-line *a*ssessment *sy*stem) highlights the available options for this learner. Unavailable options can still be selected, and result in an explanation as to why they are not a valid option at present.

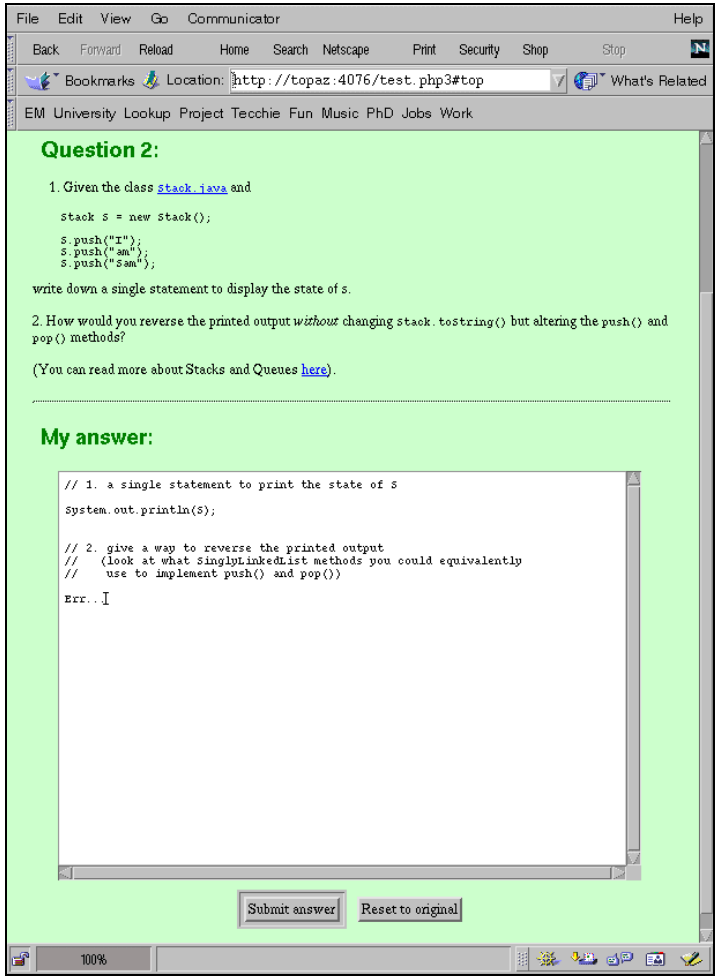
The first question in the first test is a *Multiple Choice Question* (MCQ). The background colour (which can easily be seen at a distance by invigilators!) for this test is light blue and is used to represent this test consistently throughout the system.



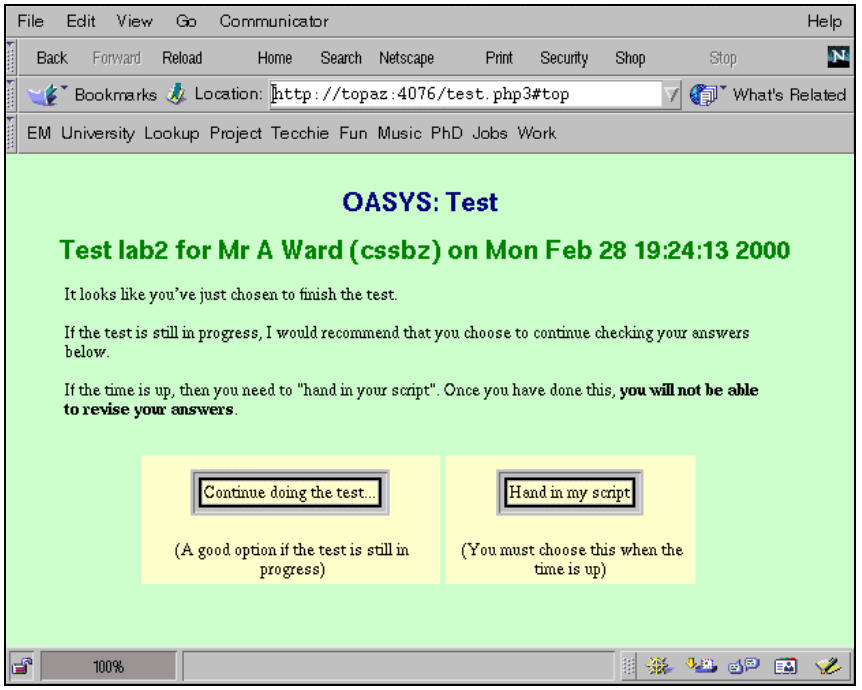
The *question navigator* is shown at the top – green buttons represent answered questions, red unanswered, and a larger button for the current question. The test need not be done in linear sequence – this learner has only answered questions 1, 3 and 4.



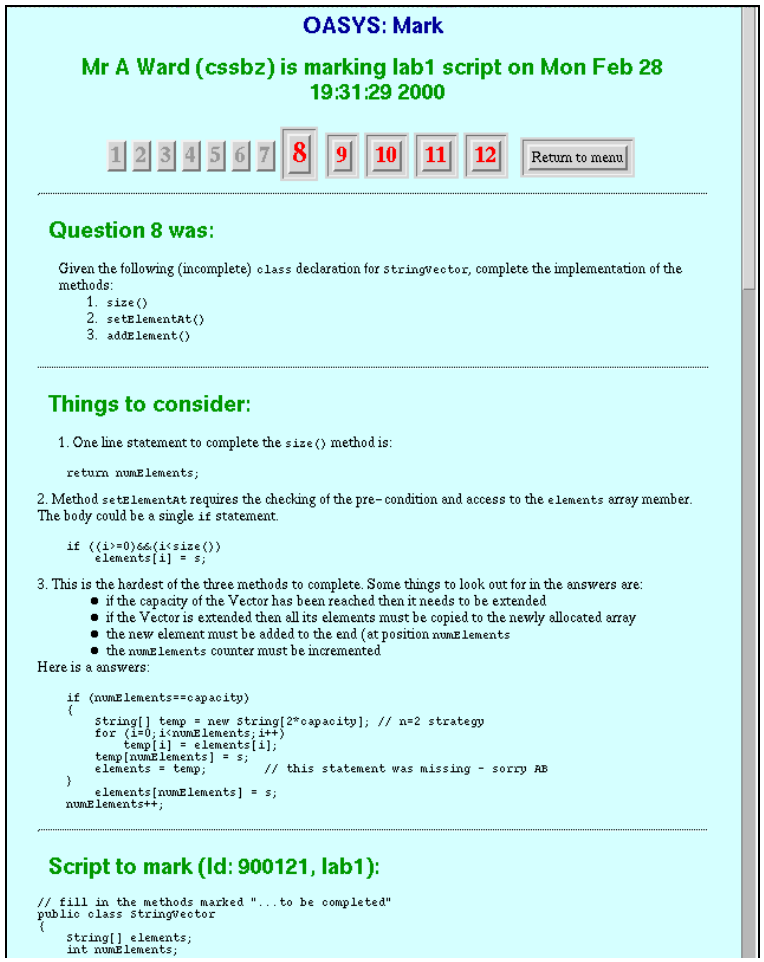
Permutational Multiple Choice Questions (PMCQs) are another question type. Potential answers are *jumbled* up "randomly" (actually, not quite – this learner will always see this question this way) so that answers cannot be copied from a nearby learner.



Entirely *free response answers* are a major feature of OASYS - free responses are marked by other learners. Also note here the use of HTML *links* to supporting material, and the use of *images* in the question and potential answers on the previous page.



When the *End Test* button is been pressed, the learner is asked to confirm their choice with this screen. After they do, the test is "handed in" and the learner can proceed with marking.



OASYS: Mark

Mr A Ward (cssbz) is marking lab1 script on Mon Feb 28 19:31:29 2000

1 2 3 4 5 6 7 **8** 9 10 11 12 Return to menu

Question 8 was:

Given the following (incomplete) `class` declaration for `StringVector`, complete the implementation of the methods:

1. `size()`
2. `setElementAt()`
3. `addElement()`

Things to consider:

1. One line statement to complete the `size()` method is:

```
return numElements;
```

2. Method `setElementAt` requires the checking of the pre-condition and access to the `elements` array member. The body could be a single `if` statement.

```
if ((i>=0)&&(i<size())
    elements[i] = s;
```

3. This is the hardest of the three methods to complete. Some things to look out for in the answers are:
   - if the capacity of the Vector has been reached then it needs to be extended
   - if the Vector is extended then all its elements must be copied to the newly allocated array
   - the new element must be added to the end (at position `numElements`
   - the `numElements` counter must be incremented

Here is a answers:

```
if (numElements==capacity)
{
    string[] temp = new String[2*capacity]; // n=2 strategy
    for (i=0;i<numElements;i++)
        temp[i] = elements[i];
    temp[numElements] = s;
    elements = temp;        // this statement was missing - sorry AB
}
    elements[numElements] = s;
numElements++;
```

Script to mark (Id: 900121, lab1):

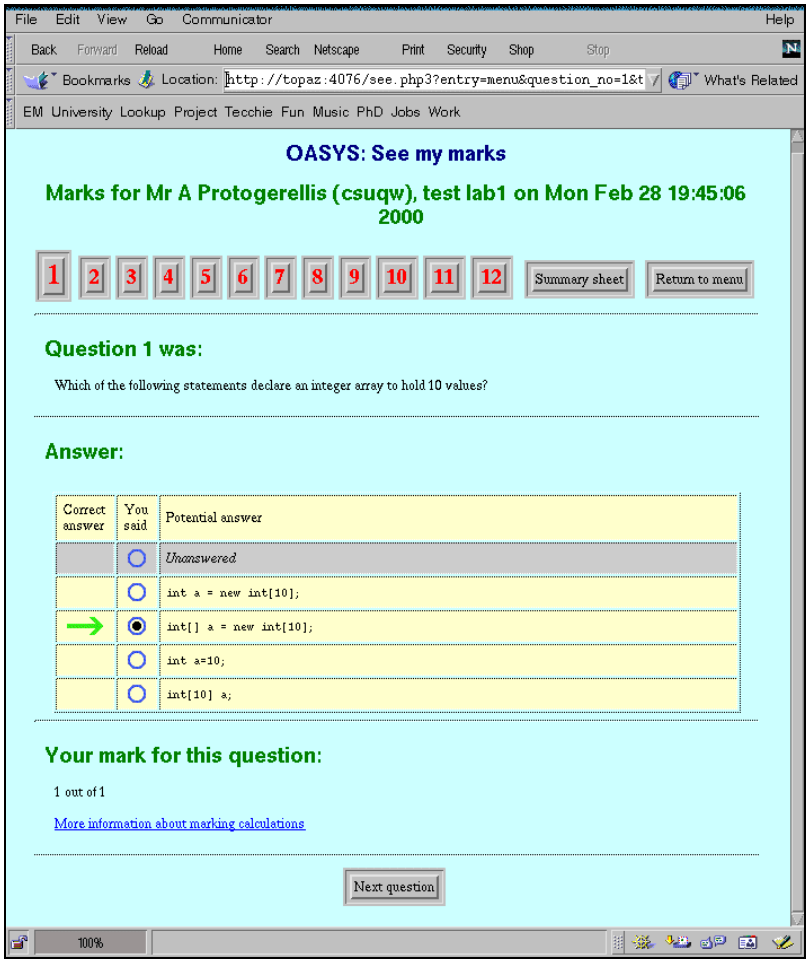
```
// fill in the methods marked "...to be completed"
public class StringVector
{
    string[] elements;
    int numElements;

    public StringVector(int capacity)
    // pre: capacity>=0
    // post: construct an empty vector of strings with size capacity
    {
        elements = new String[capacity];
        numElements = 0;
    }

    public int size()
    {
        return (numElements);
    }

    public void setElementAt(String s, int i)
    // pre: ...this is referred to in an another question...
    // post: replaces element at i with s
    {
        if ((i>=0) & (i<numElements))
                elements[i] = s;
    }

    public void addElement(String s)
    // post: append element to end of vector extending it if necessary
    {
        if(numElements==elements.length)
        {
                String oldElements[] = elements;
                elements = new String[elements.length*2];

                for(int i=0; i<oldElements.length; i++)
                {
                        elements[i]=oldElements[i];
                }
        }

        elements[numElements] = s;

        numElements++; // keep track of vector size

    }
}
```

My marks:

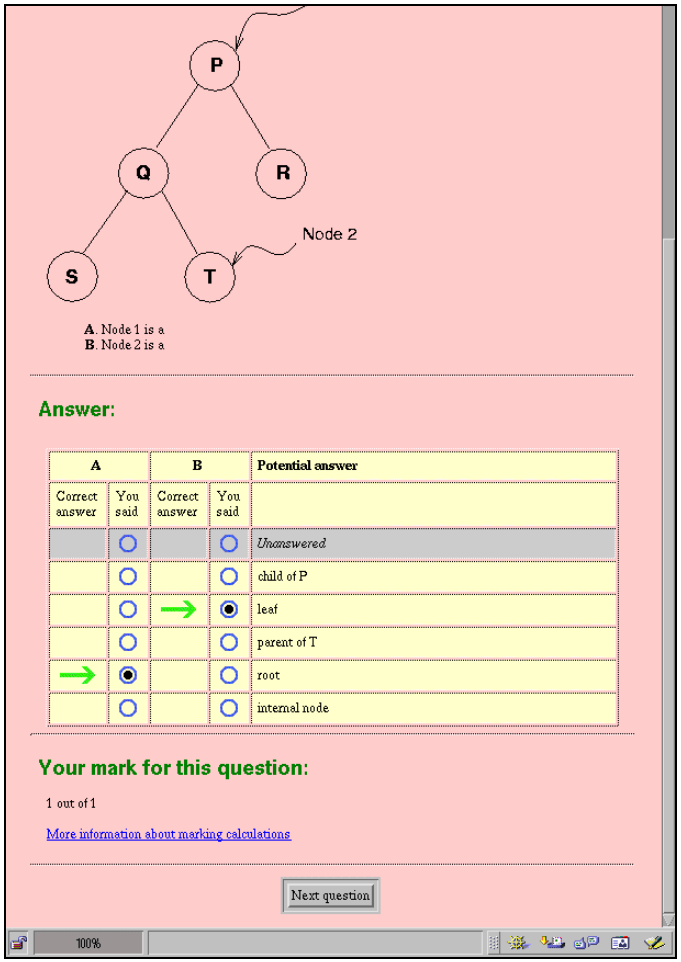
Readability	Excellent ◉ ○ ○ ○ ○ Poor (○ unmarked)
Correctness	Excellent ○ ◉ ○ ○ ○ Poor (○ unmarked)
Style	Excellent ◉ ○ ○ ○ ○ Poor (○ unmarked)
Suggestions	Excellent! I like your use of 'oldElements': makes what you are doing very clear. Unfortunately, I think ((i>=0) & (i<numElements)) is incorrect: & is a bit-wise operator - in this case && is required.

Submit marks | Reset to original

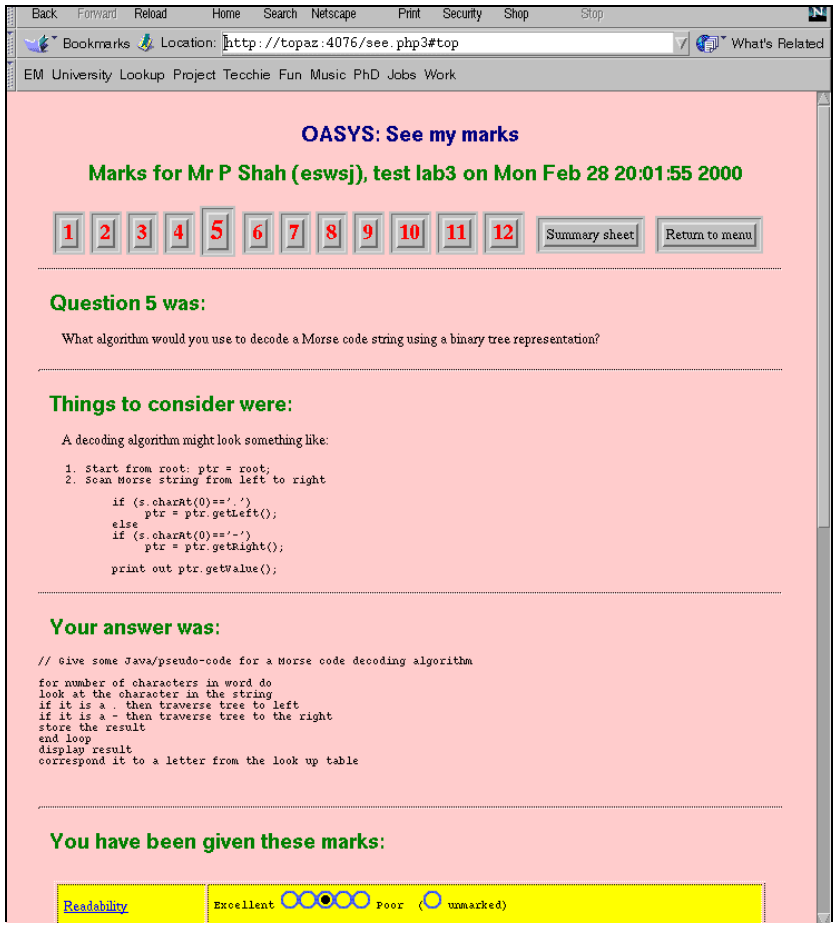
Learners are required to mark three scripts to gain full credit. The *marking page* shows the original question; advice on the correct use of the mark scheme for this question; the answer to be marked (which was previously entered by another learner) and finally the *marking interface*. Notice that answers to the multiple choice questions (in this test, questions 1-7) are *automatically marked* and so cannot be selected in the question navigator – feedback on MCQ answers is given in the "see my marks" interface, described next.



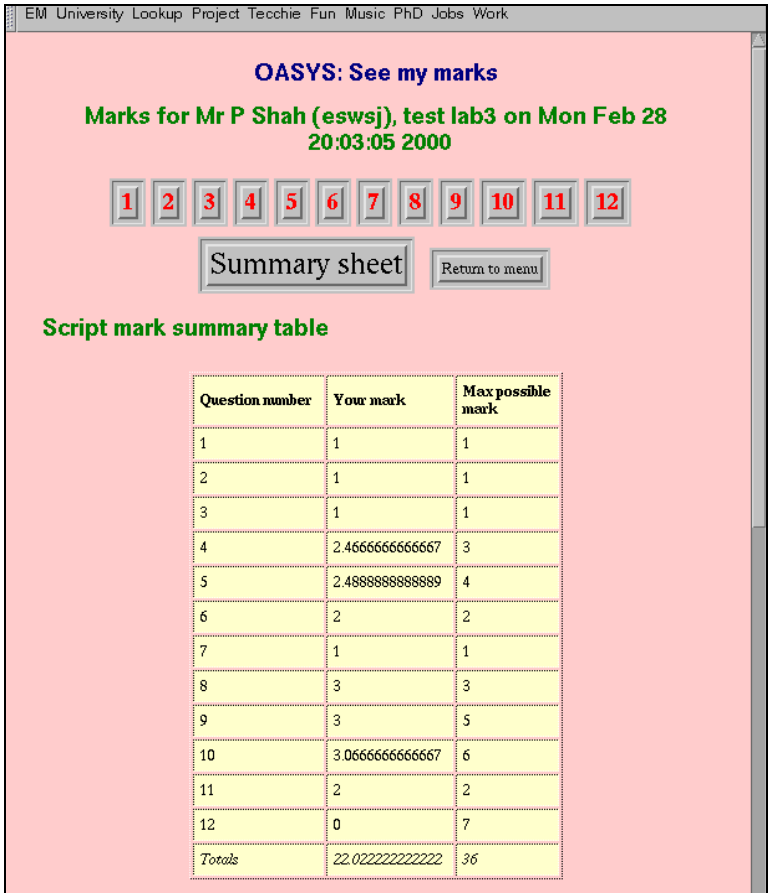
"*See my marks*" is where learners obtain feedback on their own scripts, and can be viewed at *any time* from *anywhere* on the Internet. Marking can also be done in learners' own time. The MCQ is "jumbled" the same way as the learner originally saw it.

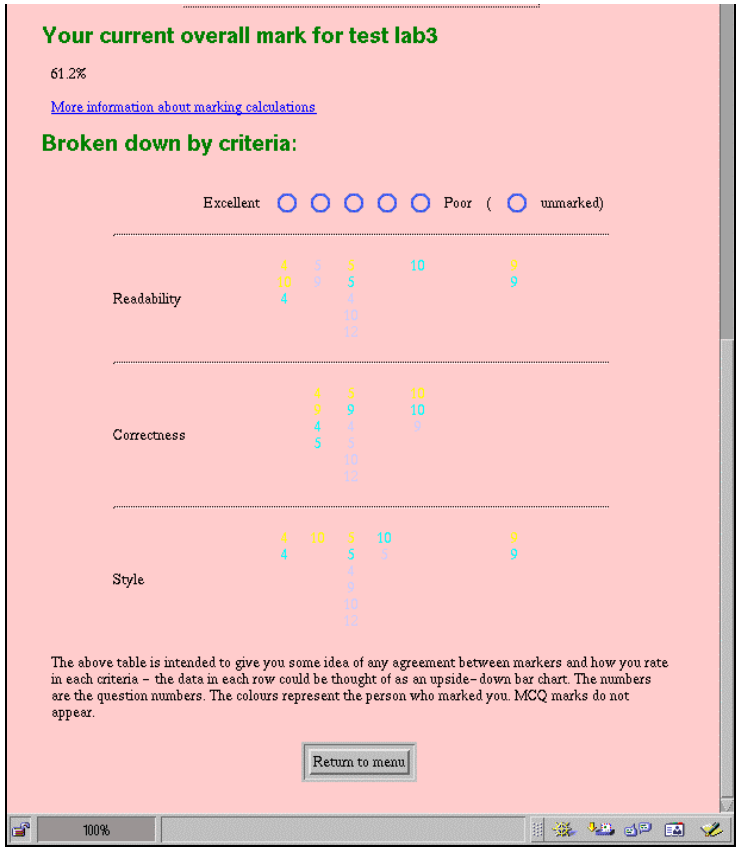


PMCQs attempt to minimise the probability of a correct response from a simple guess, and so the learner receives all the points available (as set in a *mark scheme*) if they are correct in both answers, and no points if either or both answers are incorrect.

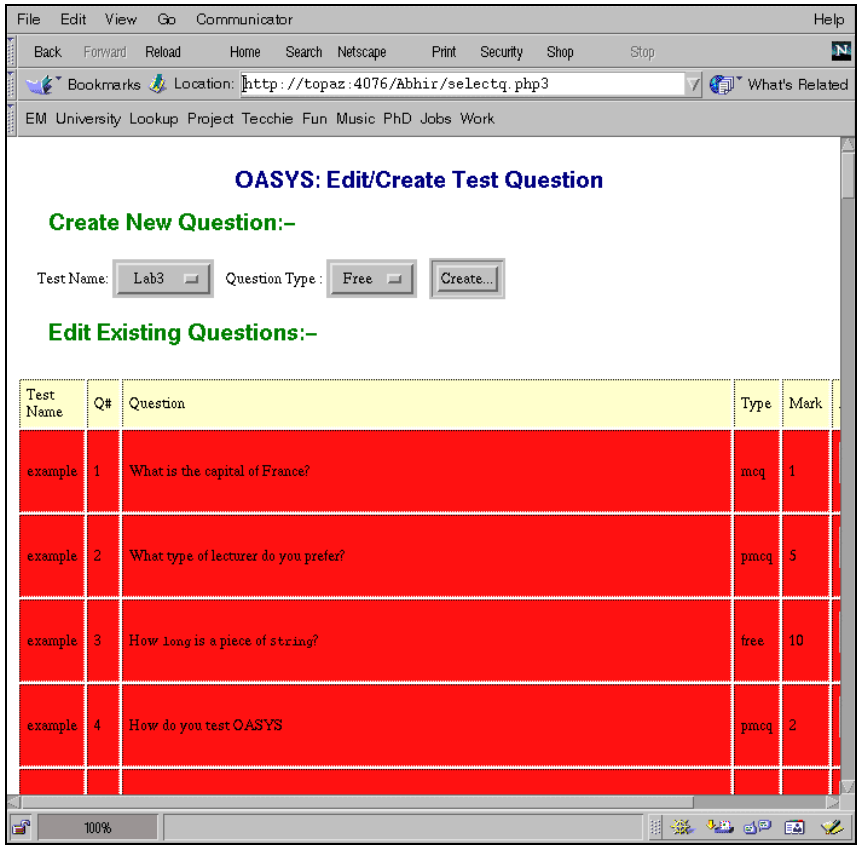


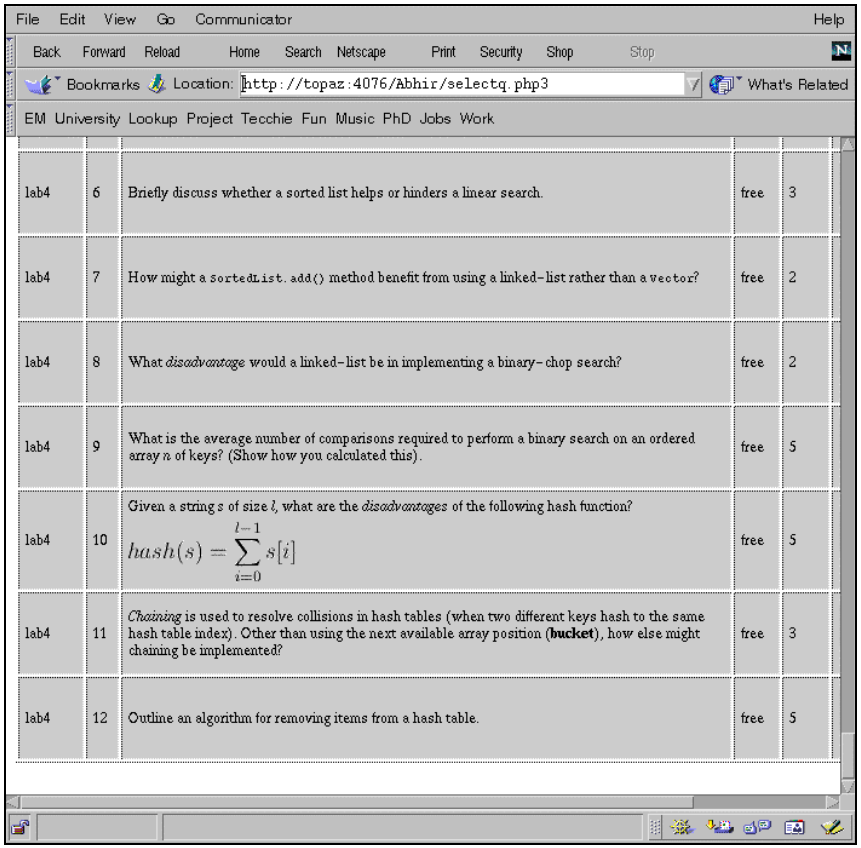
Colours are used to provide cues throughout OASYS: here, colours are used to represent the *anonymous* markers of a script. The three learners who performed this marking seem to have given similar opinions, and one has entered a *feedback comment*. The final mark for the question is derived from the marks given, the maximum possible being set (by the administrator) in the mark scheme.





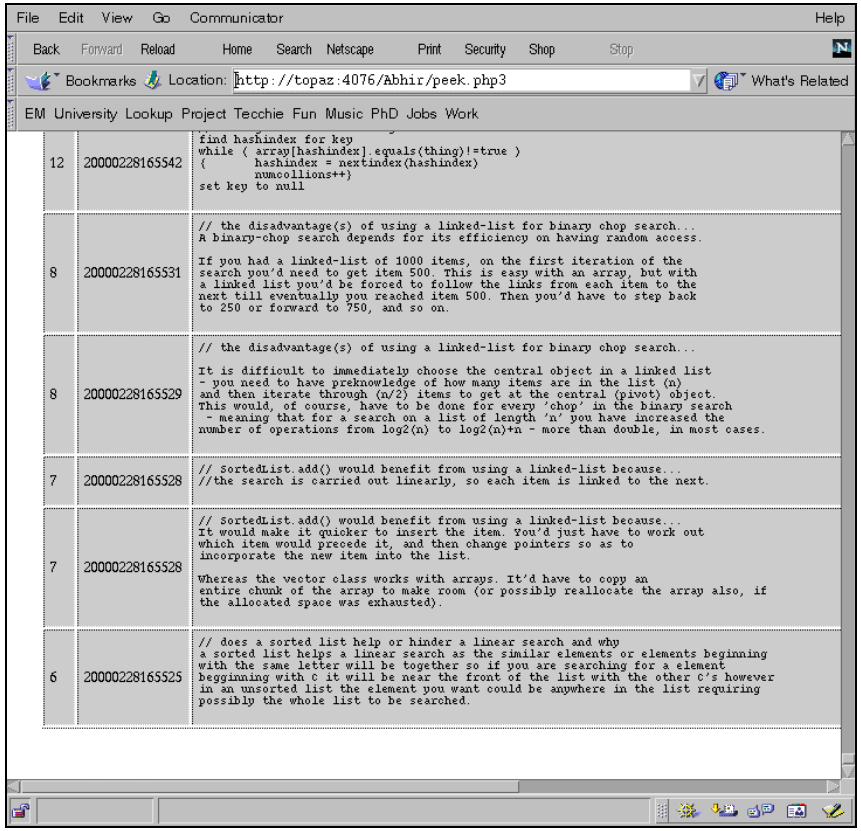
All the marking information about a script is collated for the learner in the "*summary sheet*". This particular learner has quite a good overall result, but the overall spread of marks given by other learners (shown in the "*break down table*") seems to be fairly wide, despite the lack of variance in marking by the learner represented in mauve.



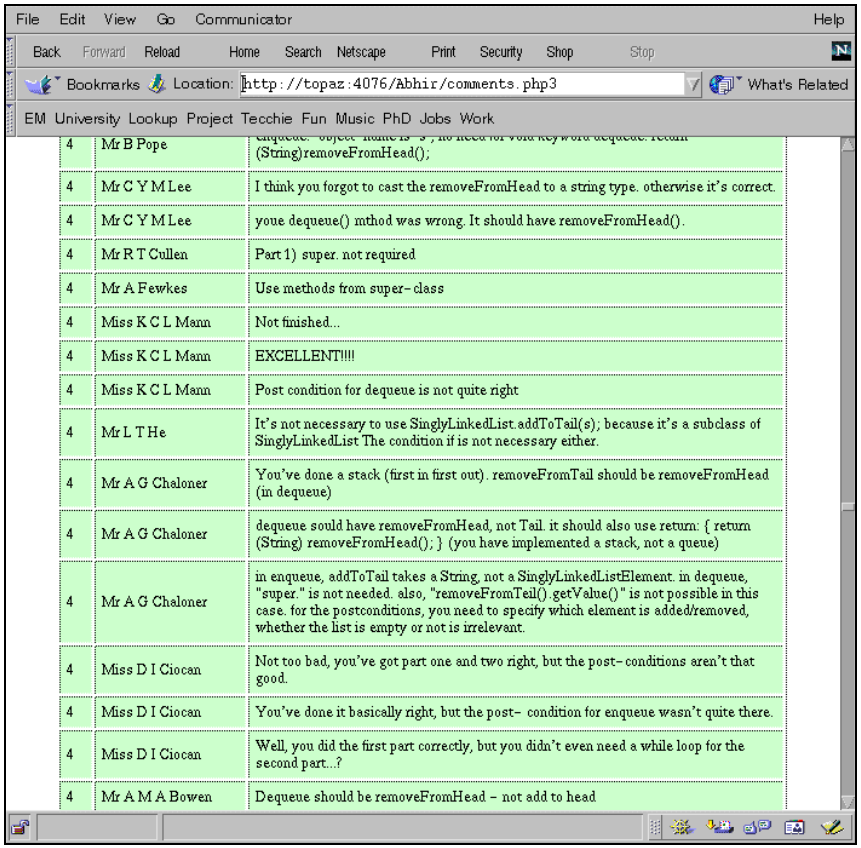


At the *administration* end, some common tasks, such as *creating and editing questions*, are eased by the provision of a web based interface. The stark red background here distinguishes the *example test*, which is not seen by normal learners.

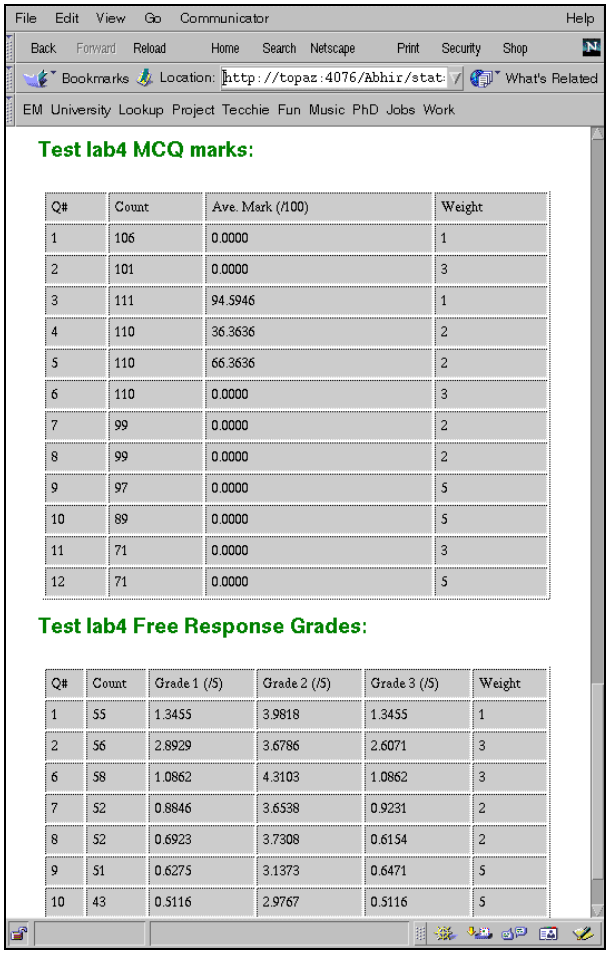
Questions can use mathematical symbols, but these must be created as images as current web browsers do not support these symbols directly.



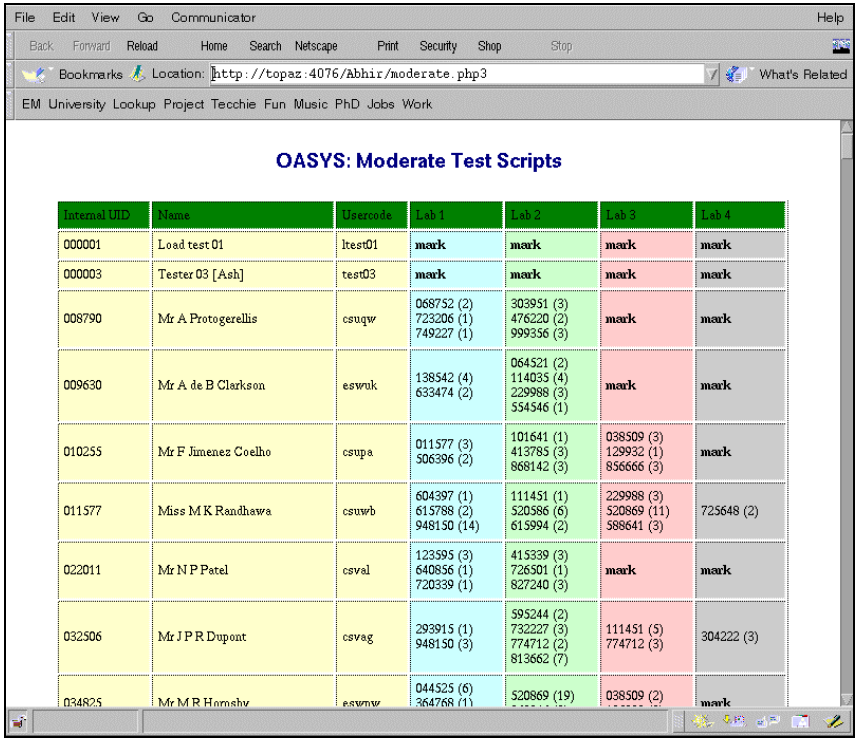
Administrators can browse learners' responses to questions…



… and also their feedback comments entered whilst marking.



Statistics can be generated in *real time* from the *database*. Question 4 (an MCQ) seems to have mislead the majority of learners who have done the test so far (there are approximately another 80 learners still to complete this test at this point in time).



Lastly, an interface allowing *moderation* is currently under development (this screen shot is of a prototype).

Thank you for browsing this document.