Question 1  [25 marks]

The purposes of this question are:

(i) to remind you of some basic Pascal programming techniques;

(ii) to ensure that you are able to access course team provided software;

(iii) to test your understanding of some of the advanced Pascal features discussed in Unit 1.

The early parts of the question is designed to enable you quickly and easily to tackle the practical exercises in the remainder of the course and should identify programming and computer system difficulties; they do not represent a major theme of the course. Abstract data types are, however, a major theme of the course.

(i) An analytical chemist performs experiments on unknown substances and compares her results with those of known compounds. In one experiment she places a small amount of a substance in an analyzing machine and obtains a colour spectrum as the result. A colour spectrum is a list of colours which characterize the substance. There are five possible colours, which are

- red, orange, yellow, green, blue.

A substance's colour spectrum consists of at least one colour and can contain all five colours. A typical output from the machine is

- orange, yellow, blue.

The chemist has a table of the colour spectrums of many known compounds which she uses to determine the identity of the unknown substance.

For the purpose of this question, assume that the table of known colour spectrums contains information about 8 compounds designated A to H, and is to be represented as an array of Boolean values, named table, as follows.

<table>
<thead>
<tr>
<th></th>
<th>red</th>
<th>orange</th>
<th>yellow</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>B</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>C</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>D</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>G</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>H</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

So, for example, known compound B has a colour spectrum of red, yellow, and green, indicated by the T(ue) values in the table.
Your task is to write part of a Pascal program which will input the colour spectrum of an unknown substance and compare it with the table of colour spectrums of known compounds. Here is a design of for the program.

1 initialize the table of compounds
2 read in the colour spectrum of an unknown substance
3 search through the table looking for a match
4 if a match is found
5 then
6 write out the letter code of the matched compound
7 else
8 write out the message 'Substance unknown'

(a) Write down the declaration of a user-defined ordinal type named colour to represent the five spectrum colours (keep the order of colours the same as that shown in the opening paragraph of the question).  

[2 mark]

(b) The rows of a two-dimensional array are to be indexed by the lower-case characters 'a' to 'h'. Write down a Pascal subrange type named compound to represent such an index.

[1 mark]

(c) Write down the declaration of a two-dimensional array type named colourtable indexed by variables of type compound and by variable of type colour, as shown in the figure.

[2 marks]

(d) Refine Step 1 of the program design given above. Initialize the array table with the data given in the figure above. Use nested for loops to initialize the whole of the array to false and then use assignment statements to fill in the true values.

[2 marks]

(e) Refine Step 2 as follows. The colour spectrum of an unknown substance is to be held in a variable named substance whose type is set of colour named spectrum. For input, the user types the first letter of each of the colours making up the spectrum; any character which is not the first letter of a spectrum colour terminates the sequence. For example, the spectrum yellow, green, blue is input as the sequence y, g, b and z (say). You should use a case statement to enter the appropriate colour value into the set substance.

[4 marks]

(f) Refine Steps 3 to 8. You should assume the existence of a procedure named search with the following heading.

procedure search(table: colourtable; substance: spectrum; var found: Boolean; var name: compound)

This procedure returns the value true (in found) if there is a known compound with the same spectrum as the unknown substance, and the parameter name holds the identifying letter of the matched compound. If there is no match, the parameter found is set to false.

[3 marks]

Note: do NOT attempt to write the procedure search.
(g) Input and test your completed program. The procedure \textit{search} is held in the course file D1, in a compiled unit named \textit{usearch} held in the file named \textit{usearch.code}, and to make use of it you should include the statement

\begin{verbatim}
uses{$u usearch.code}usearch;
\end{verbatim}

into your program as the first statement after the \textbf{program} statement. Do not forget to set the prefix of your working directory to the directory in which \textit{usearch.code} is stored.

\textbf{Note}: the declarations of the types \textit{colour}, \textit{spectrum}, and \textit{colourtable} are included in the unit \textit{usearch} and, therefore should \textbf{NOT} be in your main program.

[6 marks]

For Part (g) you must submit to your tutor a computer listing of your program together with the output for the following sets of data:

(a) the spectrum of the first test substance: red, green, blue.

(b) the spectrum of the second test substance: red, orange.

(c) the spectrum of the third test substance: orange.

(ii) The analysing machine mentioned in (i) has a brand name SAM. A SAM has a push-button on-off switch, a drawer for holding a sample, a push-button for opening and closing the sample drawer, a push button for starting the analysis of a sample and displaying the results of the analysis.

Describe how a SAM can be characterized as:

(a) an abstract data type; [3 marks]

(b) a user-defined type in Pascal (making suitable use of ordinal types) [2 marks]
Question 2 [25 marks]

The question tests your understanding of the materials in Unit 2. The abstract data type stack is defined on page 23.

(i) Describe, in your own words, the way in which the meanings of the operations PUSH and POP are related. [3 marks]

(ii) Complete the following axioms:

\[
PUSH(TOP(s), POP(s)) =
\]

\[
POP(PUSH(i, CREATESTACK)) =
\]

[3 marks]

(iii) Why is the following proposed axiom incorrect?

\[
ISEMPTYSTACK(TOP(PUSH(i, s))) = false
\]

[2 marks]

A new stack operation SIZE is to be defined. SIZE takes a stack and returns the number of items in the stack.

(iv) Give the syntax for SIZE, defining any new sets you may need. [3 marks]

(v) Give two axioms which define SIZE in terms of CREATESTACK and PUSH. [4 marks]

(vi) Using the underlying model of a list, give pre- and post-conditions for the operation SIZE. [4 marks]

(vii) Using the same stack, s1 as given below, each time; state the values of the following:

\[
TOP(s1)
ISEMPTYSTACK(POP(s1))
POP(POP(S1))
PSUH(elm, POP(S1))
SIZE(PUSH(elm, S1))
\]

[6 marks]

The stack s1 contains:

```
dogwood
cedar
beech
ash
```
Question 3 [25 marks]

This question tests your understanding of the specification (using the constructive approach) and representation of abstract data types.

(i) On pages 26 and 27 of Unit 3 the formal specification of the abstract data type queue is given, using the constructive approach. You are now asked to make various changes to this formal specification.

(a) A new operation INRANGE is to be introduced. INRANGE takes a queue as source data and returns as the result the value true, when the queue contains between 10 and 20 items inclusively, or the value false when the number of items in the queue is outside this range. Write down the changes to the formal specification needed to incorporate INRANGE.

(b) A new operation NEXTTOFRONT is included in the formal specification. Its syntax is

\[
\text{NEXTTOFRONT}: Q \rightarrow 1 \cup \text{E}
\]

and its semantics are

\[
\text{pre-NEXTTOFRONT}(q) := \text{true} \\
\text{post-NEXTTOFRONT}(q; r) := \begin{cases} \\
\text{IF } q = \text{createlast} & \text{THEN} \\
\text{r} = \text{error - empty queue} \\
\text{ELSE} & \text{IF } \text{length}(q) = 1 \\
\text{THEN} & \text{r} = \text{first}(q) \\
\text{ELSE} & \text{r} = \text{first}(\text{trailer}(q)) \\
\end{cases}
\]

Write down in your own words an informal description of NEXTTOFRONT.

(ii) Section 6.3 of Unit 3 examines a cursor-based representation of a queue based on a circular structure where one way of distinguishing a full queue from an empty queue is never to let the queue contain more than max_size-1 items. Using this approach, draw diagrams similar to Fig 6.5 (or part thereof) which show the relative positions of head and tail when the queue is:

(a) Empty;

(b) Full.

(iii) Continuing with the approach described in part (ii) and using the data structure declarations given at the start of Section 6.4

(a) Write down the function body for isemptyqueue declared as:

\[
\text{function isemptyqueue}(q: \text{queue}): \text{Boolean}
\]
(b) Write down the procedure body for `addtoqueue` declared as:

```plaintext
procedure addtoqueue(i: item_type; var q: queue)
```

[4 marks]

(c) Write down the procedure body for `deletefromqueue` declared as:

```plaintext
procedure deletefromqueue(var q: queue)
```

[3 marks]

Note: Each of the routines in (a), (b), and (c) must make use of the function `movercursor` defined on page 32 of Unit 3.

Question 4 [25 marks]

This question tests your understanding of the abstract data types BinarySearch Tree, and is primarily based on the material of Section 8.4 of Unit 4.

Section 8 of Unit 4 does not discuss the deletion of items from a BSTree. This question asks you to investigate the following axiomatic specification for an operation, named `DELETE`, that removes an item (if it exists) from a given tree. In the specification, `e` is the item to be deleted and `LARGEST` is an operation that takes a BSTree as source data and returns as the result the item with the largest value in the source tree, with SYNTAX:

\[
\text{Largest: } T \rightarrow I
\]

Here is the specification of `DELETE` with axioms labelled (A) and (B):

(A) \(\text{DELETE(CREATETREE, e)} = \text{CREATETREE}\)

(B) \(\text{DELETE(MAKETREE(l, i, r), e)} =\)

\[
\begin{align*}
&\text{IF } e = i \\
&\quad \text{THEN} \\
&\quad \text{IF } \text{ISEMPTYTREE}(l) \\
&\quad \quad \text{THEN} \\
&\quad \quad r \\
&\quad \text{ELSE} \\
&\quad \quad \text{IF } \text{ISEMPTYTREE}(r) \\
&\quad \quad \quad \text{THEN} \\
&\quad \quad \quad l \\
&\quad \quad \text{ELSE} \\
&\quad \quad \quad \text{MAKETREE(DELETE(l, LARGEST(l), LARGEST(l), r)} \\
&\text{ELSE} \\
&\quad \text{IF } e < i \\
&\quad \quad \text{THEN} \\
&\quad \quad \text{MAKETREE(DELETE(l, e), i, r)} \\
&\quad \text{ELSE} \\
&\quad \text{MAKETREE(l, i, DELETE(r, e))}
\end{align*}
\]

(i) If `l` is a BSTree, express in your own words the meaning of

\(\text{DELETE(l, LARGEST(l))}\). [2 marks]
(ii) For the following BSTree, identify the two BSTrees (l and r) and the item (i) which, when input as source data to the operation MAKETREE, result in the following tree.

```
    37
    / \ \
  15   47 \
    / \ \
25  42  56 \
    / \     \
39  
```

[3 marks]

(iii) If l and r are the left and right subtrees of the tree shown in Part (ii), draw the tree represented by the following expression.

```
MAKETREE(DELETE(l, LARGEST(l)), LARGEST(l), r)
```

[5 marks]

(iv) Show how the axioms given above define the actions of DELETE by using them to delete the item, 16, from the following BSTree. Your solution must show the left and right subtrees and the root item at each stage, and must show how the recursive invocations of the axioms build up the final tree.

```
  10  
    \  
    16  
    / \ \
  13  18 \
    /     \
  12  
```

**Hint:** Examine Example 8.2 beginning on Page 20 of Unit 4.

End