Functional Programming, COMP0020 (A6U, A7P)
Sample paper, April 2024

Suitable for Cohorts: 2023/24, 2022/23, 2021/22, 2020/21

Answer BOTH questions. Your answers will be marked for correctness, completeness and quality (including appropriate use of algebraic types, type synonyms, list comprehensions, higher order functions and partial applications; appropriate error checking; type definitions for functions; coherence, elegance, clarity and understandability; commentary and brevity).

Marks for each part of each question are indicated in square brackets.

Standard calculators are permitted.

1. (a) Explain what the following Miranda algebraic type represents:

```
fred * ::= Empty | Node * [fred *]
```

[4 marks]

(b) Consider the following Miranda code (the ! operator provides an index into a list counting from zero, for example [1,2,3]!0 returns the value 1):

```
t_graph * ::= Emptygraph | Node * [t_graph *]

glist :: [t_graph char]
glist = [ Node 'A' [glist!2, glist!1],
          Node 'B' [glist!3],
          Node 'C' [glist!0, glist!3],
          Node 'D' []
        ]

graph :: t_graph char
graph = hd glist
```

Give a diagram to illustrate the data structures glist and graph. [8 marks]

(c) The predefined Miranda function member is defined as:

```
member :: [*] -> * -> bool
member [] x = False
member (x:xs) y = (x=y) \/ member xs y
```

Using the function member, give the definition for a Miranda function called printgraph that takes an argument of type t_graph char and returns a list of characters representing the data structure. The word "empty" should be used to signify an empty graph node. Your program must not loop forever; if a Node has been encountered previously, the word "seen" should be printed instead of its list of successor nodes.

For example, the application (printgraph graph) should return the following result (NB a node that is shared and appears in different branches may appear multiple times in the output):

```
Node A [ Node C [ Node A seen, Node D empty], Node B [Node D empty] ]
```

[27 marks]

[Total for Question 1: 39 marks]

2. Provide Miranda code to implement the following simple game, which shuffles a collection of an arbitrary number of playing cards. Do **not** write a complex user interface.

   A “shuffle” function should take as arguments a number, a list of elements and a function that can be applied to two cards to determine if they are equal. Each element in the list represents a playing card, where each playing card has a “suit” (“Spades”, “Hearts”, “Diamonds” and “Clubs”) and a “number” (there are thirteen possible numbers, with the top four in increasing order being “Jack”, “Queen”, “King” and “Ace”).

   The Miranda code should produce as its output a shuffled version of the input list (the second argument). The function should shuffle the list of cards as many times as is indicated by the first argument. The action of shuffling should cut the deck in half and then interleave the cards, with the previous top card now being the second card in the pack. For example, if a list of four items A, B, C and D is shuffled once the result should be C, A, D, B. If that result is shuffled again the result should be D, C, B, A.

   If the list contains an odd number of cards, you should detect this case and provide an appropriate solution. You should explain your solution – what it does and why. Your function should detect the following two errors and provide appropriate error handling:
   (i) where there are more than 52 elements in the input list, and
   (ii) where there is any duplicated card in the input list.

   Next provide Miranda code to deal four hands of five cards from a shuffled pack of 52 cards, then to determine whether there is a winning hand and if so which hand would win according to the following rules:

   - The best hand is a “straight flush” where all five cards are in the same suit, and where the five values make a sequence with no gaps. If two or more players have a straight flush, the one that wins is the one with the highest top-ranked card (an Ace is the highest ranked card). If two or more players have identically high straight flushes, the hands are ranked by suit in descending order: Spades, Hearts, Diamonds, Clubs.
   - Second-best is “four of a kind”: four of the same-valued cards (one from each suit). If two or more players have four of a kind the winner is the one with the highest value.
   - Third-best is a “full house” that contains three cards of one value and two cards (a pair) of another value (e.g. 3 Kings and 2 Jacks). Hands are ranked first by the value of the triplet, and then by the value of the pair.
   - Fourth-best is a “flush” that contains five cards all of the same suit. Hands are ranked firstly by the value of the highest card, then of the second card, and so on.

   Your answer will be marked for correctness, completeness and quality (including Appropriate use of algebraic types, type synonyms, list comprehensions, higher-order Functions and partial applications; appropriate error checking; type definitions for functions; coherence, elegance, clarity and understandability; commentary and brevity).

[61 Marks]
[Total for Question 2: 61 marks]