**3.9 Homework Task Submission**

|  |  |
| --- | --- |
|  |  |
| **Answer all parts of this Homework task and submit your work into VSV Online as PDF and SnapApps/Edgy .xml or Python3/Trinket code files. This homework is required to demonstrate learning outcomes to a satisfactory standard.**  **For this Homework Submission 2 Files are expected:**  **• Question 1: a PDF file with the algorithm for solving the problem in pseudocode.**  **• Question 2: an exported coding file in .xml from SnapApps/Edgy or Python3/Trinket export creates the model and then finds a path using recursion from initial state to the goal state** |

|  |  |
| --- | --- |
| **There is family unit of a man and woman of equal weight 80kg, together with their two children, where each child weighs 40kg. This family wishes to cross a river in their little boat, which can only bear the weight of 90kg.** | Image result for family unit |
| **A pictorial solution for this planning problem for the family crossing the river from initial state to goal state is shown below:** | |
|  | |

**In this Homework Task you are required to:**

1. **Write an Algorithm in structured pseudocode that:**

* Generates all legal states for this planning problem stored in Abstract Data Types using a **Brute Force** approach.
* Uses conditional logic to determine and connect legal transitions from state to state. This method is known as **Generate and Test**.
* Finds a path from initial state to the final state by a **Recursive Depth First Search** algorithm.

The algorithm in structured pseudocode for recursive Depth First Search (DFS) is:

**Algorithm Depth-First-Search**(Inputs node *c, node target* )

If (*c* is equal to the target) then

Return

Else

Mark *c* "Visit In Progress"

For each neighbor *n* of *c do*

If (*n* is "Unvisited") then

**Depth-First-Search**( *n, target* )

end if

end do

Mark *c* as "Visited"

end if

**End Algorithm**

1. **Code and implement your algorithm for all steps from part (1) in SnapApps/Edgy or Python3/Trinket.**