**Little Heath Sixth Form**

**Mathematics (Decision 1)**

 Personal Learning Checklist

**Student Name: ……………………….…………………………………..………**

|  |  |
| --- | --- |
| **Unit Name:****Mathematics (Decision 1)** | **Unit Code:****MD01** |
| *Minimum Target Grade:* | *Aspirational Target Grade:* |

*KEY:* ***Red =*** *with difficulty* ***Amber*** *= not sure* ***Green*** *= yes*

|  |  |  |  |
| --- | --- | --- | --- |
| **GCSE Re-Cap (Skills and Knowledge)** | **Red** | **Amber** | **Green** |
| * Able to draw a linear graph of the form ax + by = c
 |  |  |  |
| * Identify regions on a graph of the form eg y>ax +c, 2y<x+1
 |  |  |  |
| * Use an inequality test to identify shaded regions
 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skills/Knowledge/Specification** | **Red** | **Amber** | **Green** | **To address this before the exam I will:-** |
| **Algorithms** |  |  |  |  |
| * Able to implement a given flow chart of an algorithm
 |  |  |  |  |
| * Able to carry out the bubble sort algorithm on a set of numbers or words
 |  |  |  |  |
| * Able to carry out the quick sort algorithm on a set of numbers or words using pivots recording your procedure
 |  |  |  |  |
| * Able to carry out the Shuttle sort algorithm on a set of numbers or words recording the procedure
 |  |  |  |  |
| * Able to carry out the Shell sort algorithm on a set of numbers or words recording the procedure
 |  |  |  |  |
| * Able to calculate the number of passes or swaps needed for a given sort
 |  |  |  |  |
| **Spanning Tree Algorithms** |  |  |  |  |
| * Able to define a tree, network, cycle, path and minimum spanning tree
 |  |  |  |  |
| * Able to apply Kruskal's algorithm to a network to find the minimum spanning tree
 |  |  |  |  |
| * Able to apply Prim's algorithm to a network to find the minimum spanning tree
 |  |  |  |  |
| * Able to apply Prim's algorithm to a network represented by a matrix
 |  |  |  |  |
| * Able to use Dijkstra's algorithm to find the shortest path between two vertices
 |  |  |  |  |
| * Able to track back from T to S to find the shortest route showing your method
 |  |  |  |  |
|  |  |  |  |  |
|  | **Red** | **Amber** | **Green** | **To address this before the exam I will:-** |
| **Graphs and Networks** |  |  |  |  |
| Know and use the language of graphs ie vertices, edges, edge weights, paths, cycles |  |  |  |  |
| Know the definition of a simple graph |  |  |  |  |
| Find adjacency or distance matrices |  |  |  |  |
| Understand what is meant by a connected graph |  |  |  |  |
| Understand the difference between a directed and undirected graph |  |  |  |  |
| Understand the degree of a vertex and indentify odd and even vertices |  |  |  |  |
| Find Eulerian trails |  |  |  |  |
| Find Hamiltonain cycles |  |  |  |  |
| Understand the concept of a tree |  |  |  |  |
| Understand the concept of a bipartite graph Kn and Kn,m |  |  |  |  |
| **Route Inspection (Chinese Postman)** |  |  |  |  |
| * Able to inspect a graph to see if it is traversable
 |  |  |  |  |
| * Able to use the route inspection algorithm to traverse the network in the shortest distance
 |  |  |  |  |
| * Able to use the route inspection algorithm to start and end at different vertices
 |  |  |  |  |
| **Travelling salesman problem (TSP)** |  |  |  |  |
| * Know the differences between the classical and practical problems
 |  |  |  |  |
| * Convert a network into a table of minimum distances
 |  |  |  |  |
| * Find a minimum spanning tree using Prims or Kruskals algorithms
 |  |  |  |  |
| * Find an initial upper bound for the TSP
 |  |  |  |  |
| * Use short cuts to improve upper bounds
 |  |  |  |  |
| * Use a minimum spanning tree to find a lower bound
 |  |  |  |  |
| * Use nearest neighbour algorithm to find an upper bound
 |  |  |  |  |
| **Linear Programming** |  |  |  |  |
| * Able to find the constraints of a linear programming problem given a description
 |  |  |  |  |
| * Able to draw a linear programming problem graphically
 |  |  |  |  |
| * Able to indicate the feasible region graphically.
 |  |  |  |  |
| * Able to use the objective line method to find a maximum or minimum point
 |  |  |  |  |
| * Able to use the vertex testing method to locate the optimal point
 |  |  |  |  |
| * Able to determine if a linear programming problem needs integer solutions
 |  |  |  |  |
| * Able to find the integer solutions for a linear programming problem
 |  |  |  |  |
| **Matchings** |  |  |  |  |
| * Able to define a matching and bipartite graph
 |  |  |  |  |
| * Able to model a matching from a table or paragraph into a bipartite graph
 |  |  |  |  |
| * Able to find an alternating path from a bipartite graph
 |  |  |  |  |
| * Able to improve a matching from a bipartite graph
 |  |  |  |  |
|  | **Red** | **Amber** | **Green** | **To address this before the exam I will:-** |
| * Able to prove that a matching is maximal
 |  |  |  |  |

|  |
| --- |
| **REVISION****Use the information on this checklist to make revision cards and notes** |

**Grade tracking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Grade* | *Date* | *Grade* | *Date* | *Grade* | *Date* |
|  |  |  |  |  |  |
| *Grade* | *Date* | *Grade* | *Date* | *Grade* | *Date* |
|  |  |  |  |  |  |

*Note: You should discuss this checklist regularly with your subject teacher/mentor*