Question No. 1

9(b)	**One** mark for each correct calculation as follows **(Max 4)** • Node B (from Home) (Line 3 in table) • Node C (from Home) (Line 4 in table) • Node B and Node E (from A) (Lines 5 and 6 in table) • Node F and Node School (from E) (Lines 7 and 8 in table) • Node School (from F) (Line 9 in table) **One** mark for correct path **(Max 1)**: • Home ⇨ A ⇨ E ⇨ F ⇨ School	**5**

	Node	Cost from Home Node (g)	Heuristic (h)	Total (f = g + h)
1	*Home*	*0*	*14*	*14*
2	*A*	*1*	*10*	*11*
3	B	5	7	12
4	C	4	9	13
5	B	1 + 3 = 4	7	11
6	E	1 + 6 = 7	3	10
7	F	7 + 1 = 8	3	11
8	School	7 + 5 = 12	0	12
9	School	8 + 3 = 11	0	11

Final Path	Home ⇨ A ⇨ E ⇨ F ⇨ School

Question No. 2

5(a)	Working **(Max 3)** May be seen on diagram • Initialisation: setting Base to 0 • … and the rest of the towns to ∞ • Evidence to show values at nodes being updated • Evidence to show 'visited node(s)' May be seen in working section of paper • Evidence to show calculation of at least one route • Evidence to show more than one route has been calculated for at least one town **Correct Answer (Max 2)** **One mark** for four correct values… … **One mark** for all values correct	**5**

Town 1	Town 2	Town 3	Town 4	Town 5	Town 6
3	5	2	9	3	8

Question No. 3

8(c)	MP1	Linear search O(n) **and** Binary search $O(\log_2 n)$ / O(Log n)	**3**
	MP2	time to search increases linearly in relation to the number of items in the list for a linear search and logarithmically for a Binary search	
	MP3	time to search increases less rapidly for a binary search and time to search increases more rapidly for a linear search	

Question No. 4

(a)
```
FUNCTION Hash(Key : STRING) RETURNS INTEGER
DECLARE Number : INTEGER
Number ← ASCII(LEFTSTRING(Key,1))
// Number ← ASCII(Key[1])
Number ← Number - 64
RETURN Number
// Result ← Number  // Hash ← Number
ENDFUNCTION
```

Accept `ASC` instead of `ASCII`
Accept `LEFT` instead of `LEFTSTRING`
Key can be a different identifier but must be the same in both places [5]

(b) (i)

Dictionary

Index	Key	Value
1		
2		
3	Computer	Rechner
4	Disk	Platte
5	Error	Fehler
6	File	Datei
7		
8		
:	:	
:	:	
1999		
2000		

Ignore spelling mistakes
1 mark for 2 correct pairs entered in correct slots [2]

(ii) Collision / synonym / space already occupied / same index in array
Overwrites previous key-value pair

reject error [Max 2]

(iii) Create an overflow area
The 'home' record has a pointer to others with the same key // linked list
OR
Store the overflow record at the next available address …
in sequence (= next available)
OR
Re-design the hash function …. // write a different/another algorithm
to generate a wider range of indexes // enlarging storage space // to create fewer collisions [2]

(iv) Mark as follows:

Check whether slot is empty:

```
IF Dictionary[Index,1] <>"" //  != ''    //  > NULL   // >
NONE
```

If not: update index: `THEN Index ← <some value>`

...to find an empty slot (loop / follow pointer / go to overflow area) reject FOR loop

Insert code between lines 20 and 30

```
21 WHILE Dictionary[Index,1] > ""
22   Index ← Index + 1
23   IF Index > 2000
24     THEN
25       Index ← 1
26   ENDIF
27 ENDWHILE
```

[4]