Question 1.

```
Algorithm: bonus(age,expe)
Requires: two positive integers
Returns: a positive integer, bonus amount
```

```
1. if age<=30 || expe<=5 then
2.     return 5000
3. elseif (age>30 && expe<=10) || (age<=40 && expe<=20) then
4.     return 7000
5. elseif (age>40 && expe<=20) || (age>=31 && age<=50 && expe<=30) then
6.     return 10000
7. else
8.     Return 15000
9. endif
```

TRACE:

```
bonus(45,15)
line1: FALSE, GOTO line3
line3: FALSE, GOTO line5
line5: TRUE ➔ return 10000.
```

Question 2.

a)

```
Algorithm: length(L)
Requires: a list
Returns: a positive integer>=0, list length
```

```
1. if isEmpty(L) then
2.    return 0
3. else
4.    return 1+length(tail(L))
5. Endif
```

b)

```
Algorithm: powerten(n)
Requires: a positive integer n>=0
Returns: a positive integer, 10^n
```

```
1. if n==0 then
2.    return 1
3. else
4.    return 10*powerten(n-1)
5. Endif
```

c)

```
Algorithm: list2num(L) [main]
Requires: a list of positive integers
Returns: a positive n-digit integer
```

```
1. Let n=length(L)
2.    return listHelper(L,n)
```

```
Algorithm: listHelper(L,n) [helper]
Requires: a nonempty list and a positive integer n>=1
Returns: a positive n-digit integer
```

```
1. if n==1 then
2.    return value(L)
3. else
4.    return value(L)*powerten(n-1)+listHelper(tail(L),n-1)
5. Endif
```

Question 3.

a)

(i) Any sorted list with 5 elements, e.g. [1,2,3,4,5]

(ii) $n \log_2 n = 100 \times \lceil \log_2 100 \rceil = 700$

b)

(i) Because the list is sorted and binary search algorithm only works on sorted lists.

(ii) Find middle of the list: `18 > 11;` so keep the left half of the list: `L=[7,11,13,14,16]`
Find middle of the list: `13 > 11;` keep the left half of the list: `L=[7,11]`
Find middle of the list: `7 < 11;` keep the right half of the list: `L=[11]`
Find middle of the list: `11 = 11;` return TRUE.

c)

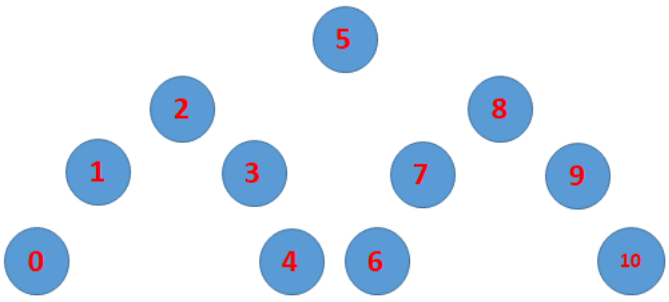
```
T=[30,7,11,9,20,5,10,40]
[7,11,9,5,20,10][30][40]
[5][7][11,9,20,10][30][40]
[5][7][9,10][11][20][30][40]
[5][7][9][10][11][20][30][40]
[5,7,9,10,11,20,30,40]
```

Question 4.

a)

(i) $\lfloor \log_2 n \rfloor + 1$

(ii) Firstly sort the list: `Q=[0,1,2,3,4,5,6,7,8,9,10].` There are 11 elements, find the middle to make the root.



(iii) Only 1 node, if the node value is less than 0 or greater than 10. Otherwise 4 nodes to fill the last row/generation.

b) A=31 or 32; B=49; C= no integer value; D=32 if A=31, no integer value if A=32.

c)

(i)

```
Algorithm: inOrder(bsT)
Requires: a binary search tree (BST)
Returns: a sorted list
```

```
1. if isLeaf(bsT) then
2.     return Nil
3. else
4.     return merge(inOrder(left(bsT)),merge(cons(root(BST),nil),inOrder(right(bsT))))
5. endif
```

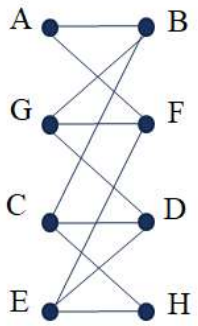
(ii) `merge(merge([],merge([30],[35])),merge([40],merge([45],merge([50],[]))))`

Question 5.

a)

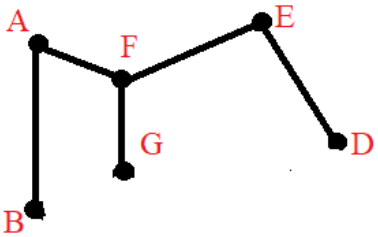
(i) It is not complete since not all vertices are connected. For example F is not connected to B.

(ii) The graph is bipartite according to the following diagram.



(iii) Neither ET nor EP. More than two vertices with odd degree.

b) 6 vertices so we need 5 edges. There are 7 edges, hence delete 2 without having a cycle.



Number of spanning trees=$\binom{7}{5} - \binom{3}{2} - \binom{3}{2} = 21 - 3 - 3 = 15.$

c) There are 8 vertices and we need 7 edges without cycles. Start at vertex A:

VERTICES	EDGE SELECTED	# OF EDGES
A	AC=5	1
A, C	CB=3	2
A, C, B	CE=5	3
A, C, B, E	ED=4	4
A, C, B, E, D	DG=5	5
A, C, B, E, D, G	GF=4	6
A, C, B, E, D, G, F	GH=5	7
SHORTEST PATH LENGTH	***31***	STOP!