Q1.

```
1. void multiply(int a[10][10], int b[10][10], int r1, int c1, int r2, int c2)
2. {
3.    int mult[10][10], i, j, k;
4.
5.    // Initializing elements of matrix mult to 0.
6.    for(i = 0; i < r1; ++i)
7.       for(j = 0; j < c2; ++j)
8.       {
9.          mult[i][j]=0;
10.       }
11.   // Multiplying matrix a and b and storing in array mult.
12.   for(i = 0; i < r1; ++i)
13.      for(j = 0; j < c2; ++j)
14.         for(k = 0; k < c1; ++k)
15.         {
16.            mult[i][j] += a[i][k] * b[k][j];
17.         }
18.   // Displaying the multiplication of two matrix.
19.   cout << endl << "Output Matrix: " << endl;
20.   for(i = 0; i < r1; ++i)
21.   for(j = 0; j < c2; ++j)
22.   {
23.      cout << " " << mult[i][j];
24.      if(j == c2-1)
25.         cout << endl;
26.   }
27.   cout << endl;
28. }
```

The C++ function given above [Adapted from: https://www.programiz.com/cpp-programming/examples/matrix-multiplication] is used for multiplying two matrices (a and b). Note that line numbers have been added at the start of each line of code for convenience.

a. Draw the control flow graph of this function.

b. What is the maximum number of linearly independent paths in the flow graph of this function?

Maximum number of linearly independent paths = V(G) = E – N + 2 =

c. Enumerate the paths.

Q2.

An economics application estimates the human poverty index (HPI) of a country by considering its GDP in billions of US dollars (0.0 – 100.0, 100.0+), its unemployment rate (UR) as a percentage (0.0 – 10.0, 10.1 – 50.0, 50.1 – 100.0), its inflation rate (IR) (low, high), and its average family size (AFS) (very small, small, medium, large, very large). The HPI estimation module of this application uses the estimates shown in the table below.

GDP		*0.0 – 100.0*						*100.0+*					
UR		*0.0 – 10.0*		*10.1 – 50.0*		*50.1 – 100.0*		*0.0 – 10.0*		*10.1 – 50.0*		*50.1 – 100.0*	
IR		*low*	*high*	*low*	*high*	*low*	*high*	*low*	*high*	*low*	*high*	*low*	*high*
AFS	*very small*	14.5	13.5	15.5	15.0	14.0	16.0	10.0	11.0	11.5	12.5	12	13
	small	15.5	14.5	16.5	16.0	15.0	17.0	11.0	12.0	12.5	13.5	13	14
	medium	16.5	15.5	17.5	17.0	16.0	18.0	12.0	13.0	13.5	14.5	14	15
	large	17.5	16.5	18.5	18.0	17.0	19.0	13.0	14.0	14.5	15.5	15	16
	very large	18.5	17.5	19.5	19.0	18.0	20.0	14.0	15.0	15.5	16.5	16	17

Use ECP and BVA to fill out the following two tables for black-box testing of the HPI estimation module. Use **minimum** test cases in the last table.

Variable	**Valid ECs**	**Representing values**		**Invalid ECs**	**Representing values for invalid ECs**
		For valid ECs	**Boundary values**		
GDP					
UR					
IR					

AFS					

Test case type	Test case no.	GDP	UR	IR	AFS	Test case results (HPI)
For valid ECs						
For invalid ECs						