Christopher Murray
CS2263
Assignment 1
January 19th, 2017

1.a Source code

```
/*Author: Christopher Murray
 *Date: Jan 20th, 2017
 *This program firstly takes the number of vertices and edges, represents them
 *in a Compressed Sparse Row format. The program then displays the number
 *vertices in each level of in and out degree
*/
#include <stdio.h>
#include <stdlib.h>

//Swap function will swap two elements of an array based on the indices provided
void swap(int Array[], int i1, int i2);

//Add adge will add an edge in CSR format
void addEdge(int vArray[], int eArray[], int v1, int v2, int vArrayLen);

//inOutDegree populates inArrray and outArray with the in and out degree of each
//vertex (respectively)
void inOutDegree(int inArray[], int outArray[], int vArray[], int eArray[],
int vArrayLen);

//maxArray returns the maximum value of an array
int maxArray(int Array[], int ArrayLen);

//max determines the max of 2 integers
int max(int i, int j);

int main(){
  int v;
  printf("Please enter the number of vertices:\n");
  int vNum = scanf("%d", &v);

  int e;
  printf("Please enter the number of edges:\n");
  int eNum = scanf("%d", &e);

  int vArray[v+1]; // extra cell is to store the length of the edge array
  int inArray[v]; // This array holds the in-degree of each vertice
  int outArray[v]; // This array holds the out degree of each vertice
  for(int i = 0; i < v; i++){
    vArray[i] = 0;
    inArray[i] = 0;
    outArray[i] = 0;
  }
```

```
vArray[v] = e; //Stores the size of the edge array
int eArray[e];
for(int i = 0; i < e; i++){
  eArray[i] = 0;
}
for(int i = 0; i < e; i++){
  int v1, v2;
  printf("Please enter an edge (set of two vertices):\n");
  int in = scanf("%d %d", &v1, &v2);
  printf("%d\n", in);
  if(in != 2){
    printf("Insufficient number of edges. Terminating\n");
    return EXIT_FAILURE;
  }
  if(v1 >= v || v2 >= v){
    printf("Invalid input. One or more vertice is out of bounds\n");
    return EXIT_FAILURE;
  }
  addEdge(vArray, eArray, v1, v2, v+1);
}

printf("%d %d\n", v, e);

for (int i = 0; i <=v; i ++){
  printf("%d ", vArray[i]);
}
printf("\n");
for(int i = 0; i < e; i++){
  printf("%d ", eArray[i]);
}
printf("\n");

inOutDegree(inArray, outArray, vArray, eArray, v+1);

int dArraySize = max(maxArray(inArray, v), maxArray(outArray, v)) + 1;

int inDArray[dArraySize];
for(int i = 0; i < dArraySize; i ++){
  inDArray[i] = 0;
}
for(int i = 0; i < v; i++){
  inDArray[inArray[i]] ++;
}

int outDArray[dArraySize];
for(int i = 0; i < dArraySize; i ++){
  outDArray[i] = 0;
}
for(int i = 0; i < v; i ++){
```

```
    outDArray[outArray[i]] ++;
  }

  for(int i = 0; i < dArraySize; i ++){
    printf("%d, in: %d, out: %d\n", i, inDArray[i], outDArray[i]);
  }
  return 0;
}

void addEdge(int vArray[], int eArray[], int v1, int v2, int vArrayLen){
  int vALen = vArrayLen;
  int offset = vArray[v1 + 1];
  for(int i = vArray[vALen -1] -1; i > offset; i--){
    swap(eArray, i, i-1);
  }
  eArray[offset] = v2;
  for(int i = vALen -2; i > v1; i--){
    vArray[i] ++;
  }
}

void inOutDegree(int inArray[], int outArray[], int vArray[], int eArray[],
int vArrayLen){
  for(int i = 0; i < vArrayLen-1; i ++){
    int diff = vArray[i + 1] - vArray[i];
    outArray[i] = diff;
  }

  int eArrayLen = vArray[vArrayLen -1];
  for(int i = 0; i < eArrayLen; i ++){
    inArray[eArray[i]] ++;
  }
}

int maxArray(int Array[], int arrayLen){
  int max = Array[0];
  if(arrayLen > 0){
    for(int i = 1; i < arrayLen; i++){
      if(Array[i] > max){
        max = Array[i];
      }
    }
  }
  return max;
}

int max( int i, int j){
  if(i < j){
    return i;
```

```
  }
  else{
    return j;
  }
}

void swap(int Array[], int i1, int i2){
  int temp = Array[i1];
  Array[i1] = Array[i2];
  Array[i2] = temp;
}
```

**I also used input.txt to speed up development

8 12
0 1
0 2
1 3
2 1
2 4
3 2
3 5
4 5
4 6
5 6
5 7
6 7

b) Terminal sessions

Case 1 a) input.txt

```
[p79hb@id415m28 As1]$ ./As1 < input.txt
Please enter the number of vertices:
Please enter the number of edges:
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
Please enter an edge (set of two vertices):
8 12
0 2 3 5 7 9 11 12 12
1 2 3 1 4 2 5 5 6 6 7 7
0, in: 1, out: 1
1, in: 2, out: 2
2, in: 5, out: 5
[p79hb@id415m28 As1]$
```

Case2: One or more vertices in an edge is out of bounds

```
[p79hb@id415m28 As1]$ ./As1
Please enter the number of vertices:
6
Please enter the number of edges:
10
Please enter an edge (set of two vertices):
1 7
Invalid input. One or more vertice is out of bounds
[p79hb@id415m28 As1]$
```

Case3: Insufficient number of edges

```
[p79hb@id415m28 As1]$ ./As1
Please enter the number of vertices:
10
Please enter the number of edges:
10
Please enter an edge (set of two vertices):
1 7
Please enter an edge (set of two vertices):
3 7
Please enter an edge (set of two vertices):
Insufficient number of edges. Terminating
[p79hb@id415m28 As1]$
```