```c
#include<stdio.h>
#include<stdlib.h>

#include "Graph.h"

int main(int argc, char* argv[]){
   ListRef L;
   GraphRef P;
   GraphRef G = newGraph(3);
   int i;
   addEdge(G,1,3);
   addEdge(G,1,2);
   addEdge(G,2,3);
   addEdge(G,1,3);
   printGraph(stdout,G);
   printGraphInfo(G);
   makeNull(G);
   printGraph(stdout,G);
   printGraphInfo(G);
   freeGraph(&G);

   P = newGraph(6);
   addEdge(P,1,2);
   addEdge(P,1,3);
   addEdge(P,2,4);
   addEdge(P,2,5);
   addEdge(P,2,6);
   addEdge(P,3,4);
   addEdge(P,4,5);
   addEdge(P,5,6);
   BFS(P,1);
   printGraph(stdout,P);
   printGraphInfo(P);
   for( i = 1; i <= getOrder(P); i++ ){
      printf("d[%d]=%d\nP[%d]=%d\n",i,P->d[i],i,P->P[i]);
   }
   L=newList();
   getPath(L,P,4);
   printList(stdout,L);
   printf("\n\n");
   BFS(P,3);
   printGraph(stdout,P);
   printGraphInfo(P);
   for( i = 1; i <= getOrder(P); i++ ){
      printf("d[%d]=%d\nP[%d]=%d\n",i,P->d[i],i,P->P[i]);
```

```
    }
    makeEmpty(L);
    getPath(L,P,6);
    printList(stdout,L);
    freeList(&L);
    freeGraph(&P);
    return 0;
}
```