Team Zombie Coders

Members: Aurora Havens and Emely Villeda-Principe

Set of initial commands that are used in stress test
mode to test the functionality of the program:

1. mmode
2. add
3. new
4. default
5. avl
6. return
7. imode
8. load
9. avl
10.   search
11.   main
12.   return
13.   mmode
14.   clear
15.   add
16.   new
17.   default
18.   hash
19.   return
20.   imode

QueryHandler

```
vector<string> andWords;
vector<string> orWords;
vector<string> notWords;
vector<string> allWords;
string notInclude;

vector<int> includeDocs;
vector<int> excludeDocs;

Query* que;
```

```
QueryHandler();
~QueryHandler();

void getWords(string line);
void addToAnd(string s);
void addToOr(string s);
void addToNot(string s);
void emptyVectors();

Query *&getQue();
```

Page

```
int pageNum;
int frequency;
```

```
Page();
Page(int pgNum);
Page(int pgNum, int freq);
Page(const Page& other);
Page& operator=(const Page &rhs);

void setPageNum(int pgNum);
int getPageNum() const;

void increaseFrequency();
int getFrequency();
void totalFrequencies(int addedFreqs);
```

DocParser

```
vector<string> files;
vector<Document> docObjects;
vector<Document> docObjectsCopy;

string dir;
string line;
string oneDot;
string twoDot;
string check;
string page;
string title;
string id;
string revision;
string text;

int idNum;
```

```
DocParser();
int getdir(string dir, vector<string> &files);
void parseDocuments(string dirName);
void cleanDocObjects();
vector<Document> &getDocObjects();
void printResults(vector<Page> &pages);
void printPage(int pageNumber);
void emptyVectors();
```

Query

```
string original;
string kweree;
string notInclude;
vector <Page> docList;
vector<string> andList;
vector<string> orList;
vector<string> notList;
```

```
Query();
Query(string kwr);
Query(vector<string> a, vector<string> o, vector<string> n, string nI);

void setOriginal(string origin);
string getOriginal();

string getKweree();
string getNotInclude();
vector<Page> &getDocList();
void addToList(vector<Page> &include);
void removeFromList(vector<Page> &exclude);

vector<string> &getAndList();
vector<string> &getOrList();
vector<string> &getNotList();
```

Document

```
vector<string> docText;
string textToClean;
int pageNum;
string plainText;
string titleOfDoc;
vector<string> stopWords;
```

```
Document(string text, int pg);
Document(string text, int pg, string title);

void setPageNum(int pg);
int getPageNum() const;
string getTitle();
string returnText();
void addToStrings(string s);
void cleanStrings();
string bleachString(string s);
vector<string> getDocText();
int sizeOfDocText();
```