**Querying the Document**

char\* kth\_word\_in\_mth\_sentence\_of\_nth\_paragraph(char\*\*\*\* document, int k, int m, int n) {

return document[n-1][m-1][k-1];

}

char\*\* kth\_sentence\_in\_mth\_paragraph(char\*\*\*\* document, int k, int m) {

return document[m-1][k-1];

}

char\*\*\* kth\_paragraph(char\*\*\*\* document, int k) {

return document[k-1];

}

char\*\* split\_string(char\* text, char delim) {

assert(text != NULL);

char\*\* result = malloc(1\*sizeof(char\*));

int size = 1;

char\* temp = strtok(text, &delim);

\*result = temp;

while(temp != NULL) {

size++;

result = realloc(result,size\*sizeof(char\*));

temp = strtok(NULL, &delim);

result[size-1] = temp;

}

return result;

}

char\*\*\*\* get\_document(char\* text) {

assert(text != NULL);

// split text by '\n' and count number of paragraphs

char\*\* paragraphs = split\_string(text, '\n');

int npar = 0;

while (paragraphs[npar] != NULL) {

npar++;

}

char\*\*\*\* doc = malloc((npar+1)\*sizeof(char\*\*\*));

// set last position to NULL for the user

// to know when the array ends.

doc[npar] = NULL;

int i = 0;

while (paragraphs[i] != NULL) {

// split sentences of paragraph by '.' and count number of sentences

char\*\* sentences = split\_string(paragraphs[i], '.');

int nsen = 0;

while(sentences[nsen] != NULL) {

nsen++;

}

doc[i] = malloc((nsen+1)\*sizeof(char\*\*));

// set last position to NULL for the user

// to know when the array ends.

doc[i][nsen] = NULL;

int j = 0;

while (sentences[j] != NULL) {

// remember that doc[0][0] means: paragraph #0,

// sentence #0 and should act like a pointer to

// the first element of an array of words (strings)

// split string by ' ' and associate doc[i][j]

// with the array of strings representing words

// that is returned by split\_string.

doc[i][j] = split\_string(sentences[j], ' ');

j++;

}

i++;

}

return doc;

}