#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char \*\*words, \*pWord, buf[BUFSIZ];

size\_t size, i, j;

printf("Input string: ");

if ( !fgets(buf, BUFSIZ, stdin) )

{

fprintf(stderr, "Can't get string!\n");

return 1;

}

printf("\n\nYour string: %s", buf);

if ( pWord = strrchr(buf, '\n') )

\*pWord = '\0';

words = NULL;

size = 0;

for ( pWord = strtok(buf, " \t"); pWord != NULL; pWord = strtok(NULL, " \t") )

{

if ( ( words = (char\*\*)realloc(words, sizeof(char\*) \* (size + 1)) ) == NULL )

{

fprintf(stderr, "Memory error!\n");

return 1;

}

if ( ( words[size] = \_strdup(pWord) ) == NULL )

{

fprintf(stderr, "Memory error!\n");

return 1;

}

++size;

}

if ( !words )

{

fprintf(stderr, "Can't find some words!\n");

return 1;

}

printf("\n");

for ( i = 0; i < size - 1; ++i )

{

if ( !\*words[i] )

continue;

for ( j = i + 1; j < size; ++j )

{

if ( !strcmp(words[i], words[j]) )

{

printf("Address of %s is %p\n", words[j],&words[j]);

\*(words[j]) = '\0';

}

}

}

printf("\nNew string: ");

for ( i = 0; i < size; ++i )

if ( \*(words[i]) )

printf("%s ", words[i]);

printf("\n");

do

{

free(words[--size]);

}

while ( size );

free(words);

return 0;

}