```
/*
Class: CPSC 346-01 & CPSC 346-02
Team Member 1: Nicholas Walker
Team Member 2: Brett Barinaga
GU Username of project lead: nwalker
Pgm Name: proj2.c
Pgm Desc: Counts words in input string
Usage: ./a.out
*/


#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAX_LINE 80

char* get_inp();

void disp_inp(char*);

int wc(char*);

int main()
{
 char* inp = get_inp();
 printf("here is your input : %s\n",inp);
 disp_inp(inp);
 printf("%i\n",wc(inp));
 free(inp);  //return dynamially allocated memory to the heap
 return 0;
}


char* get_inp()
{
 char* inp = (char*) malloc(MAX_LINE);
 char* start = inp;
 char c;

 printf("Enter text <= 80 characters in length\n");
 while ((c = getchar()) != '\n')
  *inp++ = c; //weird, yes? First add the character to the dereffed pointer
    //then go to the next position in dynamic memory
```

```
 *inp = '\0';
 return start;
}

void disp_inp(char* out)
{
 while(*out)  //continue until the null character is encountered
  putchar(*out++);
 putchar('\n');
}

/*
pre:  inp is the address of an 80 byte block of memory
post: returns the number of words stored in the block of memory.  A
word
      is any sequence of characters that
      1) are stored at the beginning of the block or
      2) are stored at end of the block or
      3) are terminated by the end-of-line character or
      4) are preceded by and ended by one or more spaces

*/
int wc(char* inp)
{
   int words = 0;
   int location = 0;
   for(int i = 0; inp[i] != '\0'; i++)
   {
      if (inp[i] == ' ')
      {
         location = 0;
      }
      else if((location == 0) && (inp[i] != ' '))
      {
         location = 1;
         words++;
      }
   }
   return words;
}
```