Almog String Manipulation

Generated by Doxygen 1.9.1

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

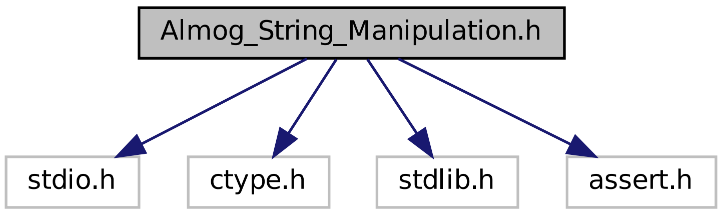
File Documentation

2.1 Almog_String_Manipulation.h File Reference

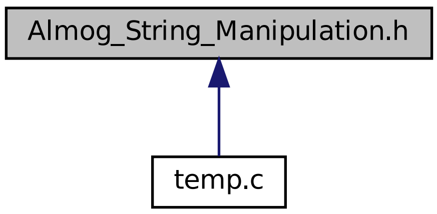
Lightweight string and line manipulation helpers.

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <assert.h>
```

Include dependency graph for Almog_String_Manipulation.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define ASM_MAXDIR 100

  Generic maximum directory length constant (not used by the functions in this header but available to callers).
- #define ASM_MAX_LEN_LINE (int)1e3

  Maximum number of characters read by asm_get_line (excluding the terminating null).
- #define asm_dprintSTRING(expr) printf(#expr " = %s\n", expr)

  Debug print a C string expression as "expr = value\n".
- #define asm_dprintCHAR(expr) printf(#expr " = %c\n", expr)

  Debug print a character expression as "expr = c\n".
- #define asm_dprintINT(expr) printf(#expr " = %d\n", expr)

  Debug print an integer expression as "expr = n\n".
- #define asm_dprintSIZE_T(expr) printf(#expr " = %zu\n", expr)

  Debug print a size_t expression as "expr = n\n".

Functions

- int asm_get_line (FILE ∗fp, char ∗dst)

  Read a single line from a stream into a buffer.
- int asm_length (char ∗str)

  Compute the length of a null-terminated C string.
- int asm_get_next_word_from_line (char ∗dst, char ∗src, char seperator)

  Extract the next word from a line without modifying the source.
- void asm_copy_array_by_indesies (char ∗target, int start, int end, char ∗src)

  Copy a substring [start, end) from src into target and null-terminate.
- int asm_get_word_and_cut (char ∗dst, char ∗src, char seperator)

  Get the next word and cut the source string at that point.
- int asm_str_in_str (char ∗src, char ∗word2search)

  Count occurrences of a substring within a string.
- int asm_strncmp (const char ∗s1, const char ∗s2, const int N)

  Compare up to N characters for equality (boolean result).

2.1.1 Detailed Description

Lightweight string and line manipulation helpers.

This single-header module provides small utilities for working with C strings:

- Reading a single line from a FILE stream
- Measuring string length
- Extracting the next "word" (token) from a line using a separator
- Cutting the extracted word from the source buffer
- Copying a substring by indices
- Counting occurrences of a substring
- A boolean-style strncmp (returns 1 on equality, 0 otherwise)

Usage

- In exactly one translation unit, define ALMOG_STRING_MANIPULATION_IMPLEMENTATION before including this header to compile the implementation.
- In all other files, include the header without the macro to get declarations only.

Notes and limitations

- All destination buffers must be large enough; functions do not grow or allocate buffers.
- asm_get_line enforces MAX_LEN_LINE characters (not counting the terminating '\0'). Longer lines cause a fatal error via exit(1).
- asm_strncmp differs from the standard C strncmp: this version returns 1 if equal and 0 otherwise.
- These functions are not locale-aware unless otherwise noted (isspace is used for whitespace handling).

Definition in file Almog_String_Manipulation.h.

2.1.2 Macro Definition Documentation

2.1.2.1 asm_dprintCHAR

```
#define asm_dprintCHAR(
              expr ) printf(#expr " = %c\n", expr)
```

Debug print a character expression as "expr = c\n".

Parameters

expr	An expression that yields a character promoted to int.

Definition at line 72 of file Almog_String_Manipulation.h.

2.1.2.2 asm_dprintINT

```
#define asm_dprintINT(
              expr ) printf(#expr " = %d\n", expr)
```

Debug print an integer expression as "expr = n\n".

Parameters

expr	An expression that yields an int.

Definition at line 79 of file Almog_String_Manipulation.h.

2.1.2.3 asm_dprintSIZE_T

```
#define asm_dprintSIZE_T(
              expr ) printf(#expr " = %zu\n", expr)
```

Debug print a size_t expression as "expr = n\n".

Parameters

expr	An expression that yields a size_t.

Definition at line 86 of file Almog_String_Manipulation.h.

2.1.2.4 asm_dprintSTRING

```
#define asm_dprintSTRING(
              expr ) printf(#expr " = %s\n", expr)
```

Debug print a C string expression as "expr = value\n".

Parameters

expr	An expression that yields a pointer to char (const or non-const).

Definition at line 65 of file Almog_String_Manipulation.h.

2.1.2.5 ASM_MAX_LEN_LINE

```
#define ASM_MAX_LEN_LINE (int)1e3
```

Maximum number of characters read by asm_get_line (excluding the terminating null).

If an input line exceeds this value before encountering '
' or EOF, asm_get_line prints an error to stderr and terminates the process with exit(1).

Definition at line 58 of file Almog_String_Manipulation.h.

2.1.2.6 ASM_MAXDIR

```
#define ASM_MAXDIR 100
```

Generic maximum directory length constant (not used by the functions in this header but available to callers).

Definition at line 47 of file Almog_String_Manipulation.h.

2.1.3 Function Documentation

2.1.3.1 asm_copy_array_by_indesies()

```
void asm_copy_array_by_indesies (
            char * target,
            int start,
            int end,
            char * src )
```

Copy a substring [start, end) from src into target and null-terminate.

Copies characters with indices i = start, start+1, ..., end-1 from src into target, then writes a terminating '\0'.

Parameters

target	Destination buffer. Must be large enough to hold (end - start) characters plus the null terminator.
start	Inclusive start index within src (0-based).
end	Exclusive end index within src (must satisfy end >= start).
src	Source string buffer.

Warning

No bounds checking is performed. The caller must ensure valid indices and sufficient target capacity.

Note

This routine supports in-place "left-shift" usage where target == src and start > 0 (used by asm_get_word_↩
and_cut).

Definition at line 232 of file Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut().

2.1.3.2 asm_get_line()

```
int asm_get_line (
            FILE * fp,
            char * dst )
```

Read a single line from a stream into a buffer.

Reads characters from the FILE stream until a newline ('
') or EOF is encountered. The newline, if present, is not copied. The result is always null-terminated.

Parameters

fp	Input stream (must be non-NULL).
dst	Destination buffer. Must have capacity of at least MAX_LEN_LINE + 1 bytes.

Returns

Number of characters stored in dst (excluding the terminating null).

Return values

-1	EOF was encountered before any character was read.

Note

If the line exceeds MAX_LEN_LINE characters before a newline or EOF, the function prints an error and calls exit(1).

An empty line returns 0 (not -1).

Definition at line 119 of file Almog_String_Manipulation.h.

References ASM_MAX_LEN_LINE.

2.1.3.3 asm_get_next_word_from_line()

```
int asm_get_next_word_from_line (
            char * dst,
            char * src,
            char seperator )
```

Extract the next word from a line without modifying the source.

Skips leading whitespace in src (as determined by isspace), then copies characters into dst until one of the following is seen: the separator, a newline ('
'), or the string terminator ('\0'). The copied word in dst is null-terminated and is never empty on success.

Special case:

- If the very first character in src (at index 0, without leading whitespace) is the separator, '
  ', or '\0', that single character is returned as a one-character "word".

Parameters

dst	Destination buffer for the extracted word. Must be large enough to hold the token plus the null terminator.
src	Source C string to parse (not modified by this function).
seperator	Separator character to stop at (spelling as in the API).

Returns

The number of characters consumed from src (i.e., the index of the first unconsumed character).

Return values

-1	No word was found (e.g., only whitespace before a delimiter or end-of-string).

Note

The source buffer is not altered. To both extract and advance/cut the source, see asm_get_word_and_cut.

Definition at line 182 of file Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut().

2.1.3.4 asm_get_word_and_cut()

```
int asm_get_word_and_cut (
             char * dst,
             char * src,
             char seperator )
```

Get the next word and cut the source string at that point.

Extracts the next word from src (per asm_get_next_word_from_line semantics) into dst. On success, src is modified in-place to remove the consumed prefix. The new src begins at the stopping character (the separator, newline, or terminator).

Example: For src = "abc,def", separator = ','

- dst becomes "abc"
- src becomes ",def" (note the leading separator remains)

Parameters

dst	Destination buffer for the extracted word (large enough for the token and terminating null).
src	Source buffer. Modified in-place if a word is found.
seperator	Separator character to stop at (spelling as in the API).

Returns

1 if a word was extracted and src adjusted, 0 otherwise.

Definition at line 260 of file Almog_String_Manipulation.h.

References asm_copy_array_by_indesies(), asm_get_next_word_from_line(), and asm_length().

Referenced by main().

2.1.3.5 asm_length()

```
int asm_length (
            char * str )
```

Compute the length of a null-terminated C string.

Parameters

str	Null-terminated string (must be non-NULL).

Returns

The number of characters before the terminating null byte.

Definition at line 146 of file Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut(), and asm_str_in_str().

2.1.3.6 asm_str_in_str()

```
int asm_str_in_str (
            char * src,
            char * word2search )
```

Count occurrences of a substring within a string.

Counts how many times word2search appears in src. Occurrences may overlap.

Parameters

src	The string to search in (must be null-terminated).
word2search	The substring to find (must be null-terminated).

Returns

The number of (possibly overlapping) occurrences found.

Definition at line 285 of file Almog_String_Manipulation.h.

References asm_length(), and asm_strncmp().

2.1.3.7 asm_strncmp()

```
int asm_strncmp (
              const char * s1,
              const char * s2,
              const int N )
```

Compare up to N characters for equality (boolean result).

Returns 1 if the first N characters of s1 and s2 are all equal; otherwise returns 0. Unlike the standard C strncmp, which returns 0 on equality and a non-zero value on inequality/order, this function returns a boolean-like result (1 == equal, 0 == different).

Parameters

s1	First string (may be shorter than N).
s2	Second string (may be shorter than N).
N	Number of characters to compare.

Returns

1 if equal for the first N characters, 0 otherwise.

Definition at line 310 of file Almog_String_Manipulation.h.

Referenced by asm_str_in_str().

2.2 Almog_String_Manipulation.h

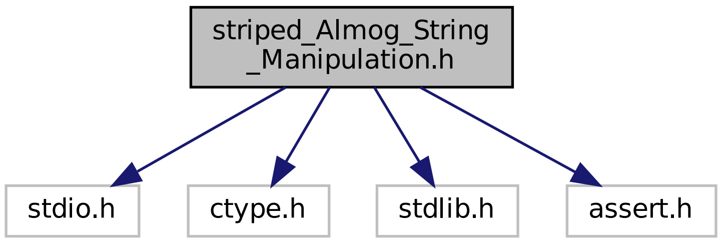
```
#ifndef ALMOG_STRING_MANIPULATION_H_
#define ALMOG_STRING_MANIPULATION_H_

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <assert.h>

#define ASM_MAXDIR 100

#define ASM_MAX_LEN_LINE (int)1e3

#define asm_dprintSTRING(expr) printf(#expr " = %s\n", expr)

#define asm_dprintCHAR(expr) printf(#expr " = %c\n", expr)

#define asm_dprintINT(expr) printf(#expr " = %d\n", expr)

```

```c
#define asm_dprintSIZE_T(expr) printf(#expr " = %zu\n", expr)

int asm_get_line(FILE *fp, char *dst);
int asm_length(char *str);
int asm_get_next_word_from_line(char *dst, char *src, char seperator);
void asm_copy_array_by_indesies(char *target, int start, int end, char *src);
int asm_get_word_and_cut(char *dst, char *src, char seperator);
int asm_str_in_str(char *src, char *word2search);
int asm_strncmp(const char *s1, const char *s2, const int N);

#endif /*ALMOG_STRING_MANIPULATION_H_*/

#ifdef ALMOG_STRING_MANIPULATION_IMPLEMENTATION
#undef ALMOG_STRING_MANIPULATION_IMPLEMENTATION


int asm_get_line(FILE *fp, char *dst)
{
    int i = 0;
    char c;

    while ((c = fgetc(fp)) != ’\n’ && c != EOF) {
        dst[i] = c;
        i++;
        if (i >= ASM_MAX_LEN_LINE) {
            fprintf(stderr, "ERROR: line too long\n");
            exit(1);
        }
    }
    dst[i] = ’\0’;
    if (c == EOF && i == 0) {
        return -1;
    }
    return i;
}

int asm_length(char *str)
{
    char c;
    int i = 0;

    while ((c = str[i]) != ’\0’) {
        i++;
    }
    return i++;
}

int asm_get_next_word_from_line(char *dst, char *src, char seperator)
{
    int i = 0, j = 0;
    char c;

    while (isspace((c = src[i]))) {
        i++;
    }

    while ((c = src[i]) != seperator &&
                      c != ’\n’&&
                      c != ’\0’) {
                        dst[j] = src[i];
                        i++;
                        j++;
    }

    if ((c == seperator ||
         c == ’\n’||
         c == ’\0’) && i == 0) {
            dst[j++] = c;
            i++;
    }

    dst[j] = ’\0’;

    if (j == 0) {
        return -1;
    }
    return i;

}

void asm_copy_array_by_indesies(char *target, int start, int end, char *src)
{
    int j = 0;
    for (int i = start; i < end; i++) {
        target[j] = src[i];
        j++;
    }
    target[j] = '\0';
}

int asm_get_word_and_cut(char *dst, char *src, char seperator)
{
    int last_pos;

    if (src[0] == '\0') {
        return 0;
    }
    last_pos = asm_get_next_word_from_line(dst, src, seperator);
    if (last_pos == -1) {
        return 0;
    }
    asm_copy_array_by_indesies(src, last_pos, asm_length(src), src);
    return 1;
}

int asm_str_in_str(char *src, char *word2search)
{
    int i = 0, num_of_accur = 0;
    while (src[i] != '\0') {
        if (asm_strncmp(src+i, word2search, asm_length(word2search))) {
            num_of_accur++;
        }
        i++;
    }
    return num_of_accur;
}

int asm_strncmp(const char *s1, const char *s2, const int N)
{
    int i = 0;
    while (i < N) {
        if (s1[i] == '\0' && s2[i] == '\0') {
            break;
        }
        if (s1[i] != s2[i] || (s1[i] == '\0') || (s2[i] == '\0')) {
            return 0;
        }
        i++;
    }
    return 1;
}


#endif /*ALMOG_STRING_MANIPULATION_IMPLEMENTATION*/

```

2.3 striped_Almog_String_Manipulation.h File Reference

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <assert.h>
```

Include dependency graph for striped_Almog_String_Manipulation.h:



Macros

- #define ASM_MAXDIR 100
- #define ASM_MAX_LEN_LINE (int)1e3
- #define asm_dprintSTRING(expr) printf(#expr " = %s\n", expr)
- #define asm_dprintCHAR(expr) printf(#expr " = %c\n", expr)
- #define asm_dprintINT(expr) printf(#expr " = %d\n", expr)
- #define asm_dprintSIZE_T(expr) printf(#expr " = %zu\n", expr)

Functions

- int asm_get_line (FILE ∗fp, char ∗dst)
- int asm_length (char ∗str)
- int asm_get_next_word_from_line (char ∗dst, char ∗src, char seperator)
- void asm_copy_array_by_indesies (char ∗target, int start, int end, char ∗src)
- int asm_get_word_and_cut (char ∗dst, char ∗src, char seperator)
- int asm_str_in_str (char ∗src, char ∗word2search)
- int asm_strncmp (const char ∗s1, const char ∗s2, const int N)

2.3.1 Macro Definition Documentation

2.3.1.1 asm_dprintCHAR

```
#define asm_dprintCHAR(
              expr ) printf(#expr " = %c\n", expr)
```

Definition at line 10 of file striped_Almog_String_Manipulation.h.

2.3.1.2 asm_dprintINT

```
#define asm_dprintINT(
              expr ) printf(#expr " = %d\n", expr)
```

Definition at line 11 of file striped_Almog_String_Manipulation.h.

2.3.1.3 asm_dprintSIZE_T

```
#define asm_dprintSIZE_T(
              expr ) printf(#expr " = %zu\n", expr)
```

Definition at line 12 of file striped_Almog_String_Manipulation.h.

2.3.1.4 asm_dprintSTRING

```
#define asm_dprintSTRING(
             expr ) printf(#expr " = %s\n", expr)
```

Definition at line 9 of file striped_Almog_String_Manipulation.h.

2.3.1.5 ASM_MAX_LEN_LINE

```
#define ASM_MAX_LEN_LINE (int)1e3
```

Definition at line 8 of file striped_Almog_String_Manipulation.h.

2.3.1.6 ASM_MAXDIR

```
#define ASM_MAXDIR 100
```

Definition at line 7 of file striped_Almog_String_Manipulation.h.

2.3.2 Function Documentation

2.3.2.1 asm_copy_array_by_indesies()

```
void asm_copy_array_by_indesies (
             char * target,
             int start,
             int end,
             char * src )
```

Definition at line 76 of file striped_Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut().

2.3.2.2 asm_get_line()

```
int asm_get_line (
             FILE * fp,
             char * dst )
```

Definition at line 23 of file striped_Almog_String_Manipulation.h.

References ASM_MAX_LEN_LINE.

2.3.2.3 asm_get_next_word_from_line()

```
int asm_get_next_word_from_line (
            char * dst,
            char * src,
            char seperator )
```

Definition at line 50 of file striped_Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut().

2.3.2.4 asm_get_word_and_cut()

```
int asm_get_word_and_cut (
            char * dst,
            char * src,
            char seperator )
```

Definition at line 85 of file striped_Almog_String_Manipulation.h.

References asm_copy_array_by_indesies(), asm_get_next_word_from_line(), and asm_length().

2.3.2.5 asm_length()

```
int asm_length (
            char * str )
```

Definition at line 41 of file striped_Almog_String_Manipulation.h.

Referenced by asm_get_word_and_cut(), and asm_str_in_str().

2.3.2.6 asm_str_in_str()

```
int asm_str_in_str (
            char * src,
            char * word2search )
```

Definition at line 98 of file striped_Almog_String_Manipulation.h.

References asm_length(), and asm_strncmp().

2.3.2.7 asm_strncmp()

```
int asm_strncmp (
              const char * s1,
              const char * s2,
              const int N )
```

Definition at line 109 of file striped_Almog_String_Manipulation.h.

Referenced by asm_str_in_str().

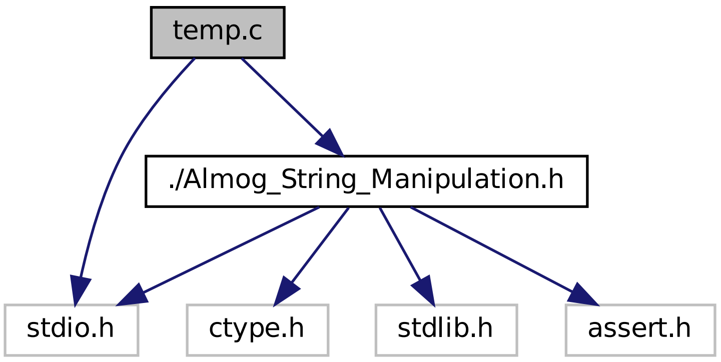
2.4 striped_Almog_String_Manipulation.h

```
#ifndef ALMOG_STRING_MANIPULATION_H_
#define ALMOG_STRING_MANIPULATION_H_
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <assert.h>
#define ASM_MAXDIR 100
#define ASM_MAX_LEN_LINE (int)1e3
#define asm_dprintSTRING(expr) printf(#expr " = %s\n", expr)
#define asm_dprintCHAR(expr) printf(#expr " = %c\n", expr)
#define asm_dprintINT(expr) printf(#expr " = %d\n", expr)
#define asm_dprintSIZE_T(expr) printf(#expr " = %zu\n", expr)
int asm_get_line(FILE *fp, char *dst);
int asm_length(char *str);
int asm_get_next_word_from_line(char *dst, char *src, char seperator);
void asm_copy_array_by_indesies(char *target, int start, int end, char *src);
int asm_get_word_and_cut(char *dst, char *src, char seperator);
int asm_str_in_str(char *src, char *word2search);
int asm_strncmp(const char *s1, const char *s2, const int N);
#endif
#ifdef ALMOG_STRING_MANIPULATION_IMPLEMENTATION
#undef ALMOG_STRING_MANIPULATION_IMPLEMENTATION
int asm_get_line(FILE *fp, char *dst)
{
    int i = 0;
    char c;
    while ((c = fgetc(fp)) != '\n' && c != EOF) {
        dst[i] = c;
        i++;
        if (i >= ASM_MAX_LEN_LINE) {
            fprintf(stderr, "ERROR: line too long\n");
            exit(1);
        }
    }
    dst[i] = '\0';
    if (c == EOF && i == 0) {
        return -1;
    }
    return i;
}
int asm_length(char *str)
{
    char c;
    int i = 0;
    while ((c = str[i]) != '\0') {
        i++;
    }
    return i++;
}
int asm_get_next_word_from_line(char *dst, char *src, char seperator)
{
    int i = 0, j = 0;
    char c;
    while (isspace((c = src[i]))) {
        i++;
    }
    while ((c = src[i]) != seperator &&
                  c != '\n'&&
                  c != '\0') {
                    dst[j] = src[i];
                    i++;
                    j++;
    }
    if ((c == seperator ||
         c == '\n'||
        c == '\0') && i == 0) {
            dst[j++] = c;
            i++;
    }
    dst[j] = '\0';
    if (j == 0) {
        return -1;
    }
    return i;
}
void asm_copy_array_by_indesies(char *target, int start, int end, char *src)
{
    int j = 0;
    for (int i = start; i < end; i++) {
        target[j] = src[i];
        j++;
    }
    target[j] = '\0';
}
int asm_get_word_and_cut(char *dst, char *src, char seperator)
{
    int last_pos;
    if (src[0] == '\0') {
        return 0;
    }
    last_pos = asm_get_next_word_from_line(dst, src, seperator);
    if (last_pos == -1) {
        return 0;
    }
    asm_copy_array_by_indesies(src, last_pos, asm_length(src), src);
    return 1;
}
int asm_str_in_str(char *src, char *word2search)
{
    int i = 0, num_of_accur = 0;
    while (src[i] != '\0') {
        if (asm_strncmp(src+i, word2search, asm_length(word2search))) {
            num_of_accur++;
        }
        i++;
    }
    return num_of_accur;
}
int asm_strncmp(const char *s1, const char *s2, const int N)
{
    int i = 0;
    while (i < N) {
        if (s1[i] == '\0' && s2[i] == '\0') {
            break;
        }
        if (s1[i] != s2[i] || (s1[i] == '\0') || (s2[i] == '\0')) {
            return 0;
        }
        i++;
    }
    return 1;
}
#endif
```

2.5 temp.c File Reference

```
#include <stdio.h>
#include "./Almog_String_Manipulation.h"
```

Include dependency graph for temp.c:



Macros

- #define ALMOG_STRING_MANIPULATION_IMPLEMENTATION

Functions

- int main (void)

2.5.1 Macro Definition Documentation

2.5.1.1 ALMOG_STRING_MANIPULATION_IMPLEMENTATION

```
#define ALMOG_STRING_MANIPULATION_IMPLEMENTATION
```

Definition at line 2 of file temp.c.

2.5.2 Function Documentation

2.5.2.1 main()

```
int main (
             void  )
```

Definition at line 5 of file temp.c.

References asm_get_word_and_cut().

2.6 temp.c

```c
#include <stdio.h>
#define ALMOG_STRING_MANIPULATION_IMPLEMENTATION
#include "./Almog_String_Manipulation.h"

int main(void)
{
    char str[] = "almog dobrescu";
    char word[256];

    asm_get_word_and_cut(word, str, ' ');

    printf("str:  %s\n", str);

    printf("word: %s\n", word);


    return 0;
}
```