

Midterm Exam 2013 Dr Ali Hamzeh

Time : 4 hours.

1) Develop the `double atof(char * string)` function which could convert any string array to corresponding double digit and return it. Consider the following table for some examples.

You are not allowed to use prepared C function. (10 pts)

Given Parameter	Return Digit
"123"	123
"12.3"	12.3
"12.3e1"	123
"123e-1"	12.3
"0123"	123
" +123"	123
" -123"	-123
"123ali"	123
"ali"	0
"ali123"	0

2) Develop the `void big_divide(char *number1 , char *number2 , char *result)` which able to process its to parameter which could be a natural number up to 100 digits in the form of strings and put their quotient into the third array called result. (15pts)

Number1	Number2	Result
"123"	"123"	"1"
"123"	"10"	"12"
"123"	"240"	"0"

3) Develop a function called strsplit with this prototype `void strsplit(char *source , char *separators , char result[][])`.

This function must split the given source using all character in the given separators and returns split parts. You can use strlen in this problem.(15pts)

Example	Result value
strsplit("hello world!" , " ")	{ "hello" , "world" }
strsplit("hello world!" , ",")	{ "hello world!" }
strsplit("hello world!" , "!")	{ "hello world" }
strsplit("hello world!" , "o")	{ "hell" , " w" , "rld!" }
strsplit("hello world!" , "l")	{ "he" , "o wor" , "d!" }
strsplit("hello world!" , "lo")	{ "he" , " w" , "r" , "d!" }
strsplit("hello world!" , "ol")	{ "he" , " w" , "r" , "d!" }
strsplit("hello world!" , "l ")	{ "he" , "o" , "wor" , "d!" }

ANSWERS :

Q1:

```
/* Developed By Amir Mohammad Tavakolli */

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

double ascii_to_e(char *);

int main(void) {
    printf("%f", ascii_to_e("1213.3e-2"));
    return 0;
}

double ascii_to_e(char *num) {
    double number;
    int counter_float, is_float, sign; //number of digit after dot
    number = 0; // actual number
    counter_float = is_float = 0;
    sign = 1;
    if (*num == '-') {
        sign = -1;
        ++num;
    }
    if (*num == '+') {
        ++num;
    }
    while (*num != 'e' && *num != '\0') {
        if (*num == '.') {
            is_float = 1; // this is number is float
            ++num;
            continue;
        }
        if (!isdigit(*num)){
            break;
        }
        number = 10 * number + (*num - '0');
        ++num;
        if (is_float) {
            counter_float++; // count digits after dot
        }
    }
}
```

```

    }
    int i;
    for (i = 0; i < counter_float; ++i) {
        number /= 10; // get float
    }
    number *= sign;
    if (*num == '\\0') {
        return number;
    }
    if (*num == 'e') {
        num++;
    }
    sign = 1;
    if (*num == '-') {
        sign = -1;
        ++num;
    }
    if (*num == '+') {
        ++num;
    }
    int power;
    power = 0;
    while (*num != '\\0' && isdigit(*num)) {
        power = 10 * power + (*num - '0');
        ++num;
    }
    if (sign < 0) {
        int j;
        for (j = 0; j < power; ++j) {
            number /= 10;
        }
    }
    else {
        int j;
        for (j = 0; j < power; ++j) {
            number *= 10;
        }
    }
    return number;
}

```

Q3 :

```
/* Developed By Amir Mohammad Tavakolli */
#include <string.h>
#include <stdio.h>

#define M 5
#define N 20

void strsplit(char *source, char *separators, char result[][N]);

int char_in_str(char *, char);

int main(void){
    char a[M][N];
    strsplit("Hello world!", " ", a);
    for(int i = 0; i < M; ++i){
        printf("%s\n", a[i]);
    }
}

int char_in_str(char *source, char key) {
    int i;
    for (i = 0; i < strlen(source); ++i) {
        if (source[i] == key) {
            return 1;
        }
    }
    return 0;
}

void strsplit(char *source, char *separators, char result[][N]) {
    int i, j;
    i = j = 0;
    while (*source != '\0') {
        if (char_in_str(separators, *source)) {
            result[i][j] = '\0';
            i++;
            j = 0;
        }
        result[i][j] = *source;
        j++;
        source++;
    }
    result[i][j] = '\0';
}
```

```
        result[i][j] = '\\0';
        i += 1;
        j = 0;
        while (char_in_str(separators, *source)){
            ++source;
        }
    }
    else {
        result[i][j] = *source;
        j++;
        ++source;
    }
}

}
```