CS 3432 – Computer Organization

Lab 2.1 – Arrays Author: Robert Alvarez

**Instructions:**

Assignments must be submitted with the same person as Lab 1, unless the other person drops the course, through GitHub Classroom. Use the following link to access the assignment: <https://classroom.github.com/a/XGW7mwbD>

Only assistance from the instructor, TA, or IA will be permitted.

**Part 1 – Familiarization with an array of pointers and characters.**

This part of the lab would focus on implementing your own “string.h” library and a tokenize function using some of the functions.

my\_string: (Note: You are not allowed to index the pointers☺) [5 points each]

1. Implement strlen.
   1. size\_t str\_len(const char \*s);
2. Implement strcmp and strncmp.
   1. int str\_cmp(const char \*s1, const char \*s2);
   2. int str\_n\_cmp(const char \*s1, const char \*s2, size\_t n);
3. Implement memcpy.
   1. void \*mem\_cpy(void \*restrict dst, const void \*restrict src, size\_t n);
4. Implement strchr.
   1. char \*str\_chr(const char \*s, int c);
5. Implement strpbrk.
   1. char \*str\_p\_brk(const char \*s, const char \*charset);
6. Implement strsep.
   1. char \*str\_sep(char \*\*stringp, const char \*delim);
7. Implement strcat.
   1. char \*str\_cat(char \*s1, const char \*s2);

Tokenizer: [10 points]

1. Make a function defined as “char \*\*tokenize(char \*str, const char \*delims)”.
   1. This function takes two pointers to an array of null-terminated characters.
      1. str: Array of characters split into tokens based on delims characters.
      2. delims: Array of characters used to split “str.”
   2. This function makes a null-terminated array of character pointers with k, not null pointers where k is defined as the number of times when str[i] == delims[j] where 0 <= i < length of str and p <= j <= length of delims.
      1. The pointers in the return value most point to memory places in str. Therefore, you must terminate each substring null and add a pointer to the beginning of the substring in str.

Example:

If str starts at 0x100, in memory, with the value of “Dodge,Challenger.Charger/Durango” and delims = “,./”. Then, tokenizer(str,delims) must return an array with pointers pointing to 0x100, 0x106, 0x111 and 0x119.

File/Libraries:

1. my\_string.h and my\_string.c.
2. tokenizer.h and tokenizer.c.
3. Makefile.
   1. It would be provided to you to compile and link main.c with the required files.

Permitted standard libraries:

* stdlib.h - size\_t

No standard C library can be included if you want to use any other library or other variables/functions from stdlib.h, contact the AI or TA, and we can help you to figure out how to do the same thing without that library or function.

Deadline (GitHub Classroom): February 7th, 2023, by 11:59 pm.

1. Source code (Only .c and .h files, and 1 Makefile)

Grading: (Total of 45 points)

Each function has the weight grade value of its implementation. We would run each function 20 times with randomly generated values and compare your output with ours. Based on the match percentage, it is how much you would get for it. For example, if your str\_len() output is only correct for 15 out of 20 runs, you would get only (15/20)\*5 points = 3.75 out of the 5 maximum possible.

***NOTE:***

Once the deadline is passed, the answer for part 1 will be given so everyone can start part 2 on time. This means that every submission after the deadline would not be considered and if that was your only submission, you get a zero.