**Course**: ENSF 614 – Fall 2023

**Lab #**: Lab 2

**Instructor**: Moussavi

**Student Name**: Yajur Vashisht, Balkaran Gill

**Submission Date**: September 27th, 2023

**Part A**

**A diagram of a computer program

Description automatically generated with medium confidence**

**Part B**

#include <iostream>

#include <cstring>

using namespace std;

int main(void)

{

char str1[7] = "banana";

const char str2[] = "-tacit";

const char\* str3 = "-toe";

*/\* point 1 \*/*

char str5[] = "ticket";

char my\_string[100]="";

int bytes;

int length;

length = (int) my\_strlen(my\_string);

cout << "\nLine 1: my\_string length is " << length;

*/\* using sizeof operator \*/*

bytes = sizeof (my\_string);

cout << "\nLine 2: my\_string size is " << bytes << " bytes.";

*/\* using strcpy libarary function \*/*

strcpy(my\_string, str1);

cout << "\nLine 3: my\_string contains: " << my\_string;

length = (int) my\_strlen(my\_string);

cout << "\nLine 4: my\_string length is " << length << ".";

my\_string[0] = '\0';

cout << "\nLine 5: my\_string contains:\"" << my\_string << "\"";

length = (int) my\_strlen(my\_string);

cout << "\nLine 6: my\_string length is " << length << ".";

bytes = sizeof (my\_string);

cout << "\nLine 7: my\_string size is still " << bytes << " bytes.";

*/\* strncat append the first 3 characters of str5 to the end of my\_string \*/*

my\_strncat(my\_string, str5, 3);

cout << "\nLine 8: my\_string contains:\"" << my\_string << "\"";

length = (int) my\_strlen(my\_string);

cout << "\nLine 9: my\_string length is " << length << ".";

my\_strncat(my\_string, str2, 4);

cout << "\nLine 10: my\_string contains:\"" << my\_string << "\"";

*/\* strncat append ONLY up ot '\0' character from str3 -- not 6 characters \*/*

my\_strncat(my\_string, str3, 6);

cout << "\nLine 11: my\_string contains:\"" << my\_string << "\"";

length = (int) my\_strlen(my\_string);

cout << "\nLine 12; my\_string has " << length << " characters.";

cout << "\n\nUsing my\_strcmp: ";

cout << "\n\"ABCD\" is less than \"ABCDE\" ... my\_strcmp returns: " <<

my\_strcmp("ABCD", "ABCDE");

cout << "\n\"ABCD\" is less than \"ABND\" ... my\_strcmp returns: " <<

my\_strcmp("ABCD", "ABND");

cout << "\n\"ABCD\" is equal than \"ABCD\" ... my\_strcmp returns: " <<

my\_strcmp("ABCD", "ABCD");

cout << "\n\"ABCD\" is less than \"ABCd\" ... my\_strcmp returns: " <<

my\_strcmp("ABCD", "ABCd");

cout << "\n\"Orange\" is greater than \"Apple\" ... my\_strcmp returns: " <<

my\_strcmp("Orange", "Apple") << endl;

return 0;

}

int my\_strlen(const char \*s) {

int i = 0;

while (s[i] != '\0') {

i++;

}

return i;

}

void my\_strncat(char \*dest, const char \*source, int n) {

int dest\_len = my\_strlen(dest);

int i;

for (i = 0; i < n && source[i] != '\0'; i++) {

dest[dest\_len + i] = source[i];

}

dest[dest\_len + i] = '\0';

}

#include <iostream>

int my\_strcmp(const char\* str1, const char\* str2) {

int sum1 = 0;

int sum2 = 0;

while (\*str1) {

sum1 += static\_cast<int>(\*str1);

str1++;

}

while (\*str2) {

sum2 += static\_cast<int>(\*str2);

str2++;

}

return sum1 - sum2;

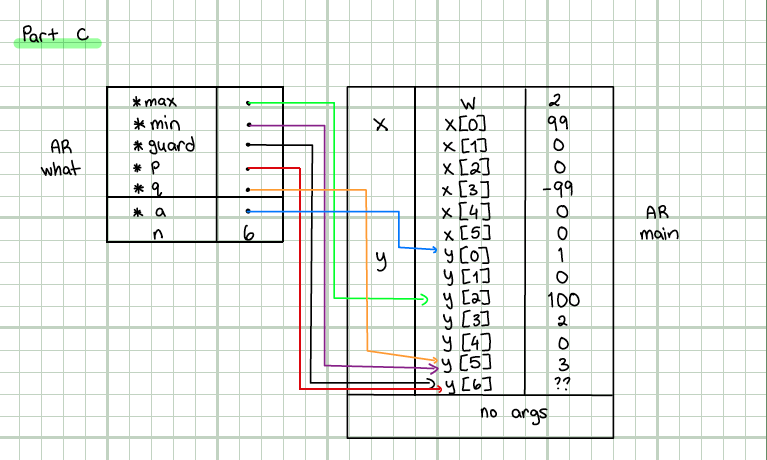
}

Output:

A screenshot of a computer program

Description automatically generated

**Part C**



**Part E**

#include "lab2exe\_E.h"

cplx cplx\_add(cplx z1, cplx z2)

{

cplx result;

result.real = z1.real + z2.real;

result.imag = z1.imag + z2.imag;

return result;

}

void cplx\_subtract(cplx z1, cplx z2, cplx \*difference) {

difference->real = z1.real - z2.real;

difference->imag = z1.imag - z2.imag;

}

void cplx\_multiply(const cplx \*pz1, const cplx \*pz2, cplx \*product) {

product->real = (pz1->real \* pz2->real) - (pz1->imag \* pz2->imag);

product->imag = (pz1->real \* pz2->imag) + (pz1->imag \* pz2->real);

}