

Course: ENSF 694 – Summer 2024

Lab Assignment #: Lab 1

Instructor: Mahmood Moussavi

Student Name: Jeff Wheeler, UCID: 30265340

Submission Date: July 3, 2023

Exercise A

Code:

```
/*
 * lab1exe_A.cpp
 * ENSF 694 Lab 1 - Exercise A
 * Created by Mahmood Moussavi
 * Completed by: Jeff Wheeler
 * Submission date: July 3, 2024
 */

#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

const double G = 9.8;    /* gravitation acceleration 9.8 m/s^2 */
const double PI = 3.141592654; /* constant for pi */

void create_table(double v);
/*
 * REQUIRES:
 *   v - velocity in m/s
 * PROMISES:
 *   Using a given velocity the function creates a table of projectile's
 *   maximum travel distance
 *   and time for trajectory angles of 0 to 90 degrees in increments of 5
 *   degrees
 */

double Projectile_travel_time(double a, double v);
/*
 * REQUIRES:
 *   a - angle in degrees
 *   v - velocity in m/s
 * PROMISES:
 *   Calculates projectile travel time for a given angle and velocity
 */

double Projectile_travel_distance(double a, double v);
/*
 * REQUIRES:
 *   a - angle in degrees
 *   v - velocity in m/s
 * PROMISES:
 *   Calculates projectile travel distance for a given angle and velocity
 */
```

```

double degree_to_radian(double d);
/*
 * REQUIRES:
 *   d - angle in degrees
 * PROMISES:
 *   Converts a degree angle into radians
 */

int main(void)
{
    double velocity;

    cout << "Please enter the velocity at which the projectile is
    launched (m/sec): ";
    cin >> velocity;

    if(!cin) // means if cin failed to read
    {
        cout << "Invalid input. Bye...\n";
        exit(1);
    }

    while (velocity < 0 )
    {
        cout << "\nplease enter a positive number for velocity: ";
        cin >> velocity;
        if(!cin)
        {
            cout << "Invalid input. Bye...";
            exit(1);
        }
    }

    std::cout << std::fixed;
    std::cout << std::setprecision(3);
    create_table(velocity);

    return 0;
}

void create_table(double v){
    cout << "\nAngle" << "\t\t" << "t" << "\t\t" << "d\n";
    cout << "(deg)" << "\t\t" << "(sec)" << "\t\t" << "(m)\n";

```

```

        for (double a = 0.00000; a <= 90.00000; a += 5.00000){
            cout << a << "\t\t" << Projectile_travel_time(a,v) << "\t\t"
            << Projectile_travel_distance(a,v) << "\n";
        }
    }

double Projectile_travel_time(double a, double v){
    return 2 * v * sin(degree_to_radian(a)) / G;
}

double Projectile_travel_distance(double a, double v){
    return pow(v,2) / G * sin(2 * degree_to_radian(a));
}

double degree_to_radian(double d){
    return d * PI / 180;
}

```

Program Output:

```

jeffw@DESKTOP-SR7596T /cygdrive/c/Users/jeffw/Documents/_Software Masters/ENSF 694/ensf694_assignment1
$ ./exercise_A
Please enter the velocity at which the projectile is launched (m/sec): 150

```

Angle (deg)	t (sec)	d (m)
0.000	0.000	0.000
5.000	2.668	398.682
10.000	5.316	785.250
15.000	7.923	1147.959
20.000	10.470	1475.788
25.000	12.937	1758.776
30.000	15.306	1988.324
35.000	17.558	2157.458
40.000	19.677	2261.038
45.000	21.646	2295.918
50.000	23.450	2261.038
55.000	25.076	2157.458
60.000	26.511	1988.324
65.000	27.744	1758.776
70.000	28.766	1475.788
75.000	29.569	1147.959
80.000	30.147	785.250
85.000	30.496	398.682
90.000	30.612	-0.000

Exercise B – Part 2

SUBJECT: Ex B2

AR
bar

No Local

a

•

b

•

AR
quux

n

503

p

•

q

•

AR
main

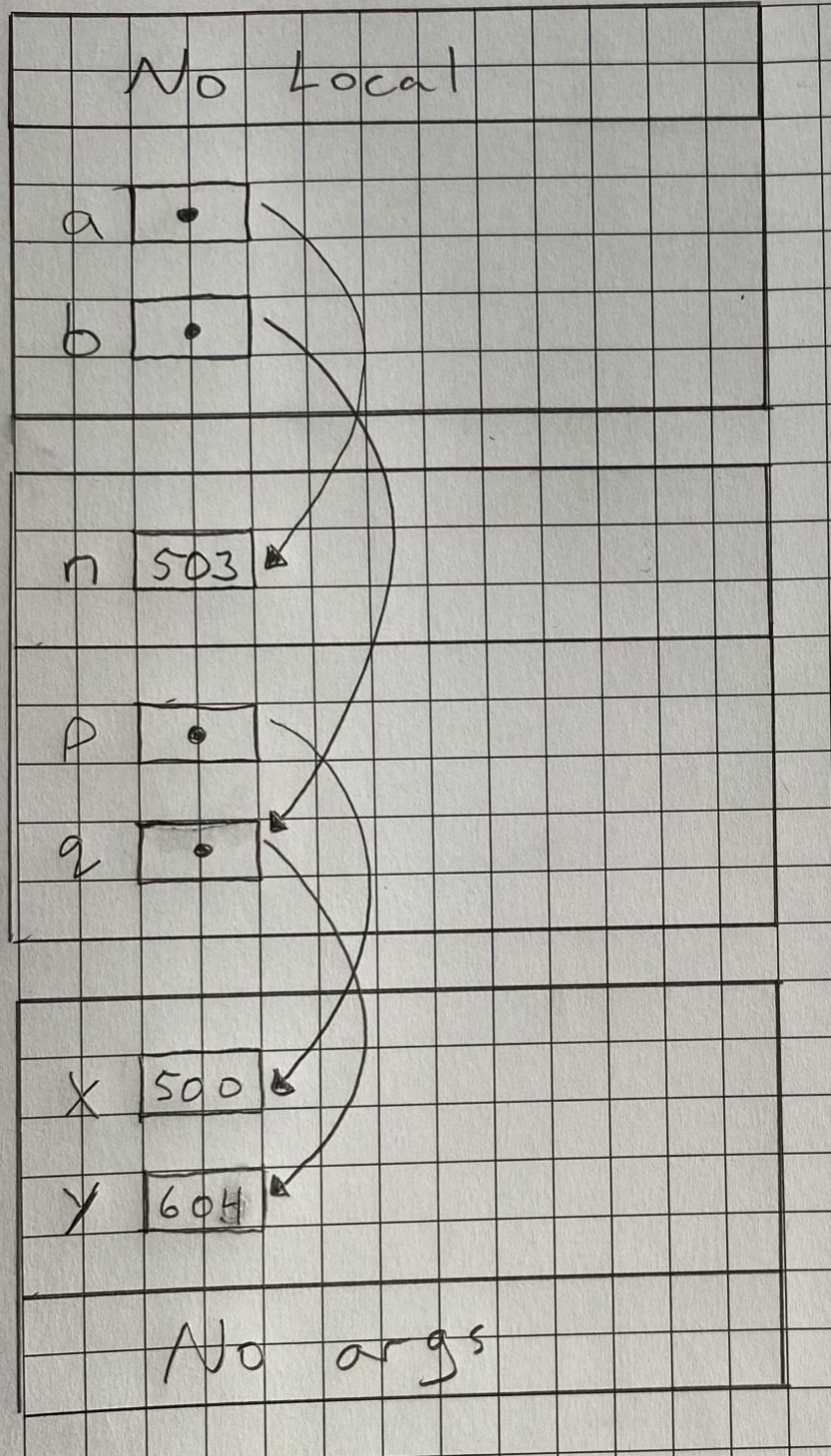
x

500

y

604

No args



Exercise C

Code:

```
/*
 * lab1exe_C.cpp
 * ENSF 694 Lab 1 - Exercise C
 * Created by Mahmood Moussavi
 * Completed by: Jeff Wheeler
 * Submission date: July 3, 2024
 */

#include <iostream>
using namespace std;

void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr);
/*
 * Converts time in milliseconds to time in minutes and seconds.
 * For example, converts 123400 ms to 2 minutes and 3.4 seconds.
 * REQUIRES:
 *     ms_time >= 0.
 *     minutes_ptr and seconds_ptr point to variables.
 * PROMISES:
 *     0 <= *seconds_ptr & *seconds_ptr < 60.0
 *     *minutes_ptr minutes + *seconds_ptr seconds is equivalent to
 *     ms_time ms.
 */

int main(void)
{
    int millisec;
    int minutes;
    double seconds;

    cout << "Enter a time interval as an integer number of milliseconds: ";

    // printf("Enter a time interval as an integer number of milliseconds:
    ");
    cin >> millisec;

    if (!cin) {
        cout << "Unable to convert your input to an int.\n";
        exit(1);
    }

    cout << "Doing conversion for input of " << millisec << " milliseconds
    ... \n";
```

```

/* MAKE A CALL TO time_convert HERE. */
time_convert(millisec, &minutes, &seconds);

cout << "That is equivalent to " << minutes << " minute(s) and " <<
    seconds << " second(s).\n";
return 0;
}

/* PUT YOUR FUNCTION DEFINITION FOR time_convert HERE. */
void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr){
    double total_seconds = ms_time / 1000.0;
    *minutes_ptr = (int) total_seconds / 60;
    *seconds_ptr = total_seconds - *minutes_ptr * 60;
}

```

Output:

```

jeffw@DESKTOP-SR7596T /cygdrive/c/Users/jeffw/Documents/_Software Masters/ENSF 694/ensf694_assignment1
● $ ./exercise_C
Enter a time interval as an integer number of milliseconds: 123400
Doing conversion for input of 123400 milliseconds ...
That is equivalent to 2 minute(s) and 3.4 second(s).

```

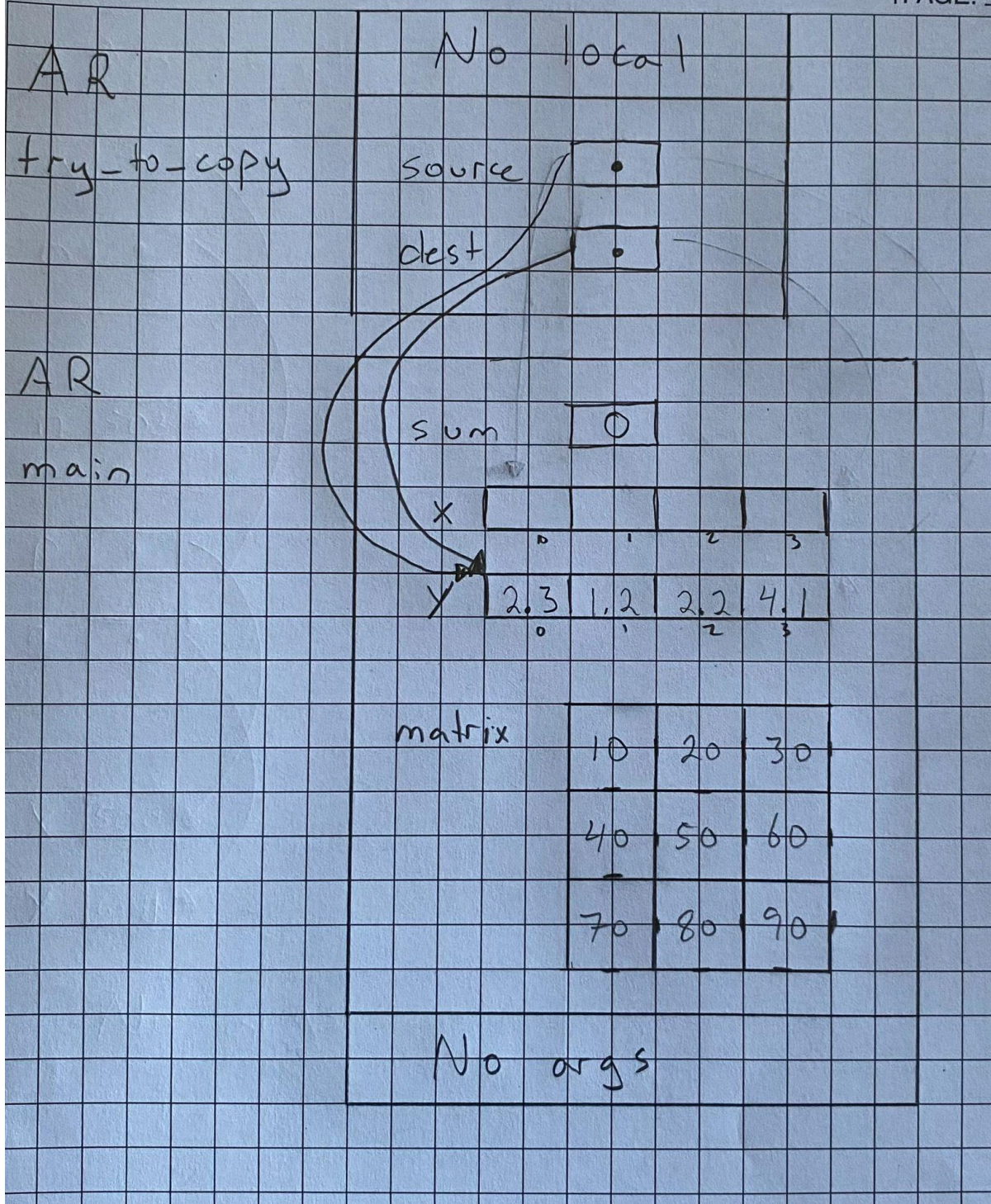

Exercise D – Part 1

Point 1

SUBJECT: Ex D: Point 1

JOB: _

PAGE: _

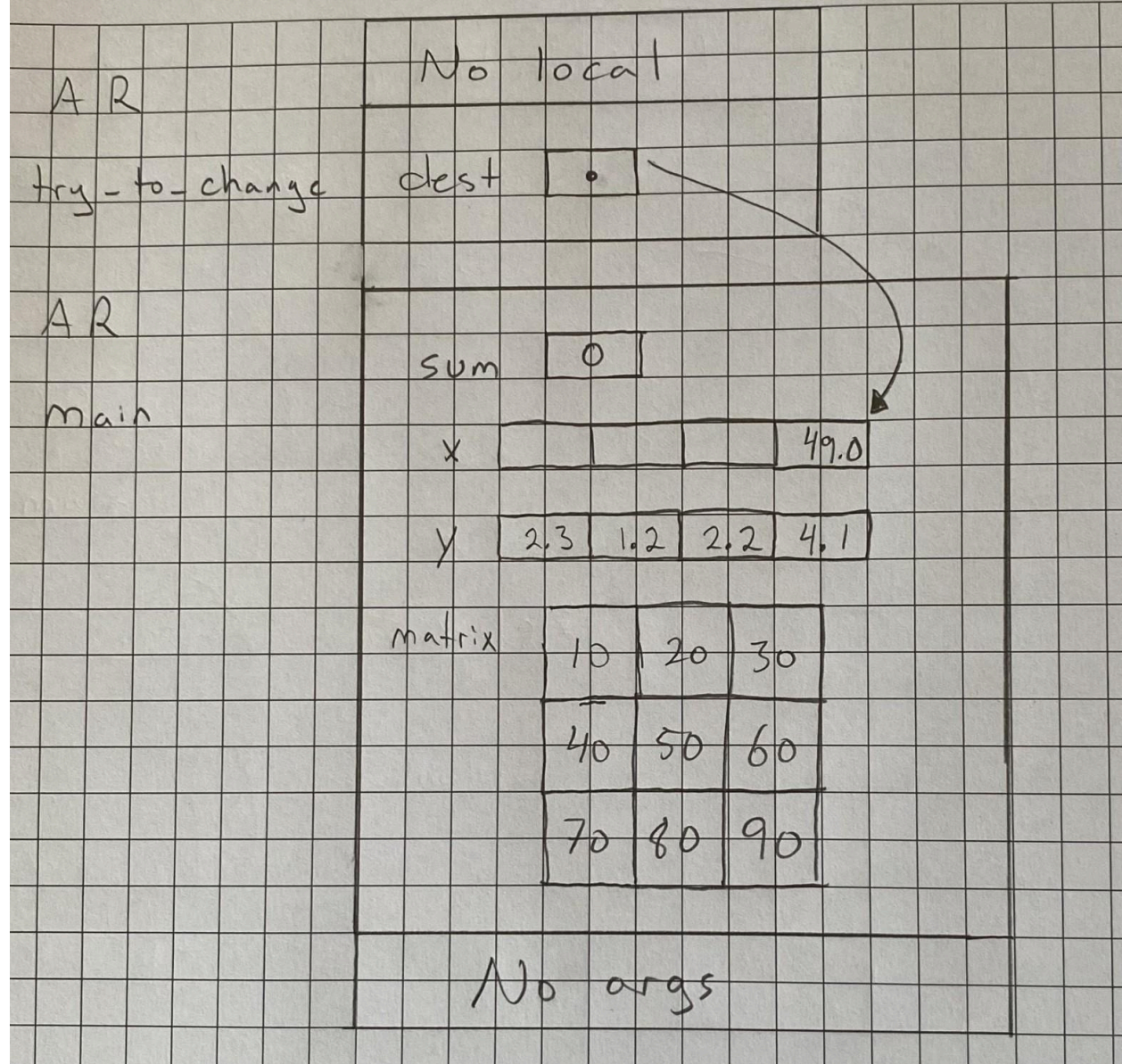


Point 2

SUBJECT: Ex D: Point 2

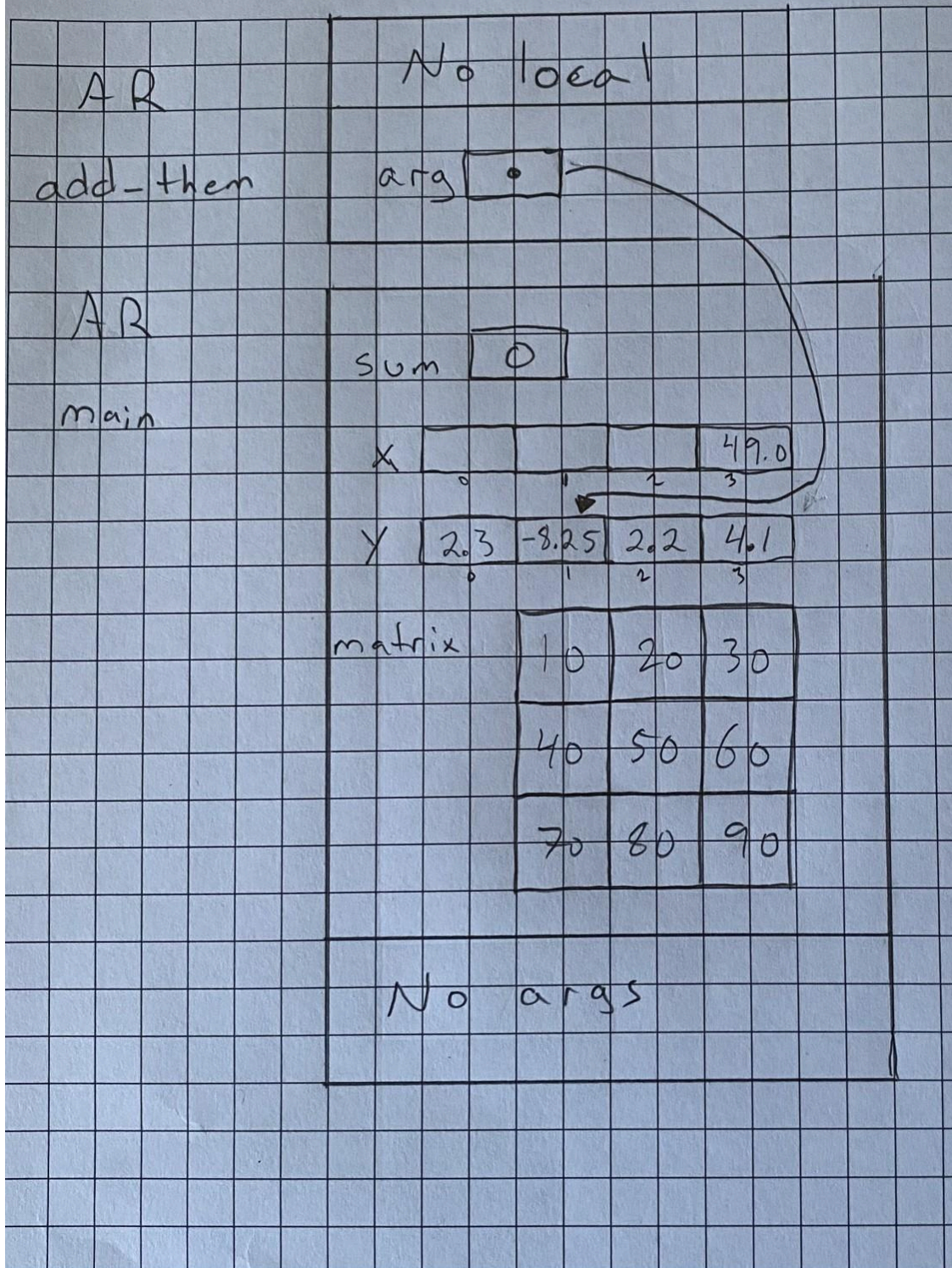
JOB:

PAGE: _____



Point 3

SUBJECT: Ex D: Point 3



Exercise D – Part 2

Code:

```
/*
 * lab1exe_D.cpp
 * ENSF 694 Lab 1 - Exercise D
 * Created by Mahmood Moussavi
 * Completed by: Jeff Wheeler
 * Submission date: July 3, 2024
 */

#include <iostream>
#include <iomanip>
using namespace std;
const int COL_SIZE = 3;
const int ROW_SIZE = 3;

void try_to_change(double* dest);
/*
 * REQUIRES: Destination point
 * PROMISES: Trys to change the destination point in an incorrect way
 */

void try_to_copy(double dest[], double source[]);
/*
 * REQUIRES: Destination & source array
 * PROMISES: Trys to copy the source array to the destination array in an
incorrect way
 */

double add_them (double a[5]);
/*
 * REQUIRES: An Array
 * PROMISES: Trys to add points within an array in an incorrect way
 */

void print_matrix(double matrix[][COL_SIZE], int rows);
/*
 * PROMISES: displays the values in the elements of the 2-D array, matrix,
 * formatted in rows columns separated with one or more spaces.
 */
```

```

void good_copy(double *dest, double *source, int n);
/* REQUIRES: dest and source points to two array of double numbers with n
 * to n-1 elements
 * PROMISES: copies the values in each element of array source to the
 * corresponding element
 * in array dest.
 */
int main(void)
{
    double sum = 0;
    double x[4];
    double y[] = {2.3, 1.2, 2.2, 4.1};
    double matrix[ROW_SIZE][COL_SIZE] = { {10, 20, 30}, {40, 50, 60}, {70,
        80, 90}};
    cout << " sizeof(double) is " << (int) sizeof(double) << " bytes.\n";
    cout << " size of x in main is: " << (int) sizeof(x) << " bytes.\n";
    cout << " y has " << (int) (sizeof(y)/ sizeof(double)) << " elements
        and its size is: "
        << (int) sizeof(y) << " bytes.\n";
    cout << " matrix has " << (int) (sizeof(matrix)/ sizeof(double)) << "
        elements and its size is: "
        << (int) sizeof(matrix) << " bytes.\n";

    try_to_copy(x, y);
    try_to_change(x);

    sum = add_them(&y[1]);
    cout << "\n sum of values in y[1], y[2] and y[3] is: " << sum << endl;

    good_copy(x, y, 4);

    cout << "\nThe values in array x after call to good_copy are expected
        to be:";
    cout << "\n2.30, -8.25, 2.20, 4.10\n";
    cout << "And the values are:\n";
    for(int i = 0; i < 4; i++)
        cout << fixed << setprecision(2) << x[i] << " ";

    cout << "\nThe values in matrix are:\n";
    print_matrix(matrix, 3);

    cout << "\nProgram Ends...\n";

    return 0;
}

```

```

void try_to_copy(double dest[], double source[])
{
    dest = source;

    /* point one*/

    return;
}

void try_to_change(double* dest)
{
    dest [3] = 49.0;

    /* point two*/
    cout << "\n sizeof(dest) in try_to_change is "<< (int)sizeof(dest) << "
        bytes.\n";
    return;
}

double add_them (double arg[5])
{
    *arg = -8.25;

    /* point three */
    cout << "\n sizeof(arg) in add_them is " << (int) sizeof(arg) << "
        bytes.\n";
    cout << "\n Incorrect array size computation: add_them says arg has "
        << (int) (sizeof(arg)/sizeof(double)) << " element.\n";

    return arg[0] + arg[1] + arg[2];
}

void good_copy(double *dest, double *source, int n)
{
    for (int i= 0; i < n; i++){
        dest [i] = source [i];
    }
}

```



```

void print_matrix(double matrix[][COL_SIZE], int rows)
{
    for (int i = 0; i < rows; i++){
        for (int j = 0; j < COL_SIZE; j++){
            cout << matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

```

Output:

```

jeffw@DESKTOP-SR7596T /cygdrive/c/Users/jeffw/Documents/_Software Masters/ENSF 694/ensf694_assignment1
$ ./exercise_D
sizeof(double) is 8 bytes.
size of x in main is: 32 bytes.
y has 4 elements and its size is: 32 bytes.
matrix has 9 elements and its size is: 72 bytes.

sizeof(dest) in try_to_change is 8 bytes.

sizeof(arg) in add_them is 8 bytes.

Incorrect array size computation: add_them says arg has 1 element.

sum of values in y[1], y[2] and y[3] is: -1.95

The values in array x after call to good_copy are expected to be:
2.30, -8.25, 2.20, 4.10
And the values are:
2.30 -8.25 2.20 4.10
The values in matrix are:
10.00 20.00 30.00
40.00 50.00 60.00
70.00 80.00 90.00

Program Ends...

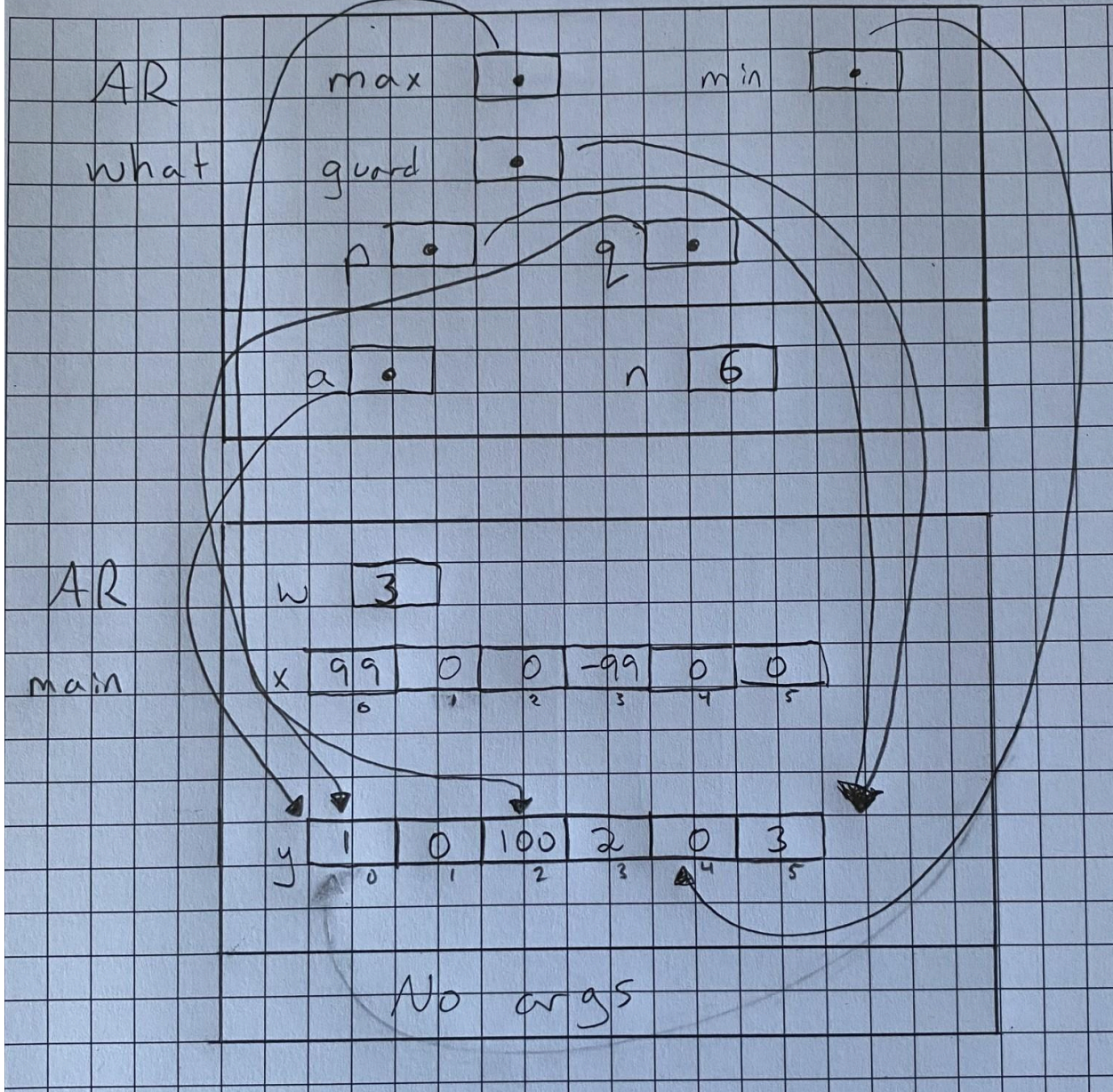
```

Exercise E

SUBJECT: Ex E : Point 1

JOB:

PAGE: ____ OF: ____



Exercise F

Code:

```
#include "MyArray.h"

int search(const MyArray* myArray, int obj){
    for (int i = 0; i < SIZE; i++){
        if (myArray->array[i] == obj){
            return i;
        }
    }
    return -1;
}

void initialize(MyArray* myArray) {
    myArray->list_size = 0;
}

int retrieve_at(MyArray* myArray, int pos){
    return myArray->array[pos];
}

int count(MyArray* myArray, int obj){
    int count = 0;
    for (int i = 0; i < myArray->list_size; i++){
        if (myArray->array[i] == obj){
            count++;
        }
    }
    return count;
}

void append(MyArray* myArray, int array[], int n) {
    if ((myArray->list_size + n) <= SIZE){
        for (int i = 0; i < n; i++){
            myArray->array[myArray->list_size + i] = array[i];
        }
    }
}
```

```

void insert_at(MyArray* myArray, int pos, int val) {
    for (int i = SIZE - 1; i > pos; i--){
        myArray->array[i] = myArray->array[i+1];
    }
    myArray->array[pos] = val;
    myArray->list_size++;
}

int remove_at(MyArray* myArray, int pos) {
    int removed = myArray->array[pos];
    for (int i = pos; i < SIZE; i++){
        myArray->array[i] = myArray->array[i+1];
    }
    myArray->list_size--;
    return removed;
}

int remove_all(MyArray* myArray, int value) {
    int removed = 0;
    for (int i = myArray->list_size - 1; i >= 0; i--){
        myArray->array[i] = 0;
        removed++;
    }
    return removed;
}

void display_all(MyArray* myArray) {
    for (int i = 0; i < myArray->list_size; i++){
        cout << myArray->array[i] << " ";
    }
}

bool is_full(MyArray* myArray){
    if (myArray->list_size == SIZE){
        return true;
    }
    return false;
}

bool isEmpty(MyArray* myArray){
    if (myArray->list_size == 0){
        return true;
    }
    return false;
}

```

```
int size(MyArray* myArray){  
    return myArray->list_size;  
}
```

Output:

```
jeffw@DESKTOP-SR7596T /cygdrive/c/Users/jeffw/Documents/_Software Masters/ENSF 694/ensf694_assignment1  
● $ ./myProgram  
Starting Test Run. Using input file.  
Line 1 >> Passed  
Line 2 >> Passed  
Line 3 >> Passed  
Line 4 >> Passed  
Line 5 >> Passed  
Line 6 >> Passed  
Line 7 >> Passed  
Line 8 >> Passed  
Line 9 >> Passed  
Line 10 >> Passed  
Line 11 >> Passed  
Line 12 >> Passed  
Line 13 >> Passed  
Line 14 >> Passed  
Line 15 >> Passed  
Line 16 >> Passed  
Line 17 >> Passed  
Line 18 >> Passed  
Line 19 >> Passed  
Exiting...  
Finishing Test Run  
Showing Data in the List:  
101 200 100 500  
Program Ended ....
```