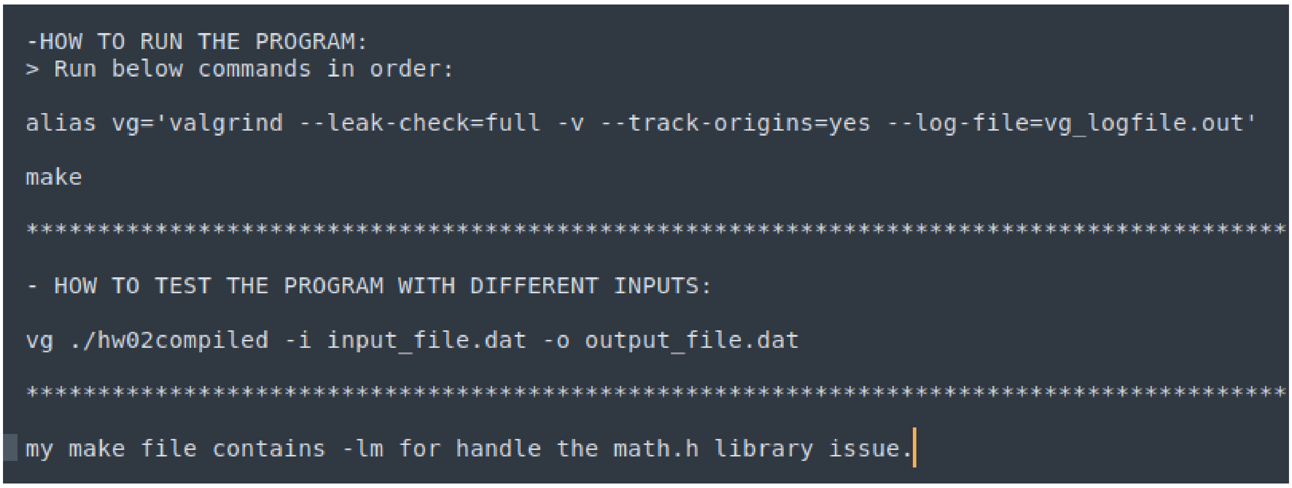
Systems Programming HW2 Report
Mohammad Ashraf Yawar 161044123
March 30, 2022

- HOW TO RUN AND TEST THE PROGRAM WITH DIFFERENT INPUTS?
- Working DEMO link: https://youtu.be/oA8G_84yfBQ

- You can find the instructions in README.txt in order to run and test the program.



```
-HOW TO RUN THE PROGRAM:
> Run below commands in order:

alias vg='valgrind --leak-check=full -v --track-origins=yes --log-file=vg_logfile.out'

make

****************************************************************************************

- HOW TO TEST THE PROGRAM WITH DIFFERENT INPUTS:

vg ./hw02compiled -i input_file.dat -o output_file.dat

****************************************************************************************

my make file contains -lm for handle the math.h library issue.
```

Implemented Concepts:

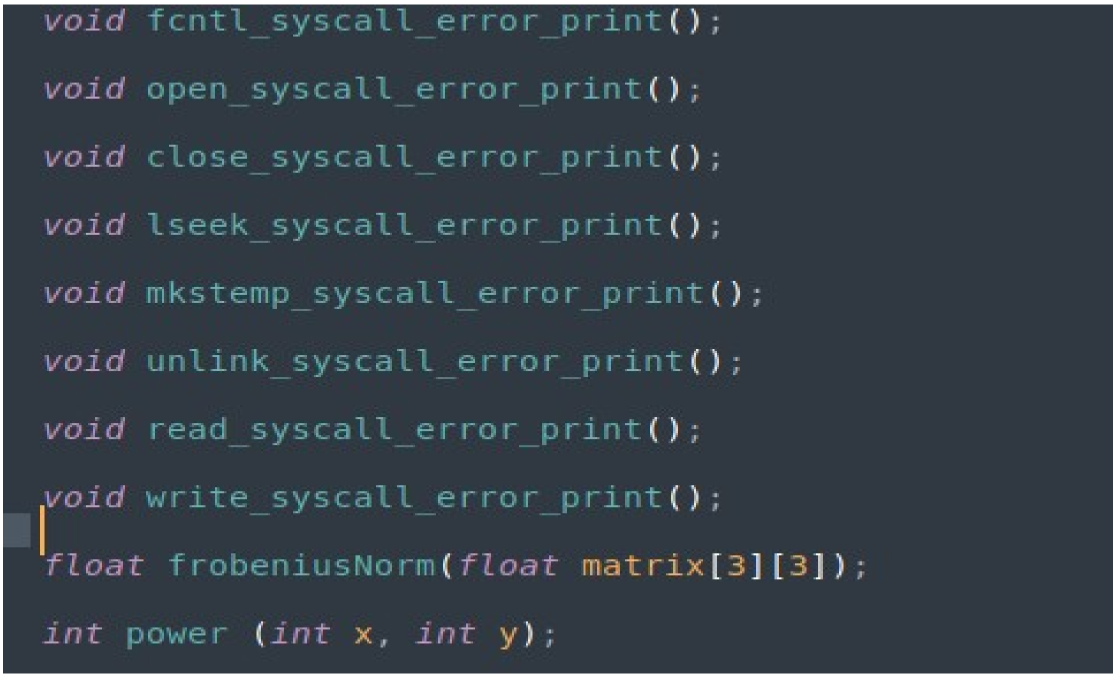
- File read, write, lock using syscalls.
- Signal handaling, parent child signal relations.
- Multiple child process, fork, exec family.
- Make files.
- Waiting for the child process to finish their task.
-

Working Cases:
- This program works for cases all the cases.
- Works on **relative path**.

Note Working Cases:
- NONE

Design Explanation:

- I separated my program into functions using make file where each function does a specific task:



```
void fcntl_syscall_error_print();

void open_syscall_error_print();

void close_syscall_error_print();

void lseek_syscall_error_print();

void mkstemp_syscall_error_print();

void unlink_syscall_error_print();

void read_syscall_error_print();

void write_syscall_error_print();

float frobeniusNorm(float matrix[3][3]);

int power (int x, int y);
```

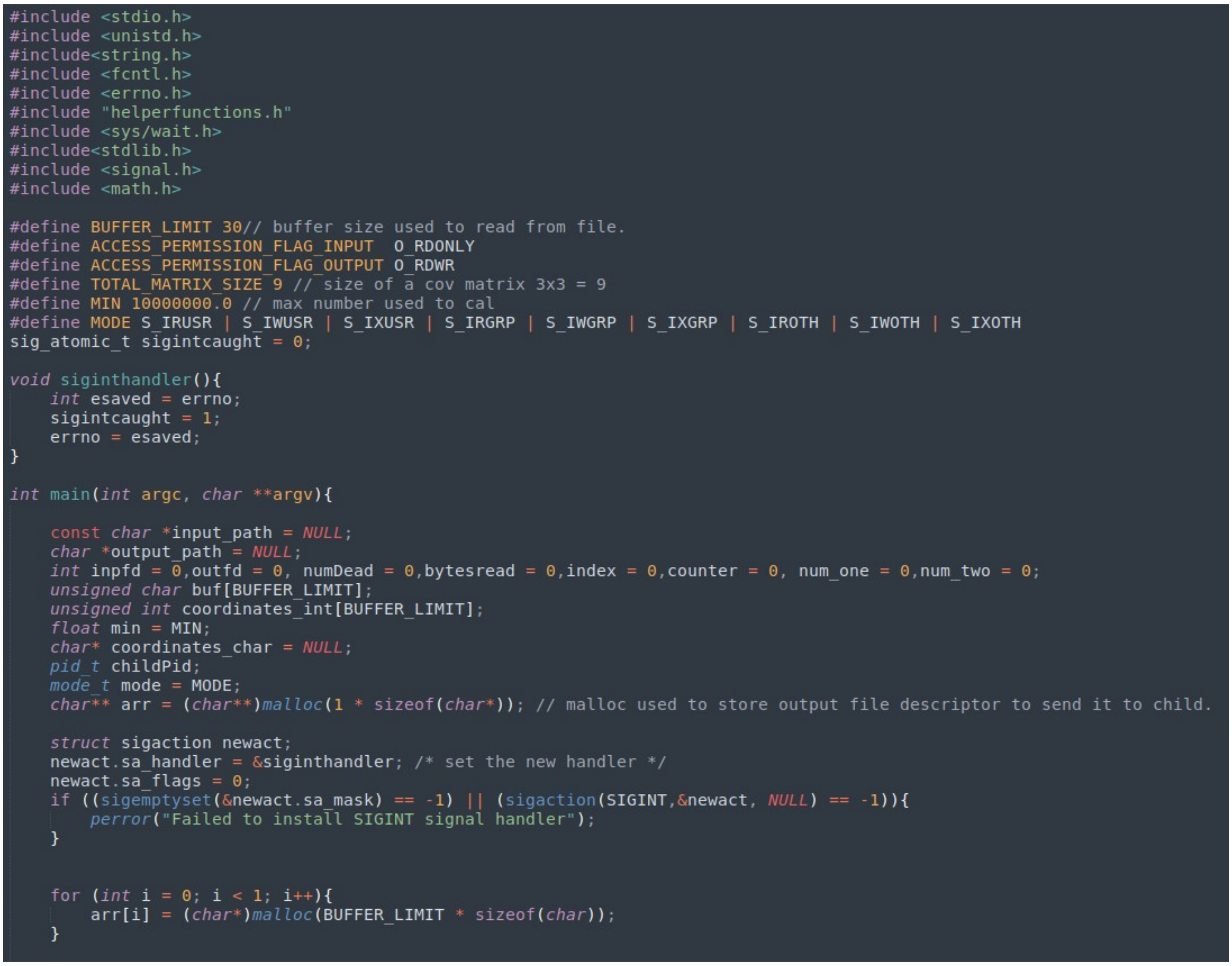
- All the System-Calls and their possible return error values are checked with detailed errno checks.

- the program start by setting some variables and setting the signal handler mechanism to catch in SIGINT signal.
- when ever the user press control+c while the program is running, the handler will catch the signal and notify the child, clean up the necessary spaces and close the file and exit elegantly.

- NOTE: while testing the program each time, the output_file.dat file should be set empty (remove the prev data).

- I created two .c files , one for parent and one for it's children to run.
- once I read the file into the program, I create n number of child according to the size of the input file (it's dynamically taken care of). And create the child process using that data.

- first convert the 30 chars into int's and then convert them into string array so that we can send it in exev() syscall.



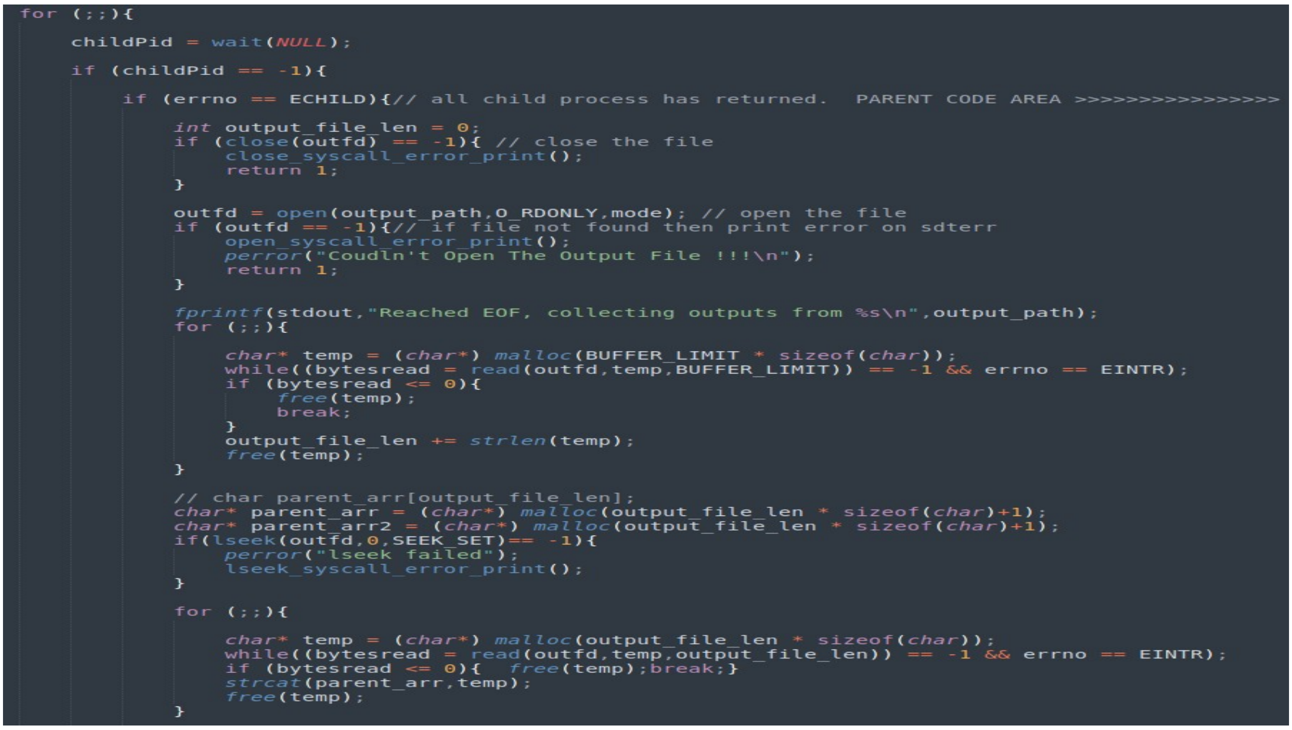
```c
#include <stdio.h>
#include <unistd.h>
#include<string.h>
#include <fcntl.h>
#include <errno.h>
#include "helperfunctions.h"
#include <sys/wait.h>
#include<stdlib.h>
#include <signal.h>
#include <math.h>

#define BUFFER_LIMIT 30// buffer size used to read from file.
#define ACCESS_PERMISSION_FLAG_INPUT  O_RDONLY
#define ACCESS_PERMISSION_FLAG_OUTPUT O_RDWR
#define TOTAL_MATRIX_SIZE 9 // size of a cov matrix 3x3 = 9
#define MIN 10000000.0 // max number used to cal
#define MODE S_IRUSR | S_IWUSR | S_IXUSR | S_IRGRP | S_IWGRP | S_IXGRP | S_IROTH | S_IWOTH | S_IXOTH
sig_atomic_t sigintcaught = 0;

void siginthandler(){
    int esaved = errno;
    sigintcaught = 1;
    errno = esaved;
}

int main(int argc, char **argv){

    const char *input_path = NULL;
    char *output_path = NULL;
    int inpfd = 0,outfd = 0, numDead = 0,bytesread = 0,index = 0,counter = 0, num_one = 0,num_two = 0;
    unsigned char buf[BUFFER_LIMIT];
    unsigned int coordinates_int[BUFFER_LIMIT];
    float min = MIN;
    char* coordinates_char = NULL;
    pid_t childPid;
    mode_t mode = MODE;
    char** arr = (char**)malloc(1 * sizeof(char*)); // malloc used to store output file descriptor to send it to child.

    struct sigaction newact;
    newact.sa_handler = &siginthandler; /* set the new handler */
    newact.sa_flags = 0;
    if ((sigemptyset(&newact.sa_mask) == -1) || (sigaction(SIGINT,&newact, NULL) == -1)){
        perror("Failed to install SIGINT signal handler");
    }


    for (int i = 0; i < 1; i++){
        arr[i] = (char*)malloc(BUFFER_LIMIT * sizeof(char));
    }
```

- parent waits until all the child process are returned and then parent runs and calculates the final results.

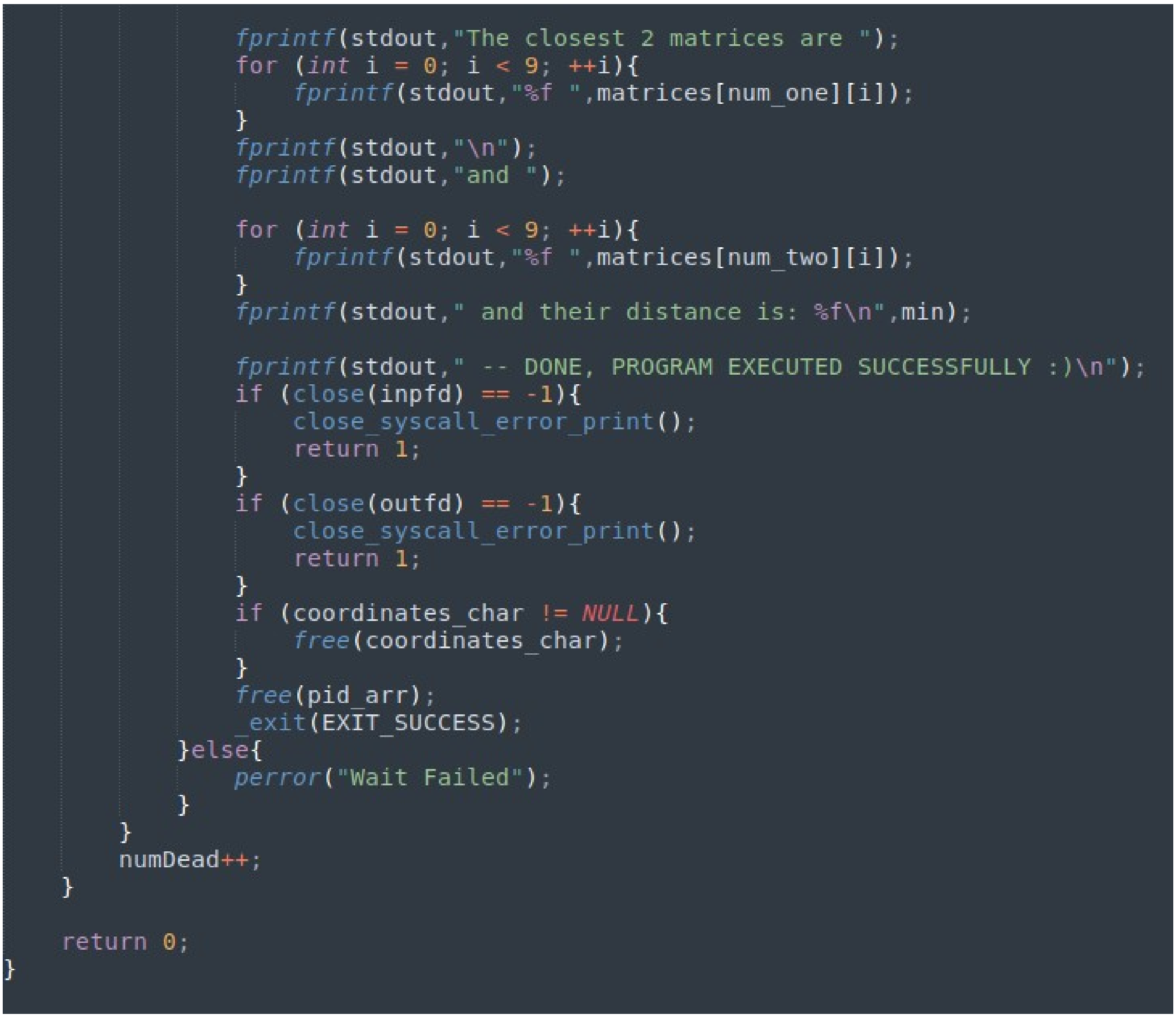
- in the child process we also have the SIGINT signal handler so that we can handle the signal when it arrive from the parent process.

- I open the output_file.dat inside the parent process and send the file descriptor the child process, in this way we can avoid opening the closing the file for each child instead we only need to open and close it once during the lifetime of the program.

-all the mallocs are freed and files are closed in the desired exit conditions.

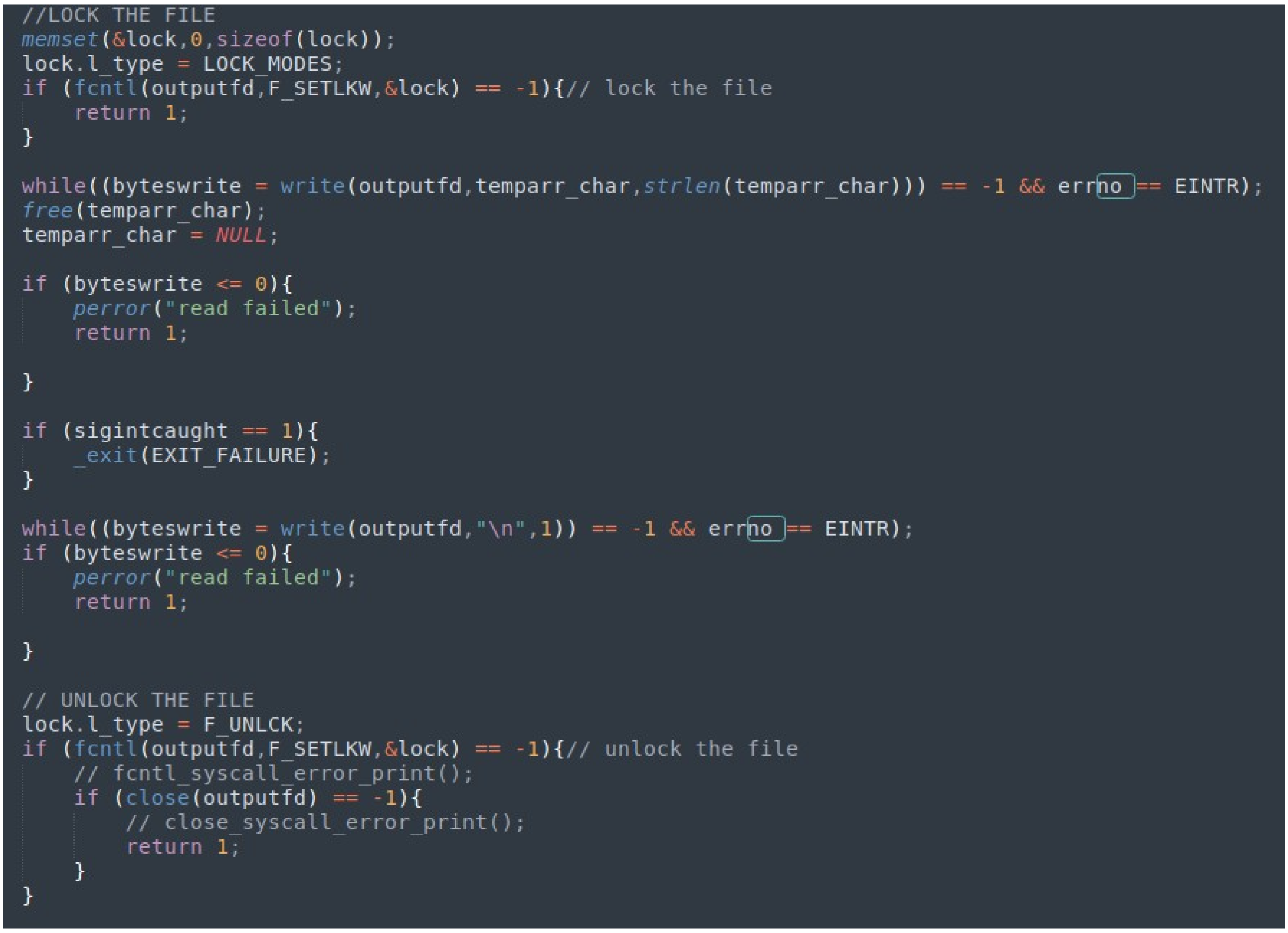


```c
for (;;){

    childPid = wait(NULL);

    if (childPid == -1){

        if (errno == ECHILD){// all child process has returned.  PARENT CODE AREA >>>>>>>>>>>>>>>>

            int output_file_len = 0;
            if (close(outfd) == -1){ // close the file
                close_syscall_error_print();
                return 1;
            }

            outfd = open(output_path,O_RDONLY,mode); // open the file
            if (outfd == -1){// if file not found then print error on sdterr
                open_syscall_error_print();
                perror("Coudln't Open The Output File !!!\n");
                return 1;
            }

            fprintf(stdout,"Reached EOF, collecting outputs from %s\n",output_path);
            for (;;){

                char* temp = (char*) malloc(BUFFER_LIMIT * sizeof(char));
                while((bytesread = read(outfd,temp,BUFFER_LIMIT)) == -1 && errno == EINTR);
                if (bytesread <= 0){
                    free(temp);
                    break;
                }
                output_file_len += strlen(temp);
                free(temp);
            }

            // char parent_arr[output_file_len];
            char* parent_arr = (char*) malloc(output_file_len * sizeof(char)+1);
            char* parent_arr2 = (char*) malloc(output_file_len * sizeof(char)+1);
            if(lseek(outfd,0,SEEK_SET)== -1){
                perror("lseek failed");
                lseek_syscall_error_print();
            }

            for (;;){

                char* temp = (char*) malloc(output_file_len * sizeof(char));
                while((bytesread = read(outfd,temp,output_file_len)) == -1 && errno == EINTR);
                if (bytesread <= 0){  free(temp);break;}
                strcat(parent_arr,temp);
                free(temp);
            }
```



```c
                fprintf(stdout,"The closest 2 matrices are ");
                for (int i = 0; i < 9; ++i){
                    fprintf(stdout,"%f ",matrices[num_one][i]);
                }
                fprintf(stdout,"\n");
                fprintf(stdout,"and ");

                for (int i = 0; i < 9; ++i){
                    fprintf(stdout,"%f ",matrices[num_two][i]);
                }
                fprintf(stdout," and their distance is: %f\n",min);

                fprintf(stdout," -- DONE, PROGRAM EXECUTED SUCCESSFULLY :)\n");
                if (close(inpfd) == -1){
                    close_syscall_error_print();
                    return 1;
                }
                if (close(outfd) == -1){
                    close_syscall_error_print();
                    return 1;
                }
                if (coordinates_char != NULL){
                    free(coordinates_char);
                }
                free(pid_arr);
                _exit(EXIT_SUCCESS);
            }else{
                perror("Wait Failed");
            }
        }
        numDead++;
    }

    return 0;
}
```

- in the child procees’s .c file:
- I set up the signal handler so that it will detect the incoming signal and handle it accordingly.
- I lock the file while write into it in the child process so, when a child process locked the file the other process can’t write into it until the lock has been released.
- when the file if locked I calculate the covariance matrix and write it into the file so that parent can read it.

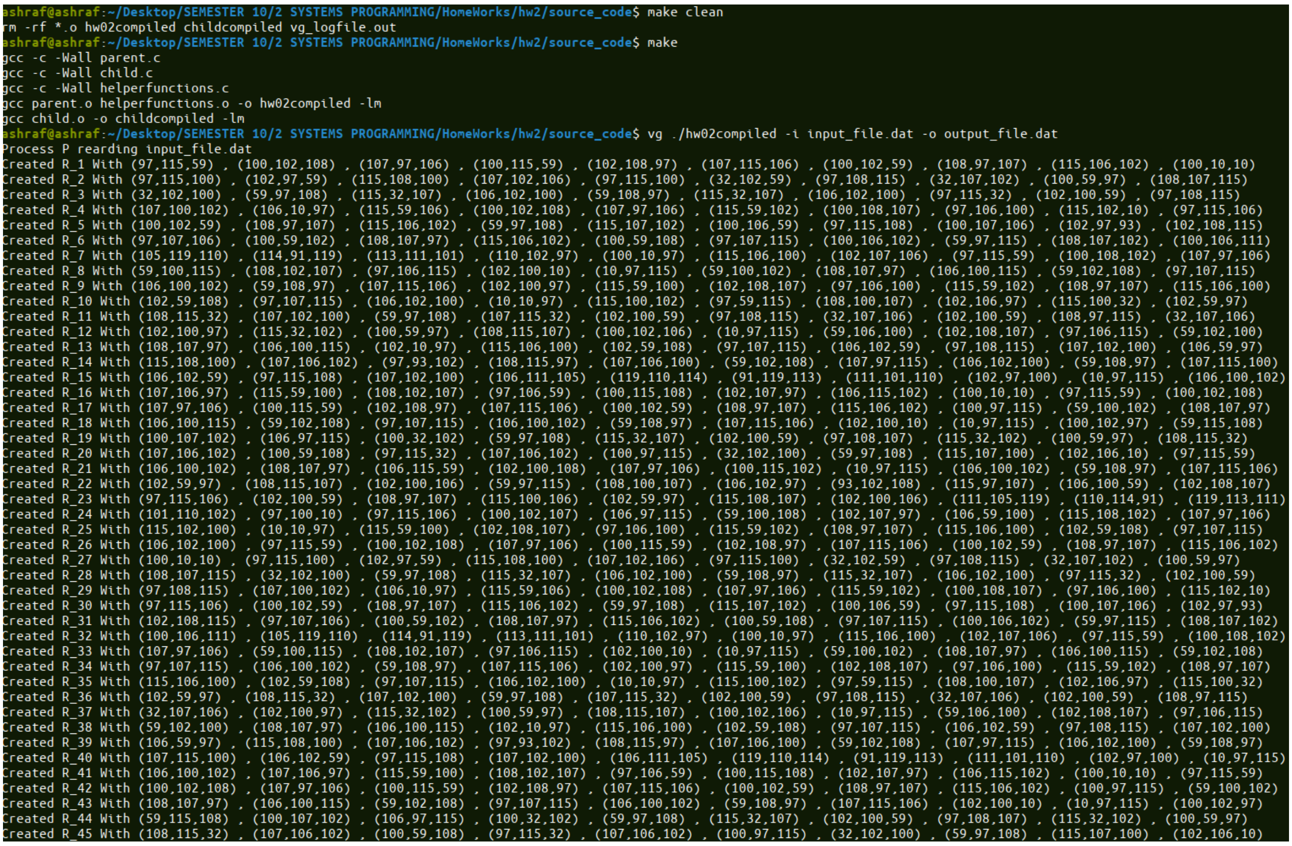


```
//LOCK THE FILE
memset(&lock,0,sizeof(lock));
lock.l_type = LOCK_MODES;
if (fcntl(outputfd,F_SETLKW,&lock) == -1){// lock the file
    return 1;
}

while((byteswrite = write(outputfd,temparr_char,strlen(temparr_char))) == -1 && errno == EINTR);
free(temparr_char);
temparr_char = NULL;

if (byteswrite <= 0){
    perror("read failed");
    return 1;

}

if (sigintcaught == 1){
    _exit(EXIT_FAILURE);
}

while((byteswrite = write(outputfd,"\n",1)) == -1 && errno == EINTR);
if (byteswrite <= 0){
    perror("read failed");
    return 1;

}

// UNLOCK THE FILE
lock.l_type = F_UNLCK;
if (fcntl(outputfd,F_SETLKW,&lock) == -1){// unlock the file
    // fcntl_syscall_error_print();
    if (close(outputfd) == -1){
        // close_syscall_error_print();
        return 1;
    }
}
```

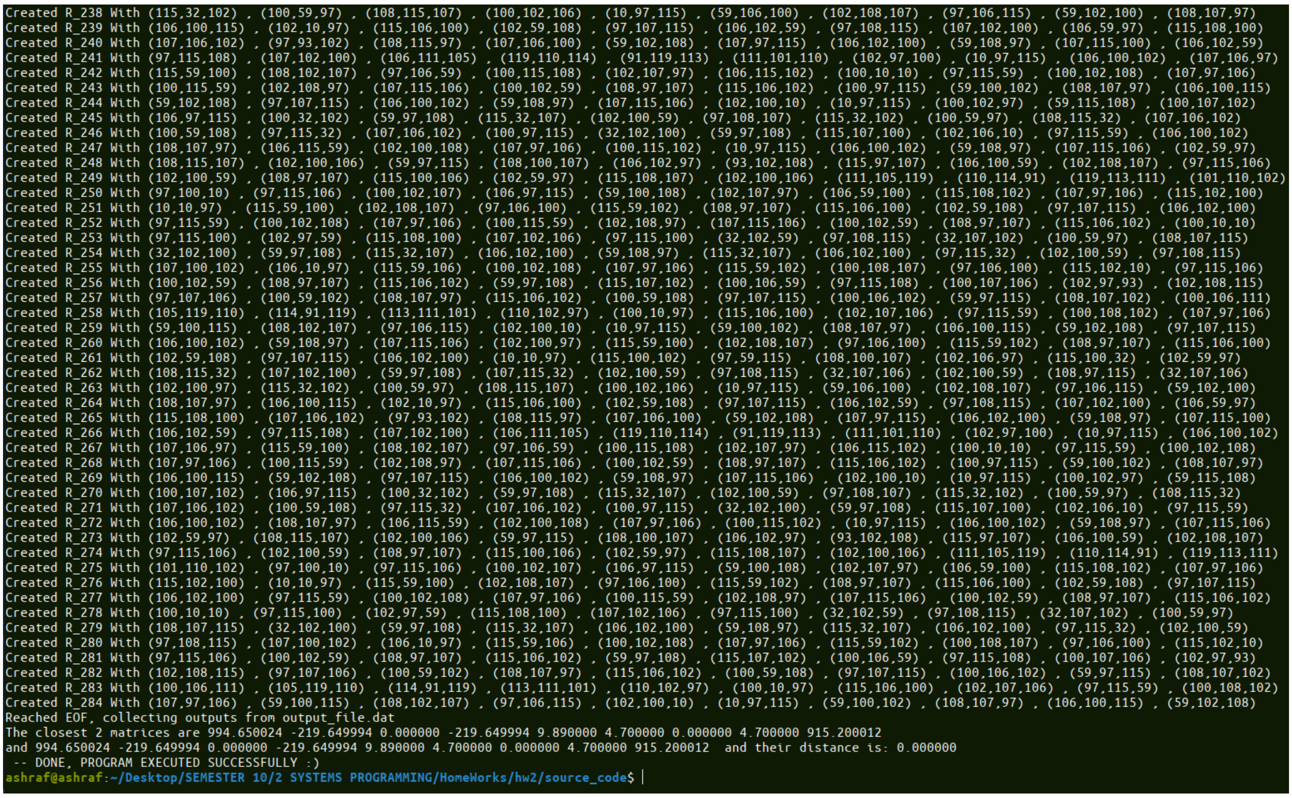
- I printed the outputs using fprintf and put the result int stdout as instruced in the homeworks PDF file.

- I have obtained the system programming rules as much as possible.

SCREEN SHOTS FROM THE PROGRAMS:



```
ashraf@ashraf:~/Desktop/SEMESTER 10/2 SYSTEMS PROGRAMMING/HomeWorks/hw2/source_code$ make clean
rm -rf *.o hw02compiled childcompiled vg_logfile.out
ashraf@ashraf:~/Desktop/SEMESTER 10/2 SYSTEMS PROGRAMMING/HomeWorks/hw2/source_code$ make
gcc -c -Wall parent.c
gcc -c -Wall child.c
gcc -c -Wall helperfunctions.c
gcc parent.o helperfunctions.o -o hw02compiled -lm
gcc child.o -o childcompiled -lm
ashraf@ashraf:~/Desktop/SEMESTER 10/2 SYSTEMS PROGRAMMING/HomeWorks/hw2/source_code$ vg ./hw02compiled -i input_file.dat -o output_file.dat
Process P rearding input_file.dat
Created R_1 With (97,115,59) , (100,102,108) , (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,10,10)
Created R_2 With (97,115,100) , (102,97,59) , (115,108,100) , (107,102,106) , (97,115,100) , (32,102,59) , (97,108,115) , (32,107,102) , (100,59,97) , (108,107,115)
Created R_3 With (32,102,100) , (59,97,108) , (115,32,107) , (106,102,100) , (59,108,97) , (115,32,107) , (106,102,100) , (97,115,32) , (102,100,59) , (97,108,115)
Created R_4 With (107,100,102) , (106,10,97) , (115,59,106) , (100,102,108) , (107,97,106) , (115,59,102) , (100,108,107) , (97,106,100) , (115,102,10) , (97,115,106)
Created R_5 With (100,102,59) , (108,97,107) , (115,106,102) , (59,97,108) , (115,107,102) , (100,106,59) , (97,115,108) , (100,107,106) , (102,97,93) , (102,108,115)
Created R_6 With (97,107,106) , (100,59,102) , (108,107,97) , (115,106,102) , (100,59,108) , (97,107,115) , (100,106,102) , (59,97,115) , (108,107,102) , (100,106,111)
Created R_7 With (105,119,110) , (114,91,119) , (113,111,101) , (110,102,97) , (100,10,97) , (115,106,100) , (102,107,106) , (97,115,59) , (100,108,102) , (107,97,106)
Created R_8 With (59,100,115) , (108,102,107) , (97,106,115) , (102,100,10) , (10,97,115) , (59,100,102) , (108,107,97) , (106,100,115) , (59,102,108) , (97,107,115)
Created R_9 With (106,100,102) , (59,108,97) , (107,115,106) , (102,100,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107) , (115,106,100)
Created R_10 With (102,59,108) , (97,107,115) , (106,102,100) , (10,10,97) , (115,100,102) , (97,59,115) , (108,100,107) , (102,106,97) , (115,100,32) , (102,59,97)
Created R_11 With (108,115,32) , (107,102,100) , (59,97,108) , (107,115,32) , (102,100,59) , (97,108,115) , (32,107,106) , (102,100,59) , (108,97,115) , (32,107,106)
Created R_12 With (102,100,97) , (115,32,102) , (100,59,97) , (108,115,107) , (100,102,106) , (10,97,115) , (59,106,100) , (102,108,107) , (97,106,115) , (59,102,100)
Created R_13 With (108,107,97) , (106,100,115) , (102,10,97) , (115,106,100) , (102,59,108) , (97,107,115) , (106,102,59) , (97,108,115) , (107,102,100) , (106,59,97)
Created R_14 With (115,108,100) , (107,106,102) , (97,93,102) , (108,115,97) , (107,106,100) , (59,102,108) , (107,97,115) , (106,102,100) , (59,108,97) , (107,115,100)
Created R_15 With (106,102,59) , (97,115,108) , (107,102,100) , (106,111,105) , (119,110,114) , (91,119,113) , (111,101,110) , (102,97,100) , (10,97,115) , (106,100,102)
Created R_16 With (107,106,97) , (115,59,100) , (108,102,107) , (97,106,59) , (100,115,108) , (102,107,97) , (106,115,102) , (100,10,10) , (97,115,59) , (100,102,108)
Created R_17 With (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,97,115) , (59,100,102) , (108,107,97)
Created R_18 With (106,100,115) , (59,102,108) , (97,107,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,100,10) , (10,97,115) , (100,102,97) , (59,115,108)
Created R_19 With (100,107,102) , (106,97,115) , (100,32,102) , (59,97,108) , (115,32,107) , (102,100,59) , (97,108,107) , (115,32,102) , (100,59,97) , (108,115,32)
Created R_20 With (107,106,102) , (100,59,108) , (97,115,32) , (107,106,102) , (100,97,115) , (32,102,100) , (59,97,108) , (115,107,100) , (102,106,10) , (97,115,59)
Created R_21 With (106,100,102) , (108,107,97) , (106,115,59) , (102,100,108) , (107,97,106) , (100,115,102) , (10,97,115) , (106,100,102) , (59,108,97) , (107,115,106)
Created R_22 With (102,59,97) , (108,115,107) , (102,100,106) , (59,97,115) , (108,100,107) , (106,102,97) , (93,102,108) , (115,97,107) , (106,100,59) , (102,108,107)
Created R_23 With (97,115,106) , (102,100,59) , (108,97,107) , (115,100,106) , (102,59,97) , (115,108,107) , (102,100,106) , (111,105,119) , (110,114,91) , (119,113,111)
Created R_24 With (101,110,102) , (97,100,10) , (97,115,106) , (100,102,107) , (106,97,115) , (59,100,108) , (102,107,97) , (106,59,100) , (115,108,102) , (107,97,106)
Created R_25 With (115,102,100) , (10,10,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107) , (115,106,100) , (102,59,108) , (97,107,115)
Created R_26 With (106,102,100) , (97,115,59) , (100,102,108) , (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102)
Created R_27 With (100,10,10) , (97,115,100) , (102,97,59) , (115,108,100) , (107,102,106) , (97,115,100) , (32,102,59) , (97,108,115) , (32,107,102) , (100,59,97)
Created R_28 With (108,107,115) , (32,102,100) , (59,97,108) , (115,32,107) , (106,102,100) , (59,108,97) , (115,32,107) , (106,102,100) , (97,115,32) , (102,100,59)
Created R_29 With (97,108,115) , (107,100,102) , (106,10,97) , (115,59,106) , (100,102,108) , (107,97,106) , (115,59,102) , (100,108,107) , (97,106,100) , (115,102,10)
Created R_30 With (97,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (59,97,108) , (115,107,102) , (100,106,59) , (97,115,108) , (100,107,106) , (102,97,93)
Created R_31 With (102,108,115) , (97,107,106) , (100,59,102) , (108,107,97) , (115,106,102) , (100,59,108) , (97,107,115) , (100,106,102) , (59,97,115) , (108,107,102)
Created R_32 With (100,106,111) , (105,119,110) , (114,91,119) , (113,111,101) , (110,102,97) , (100,10,97) , (115,106,100) , (102,107,106) , (97,115,59) , (100,108,102)
Created R_33 With (107,97,106) , (59,100,115) , (108,102,107) , (97,106,115) , (102,100,10) , (10,97,115) , (59,100,102) , (108,107,97) , (106,100,115) , (59,102,108)
Created R_34 With (97,107,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,100,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107)
Created R_35 With (115,106,100) , (102,59,108) , (97,107,115) , (106,102,100) , (10,10,97) , (115,100,102) , (97,59,115) , (108,100,107) , (102,106,97) , (115,100,32)
Created R_36 With (102,59,97) , (108,115,32) , (107,102,100) , (59,97,108) , (107,115,32) , (102,100,59) , (97,108,115) , (32,107,106) , (102,100,59) , (108,97,115)
Created R_37 With (32,107,106) , (102,100,97) , (115,32,102) , (100,59,97) , (108,115,107) , (100,102,106) , (10,97,115) , (59,106,100) , (102,108,107) , (97,106,115)
Created R_38 With (59,102,100) , (108,107,97) , (106,100,115) , (102,10,97) , (115,106,100) , (102,59,108) , (97,107,115) , (106,102,59) , (97,108,115) , (107,102,100)
Created R_39 With (106,59,97) , (115,108,100) , (107,106,102) , (97,93,102) , (108,115,97) , (107,106,100) , (59,102,108) , (107,97,115) , (106,102,100) , (59,108,97)
Created R_40 With (107,115,100) , (106,102,59) , (97,115,108) , (107,102,100) , (106,111,105) , (119,110,114) , (91,119,113) , (111,101,110) , (102,97,100) , (10,97,115)
Created R_41 With (106,100,102) , (107,106,97) , (115,59,100) , (108,102,107) , (97,106,59) , (100,115,108) , (102,107,97) , (106,115,102) , (100,10,10) , (97,115,59)
Created R_42 With (100,102,108) , (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,97,115) , (59,100,102)
Created R_43 With (108,107,97) , (106,100,115) , (59,102,108) , (97,107,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,100,10) , (10,97,115) , (100,102,97)
Created R_44 With (59,115,108) , (100,107,102) , (106,97,115) , (100,32,102) , (59,97,108) , (115,32,107) , (102,100,59) , (97,108,107) , (115,32,102) , (100,59,97)
Created R_45 With (108,115,32) , (107,106,102) , (100,59,108) , (97,115,32) , (107,106,102) , (100,97,115) , (32,102,100) , (59,97,108) , (115,107,100) , (102,106,10)
```



```
Created R_238 With (115,32,102) , (100,59,97) , (108,115,107) , (100,102,106) , (10,97,115) , (59,106,100) , (102,108,107) , (97,106,115) , (59,102,100) , (108,107,97)
Created R_239 With (106,100,115) , (102,10,97) , (115,106,100) , (102,59,108) , (97,107,115) , (106,102,59) , (97,108,115) , (107,102,100) , (106,59,97) , (115,108,100)
Created R_240 With (107,106,102) , (97,93,102) , (108,115,97) , (107,106,100) , (59,102,108) , (107,97,115) , (106,102,100) , (59,108,97) , (107,115,100) , (106,102,59)
Created R_241 With (97,115,108) , (107,102,100) , (106,111,105) , (119,110,114) , (91,119,113) , (111,101,110) , (102,97,100) , (10,97,115) , (106,100,102) , (107,106,97)
Created R_242 With (115,59,100) , (108,102,107) , (97,106,59) , (100,115,108) , (102,107,97) , (106,115,102) , (100,10,10) , (97,115,59) , (100,102,108) , (107,97,106)
Created R_243 With (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,97,115) , (59,100,102) , (108,107,97) , (106,100,115)
Created R_244 With (59,102,108) , (97,107,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,100,10) , (10,97,115) , (100,102,97) , (59,115,108) , (100,107,102)
Created R_245 With (106,97,115) , (100,32,102) , (59,97,108) , (115,32,107) , (102,100,59) , (97,108,107) , (115,32,102) , (100,59,97) , (108,115,32) , (107,106,102)
Created R_246 With (100,59,108) , (97,115,32) , (107,106,102) , (100,97,115) , (32,102,100) , (59,97,108) , (115,107,100) , (102,106,10) , (97,115,59) , (106,100,102)
Created R_247 With (108,107,97) , (106,115,59) , (102,100,108) , (107,97,106) , (100,115,102) , (10,97,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,59,97)
Created R_248 With (108,115,107) , (102,100,106) , (59,97,115) , (108,100,107) , (106,102,97) , (93,102,108) , (115,97,107) , (106,100,59) , (102,108,107) , (97,115,106)
Created R_249 With (102,100,59) , (108,97,107) , (115,100,106) , (102,59,97) , (115,108,107) , (102,100,106) , (111,105,119) , (110,114,91) , (119,113,111) , (101,110,102)
Created R_250 With (97,100,10) , (97,115,106) , (100,102,107) , (106,97,115) , (59,100,108) , (102,107,97) , (106,59,100) , (115,108,102) , (107,97,106) , (115,102,100)
Created R_251 With (10,10,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107) , (115,106,100) , (102,59,108) , (97,107,115) , (106,102,100)
Created R_252 With (97,115,59) , (100,102,108) , (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,10,10)
Created R_253 With (97,115,100) , (102,97,59) , (115,108,100) , (107,102,106) , (97,115,100) , (32,102,59) , (97,108,115) , (32,107,102) , (100,59,97) , (108,107,115)
Created R_254 With (32,102,100) , (59,97,108) , (115,32,107) , (106,102,100) , (59,108,97) , (115,32,107) , (106,102,100) , (97,115,32) , (102,100,59) , (97,108,115)
Created R_255 With (107,100,102) , (106,10,97) , (115,59,106) , (100,102,108) , (107,97,106) , (115,59,102) , (100,108,107) , (97,106,100) , (115,102,10) , (97,115,106)
Created R_256 With (100,102,59) , (108,97,107) , (115,106,102) , (59,97,108) , (115,107,102) , (100,106,59) , (97,115,108) , (100,107,106) , (102,97,93) , (102,108,115)
Created R_257 With (97,107,106) , (100,59,102) , (108,107,97) , (115,106,102) , (100,59,108) , (97,107,115) , (100,106,102) , (59,97,115) , (108,107,102) , (100,106,111)
Created R_258 With (105,119,110) , (114,91,119) , (113,111,101) , (110,102,97) , (100,10,97) , (115,106,100) , (102,107,106) , (97,115,59) , (100,108,102) , (107,97,106)
Created R_259 With (59,100,115) , (108,102,107) , (97,106,115) , (102,100,10) , (10,97,115) , (59,100,102) , (108,107,97) , (106,100,115) , (59,102,108) , (97,107,115)
Created R_260 With (106,100,102) , (59,108,97) , (107,115,106) , (102,100,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107) , (115,106,100)
Created R_261 With (102,59,108) , (97,107,115) , (106,102,100) , (10,10,97) , (115,100,102) , (97,59,115) , (108,100,107) , (102,106,97) , (115,100,32) , (102,59,97)
Created R_262 With (108,115,32) , (107,102,100) , (59,97,108) , (107,115,32) , (102,100,59) , (97,108,115) , (32,107,106) , (102,100,59) , (108,97,115) , (32,107,106)
Created R_263 With (102,100,97) , (115,32,102) , (100,59,97) , (108,115,107) , (100,102,106) , (10,97,115) , (59,106,100) , (102,108,107) , (97,106,115) , (59,102,100)
Created R_264 With (108,107,97) , (106,100,115) , (102,10,97) , (115,106,100) , (102,59,108) , (97,107,115) , (106,102,59) , (97,108,115) , (107,102,100) , (106,59,97)
Created R_265 With (115,108,100) , (107,106,102) , (97,93,102) , (108,115,97) , (107,106,100) , (59,102,108) , (107,97,115) , (106,102,100) , (59,108,97) , (107,115,100)
Created R_266 With (106,102,59) , (97,115,108) , (107,102,100) , (106,111,105) , (119,110,114) , (91,119,113) , (111,101,110) , (102,97,100) , (10,97,115) , (106,100,102)
Created R_267 With (107,106,97) , (115,59,100) , (108,102,107) , (97,106,59) , (100,115,108) , (102,107,97) , (106,115,102) , (100,10,10) , (97,115,59) , (100,102,108)
Created R_268 With (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (100,97,115) , (59,100,102) , (108,107,97)
Created R_269 With (106,100,115) , (59,102,108) , (97,107,115) , (106,100,102) , (59,108,97) , (107,115,106) , (102,100,10) , (10,97,115) , (100,102,97) , (59,115,108)
Created R_270 With (100,107,102) , (106,97,115) , (100,32,102) , (59,97,108) , (115,32,107) , (102,100,59) , (97,108,107) , (115,32,102) , (100,59,97) , (108,115,32)
Created R_271 With (107,106,102) , (100,59,108) , (97,115,32) , (107,106,102) , (100,97,115) , (32,102,100) , (59,97,108) , (115,107,100) , (102,106,10) , (97,115,59)
Created R_272 With (106,100,102) , (108,107,97) , (106,115,59) , (102,100,108) , (107,97,106) , (100,115,102) , (10,97,115) , (106,100,102) , (59,108,97) , (107,115,106)
Created R_273 With (102,59,97) , (108,115,107) , (102,100,106) , (59,97,115) , (108,100,107) , (106,102,97) , (93,102,108) , (115,97,107) , (106,100,59) , (102,108,107)
Created R_274 With (97,115,106) , (102,100,59) , (108,97,107) , (115,100,106) , (102,59,97) , (115,108,107) , (102,100,106) , (111,105,119) , (110,114,91) , (119,113,111)
Created R_275 With (101,110,102) , (97,100,10) , (97,115,106) , (100,102,107) , (106,97,115) , (59,100,108) , (102,107,97) , (106,59,100) , (115,108,102) , (107,97,106)
Created R_276 With (115,102,100) , (10,10,97) , (115,59,100) , (102,108,107) , (97,106,100) , (115,59,102) , (108,97,107) , (115,106,100) , (102,59,108) , (97,107,115)
Created R_277 With (106,102,100) , (97,115,59) , (100,102,108) , (107,97,106) , (100,115,59) , (102,108,97) , (107,115,106) , (100,102,59) , (108,97,107) , (115,106,102)
Created R_278 With (100,10,10) , (97,115,100) , (102,97,59) , (115,108,100) , (107,102,106) , (97,115,100) , (32,102,59) , (97,108,115) , (32,107,102) , (100,59,97)
Created R_279 With (108,107,115) , (32,102,100) , (59,97,108) , (115,32,107) , (106,102,100) , (59,108,97) , (115,32,107) , (106,102,100) , (97,115,32) , (102,100,59)
Created R_280 With (97,108,115) , (107,100,102) , (106,10,97) , (115,59,106) , (100,102,108) , (107,97,106) , (115,59,102) , (100,108,107) , (97,106,100) , (115,102,10)
Created R_281 With (97,115,106) , (100,102,59) , (108,97,107) , (115,106,102) , (59,97,108) , (115,107,102) , (100,106,59) , (97,115,108) , (100,107,106) , (102,97,93)
Created R_282 With (102,108,115) , (97,107,106) , (100,59,102) , (108,107,97) , (115,106,102) , (100,59,108) , (97,107,115) , (100,106,102) , (59,97,115) , (108,107,102)
Created R_283 With (100,106,111) , (105,119,110) , (114,91,119) , (113,111,101) , (110,102,97) , (100,10,97) , (115,106,100) , (102,107,106) , (97,115,59) , (100,108,102)
Created R_284 With (107,97,106) , (59,100,115) , (108,102,107) , (97,106,115) , (102,100,10) , (10,97,115) , (59,100,102) , (108,107,97) , (106,100,115) , (59,102,108)
Reached EOF, collecting outputs from output_file.dat
The closest 2 matrices are 994.650024 -219.649994 0.000000 -219.649994 9.890000 4.700000 0.000000 4.700000 915.200012
and 994.650024 -219.649994 0.000000 -219.649994 9.890000 4.700000 0.000000 4.700000 915.200012  and their distance is: 0.000000
 -- DONE, PROGRAM EXECUTED SUCCESSFULLY :)
ashraf@ashraf:~/Desktop/SEMESTER 10/2 SYSTEMS PROGRAMMING/HomeWorks/hw2/source_code$ |
```