CSE 344 SYSTEMS PROGRAMMING

HOMEWORK 2 REPORT

MURAT BEYAZ

151044004

Problem Definition and Solution

The objective is to have a process read an input file, interpret its contents as coordinates, and forward them for calculations to children processes. Then it will collect their outputs and make a final calculation.

Process P which is the parent process will read a non empty ASCII file provided as command line argument. Every 3 character in ASCII file represents a coordinate in 3 dimensional space. After every 30 character which means 10 coordinates parent process creates a child process with fork+exec paradigm. Output file provided as a command line argument passed to child processes via command line argument. Also the coordinates read from input file will be passed to child processes via environment variables.

```
char* arglist[] = {outputfile, fdstr, NULL};

char* envVec[] = {keysChar[0],keysChar[1],keysChar[2],keysChar[3],keysChar[4],keysChar[5],keysChar[6
int child_pid = spawn("./processC", arglist, envVec);
/****************************************/
```

Command line argument and environment variable set for the child process shown above. Spawn function will fork and then return the pid of the child process to parent process and child process will call execve with the arguments and variables. This operations shown below:

```
int spawn(char* program, char **arglist, char **envVec){

  pid_t child_pid;

  child_pid = fork();
  if(child_pid != 0){
    return child_pid;
  }
  else{
    execve(program, arglist, envVec);
    fprintf(stderr, "An error occured\n");
    abort();
  }
}
```

Process R_i is representing child processes. Child processes takes 10 3d coordinates and calculates covariance matrix for these coordinates with the formula shown below:

$$\mathrm{cov}_{XY} = \frac{\sum (X - M_X)(Y - M_Y)}{n}$$

The resulting covariance matrix is shown below:

$$C = \begin{matrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(z,z) \end{matrix}$$

After these calculations child process will write the matrices to output file in a format that is up to us. I decided to write matrices in a single row in the order of cov(x,x),cov(x,y),cov(x,z),cov(y,x),cov(y,y),cov(y,z),cov(z,x),cov(z,y),cov(z,z).

```
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
676.410,676.410,676.410,676.410,676.410,676.410,676.410,676.410,676.410
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
```

An example of the output file shown above. While child processes must not access the output file simultaneously for writing. So before writing to the file child process must lock the file write the matrix in and then unlock the file.

If process P receives a SIGINT it will forward the signal to all of its children, free all resources, close open files and remove the output file and terminate. I wrote my own handler for this purpose.

```
void handler(int signal_number){
  if(signal_number == SIGINT){
    SIGINT_FLAG = 1;
  }
}
```

```
struct sigaction sa;
memset(&sa, 0, sizeof(sa));
sa.sa_handler = &handler;
sigaction(SIGINT, &sa, NULL);
```

As you can see that the handler defined with sigaction. Because as we saw from the classes this semester signal itself is not safe to use. Sigaction is the recommended solution for this purpose. When the SIGINT signal received SIGINT_FLAG is set to 1 and almost at all places and loops I am checking the is FLAG is 1 or 0. If FLAG is 1 then SIGINT_EXIT function is called which is shown below:

```
void SIGINT_EXIT(){
  int i;
  printf("\nRECEIVED SIGINT EXITING GRACEFULLY\n");
  for(i=0;i<child_count;i++){
    free(coordinates[i]);
  }
  free(coordinates);
  free(norms);
  for(i=0;i<child_count;i++){
    kill(ptr[i], SIGUSR1);
  }
  free(ptr);
  printf("ALLOCATED SPACES ARE FREED\n");
  printf("CHILD PROCESS KILLED\n");
  remove(output_filename);
  printf("Output file deleted\n");
  exit(-1);
}
```

As you can see from the function allocated spaces are freed than send all child processes SIGUSR1 function. Then the array that we allocated space for child process pids via realloc while reading every 10 coordinates freed and exited.

If process P doesn't get a SIGINT signal then it starts to read output file after all children terminated and calculates frobenius norm for each matrix. While calculating frobenius norm the formula that I used shown below:

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

Child processes also have a signal handler to handle the SIGUSR1 that they receive from parent process. This allows to child processes to free allocated spaces and terminate.

Test Cases

Test 1:

This test case tests to see what happens if arguments are missing.

```
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$ ./processP
An argument is missing!
Example Usage Is:  ./test -i inputfilepath -o outputfilepath
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$
```

Program prints the usage and exits.

Test 2:

This test case reads 60 characters and creates 2 child processes they write to the output file and distance calculated via parent process.

Test file's content is shown below:

aaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyaz

ASCII values shown below:

a:97 b:98 c:99 d:100 e:101 f:102 g:103 h:104 i:105 j:106

m:109 u:117 r:114 a:97 t:116 b:98 e:101 y:121 a:97 z:122

with these values 3 dimensional 10 coordinates are shown below:

a:97 b:98 c:99 d:100 e:101 f:102 g:103 h:104 i:105 j:106

X	Y	Z
97	97	97
98	98	98
99	99	99
100	100	100
101	101	101
102	102	102
103	103	103
104	104	104
105	105	105
106	106	106

m:109 u:117 r:114 a:97 t:116 b:98 e:101 y:121 a:97 z:122

X	Y	Z
109	117	114
97	116	98
101	121	97
122	109	117
114	97	116
98	101	121
97	122	109
117	114	97
116	98	101
121	97	122

Calculated and written to output file covariance matrices shown below:

```
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
```

Output of the program shown below:

```
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$ ./processP -i test.dat -o out.dat
Process P reading test.dat
Created R_1 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_2 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Reached EOF, collecting outputs from out.dat
Norm 0 : 24.750000
Norm 1 : 190.552361
The closest 2 matrices are 8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250 and 92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360, their distance is 165.000000
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$
```

Test 3:

This test case shows that if there are more characters in input file but size of it is not enough the program wont create a new process.

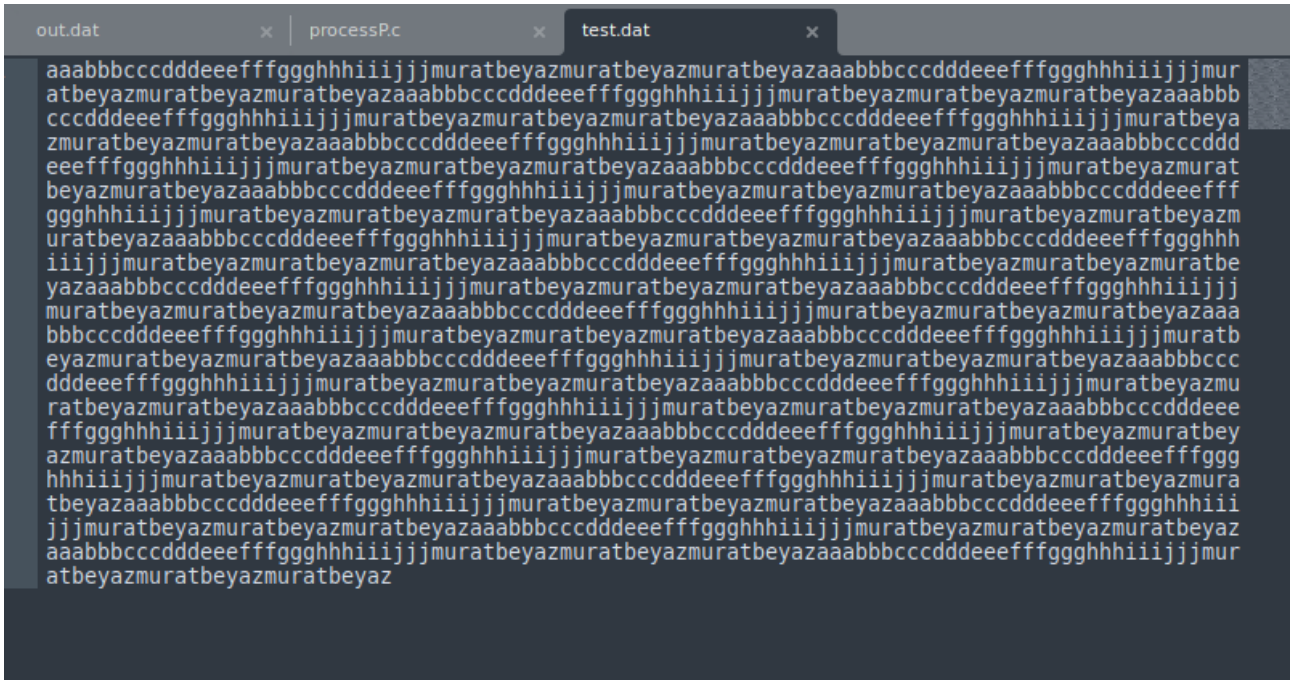
Aaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyaz*****

Input file is shown above this time there are 5 stars at the end of the file and the program will not create a 3rd process for those characters.

```
mrtbyz@mrtbyz: ~/Desktop/System_HW2/FINAL
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$ ./processP -i test.dat -o out.dat
Process P reading test.dat
Created R_1 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_2 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Reached EOF, collecting outputs from out.dat
Norm 0 : 24.750000
Norm 1 : 190.552361
The closest 2 matrices are 8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250 and 92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360, their distance is 165.000000
mrtbyz@mrtbyz:~/Desktop/System_HW2/FINAL$
```

Test 4:

This time testing the program with a big amount of data same text in the test case 1 multiplied 33 times there will be 66 processes and output file will contain 66 rows(matrices). Then frobenius calculation will be covered.


out.dat
processP.c
test.dat
aaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmur
atbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbb
cccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeya
zmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccddd
eeefffgggghhhiiijjjmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmurat
beyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefff
gggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazm
uratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghh
iiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbe
yazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjj
muratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaa
bbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratb
eyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbccc
dddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmu
ratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeee
fffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbey
azmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffggg
hhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmura
tbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiii
jjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyaz
aaabbbcccdddeeefffgggghhhiiijjjmuratbeyazmuratbeyazmuratbeyazaaabbbcccdddeeefffgggghhhiiijjjmur
atbeyazmuratbeyazmuratbeyaz


Created R_16 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_17 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_18 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_19 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_20 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_21 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_22 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_23 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_24 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_25 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_26 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_27 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_28 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_29 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_30 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_31 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_32 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_33 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_34 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_35 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_36 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_37 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_38 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_39 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_40 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_41 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_42 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_43 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_44 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_45 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_46 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_47 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_48 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_49 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_50 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_51 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_52 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_53 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_54 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_55 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_56 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_57 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_58 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_59 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_60 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_61 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_62 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_63 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_64 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Created R_65 with (97,97,97),(98,98,98),(99,99,99),(100,100,100),(101,101,101),(102,102,102),(103,103,103),(104,104,104),(105,105,105),(106,106,106)
Created R_66 with (109,117,114),(97,116,98),(101,121,97),(122,109,117),(114,97,116),(98,101,121),(97,122,109),(117,114,97),(116,98,101),(121,97,122)
Reached EOF, collecting outputs from out.dat

```
Norm 59 : 24.750000
Norm 60 : 190.552361
Norm 61 : 24.750000
Norm 62 : 24.750000
Norm 63 : 190.552361
Norm 64 : 190.552361
Norm 65 : 24.750000
The closest 2 matrices are 8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250 and 8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250, their distance is 0.000000
```

out.dat | processP.c | test.dat

```
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
92.360,-48.940,23.860,-48.940,92.360,-48.940,23.860,-48.940,92.360
8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250,8.250
```