

# Fundamental Exercise on Computer and Information Engineering 1B Assignment 3

XL14608 Thiago Machado da Silva

April 30, 2015

## Question 4

### Subquestion 3

Referring to Figure 1, the left half shows the source code of `prog3-1a` and its output, and the right half shows the source code of `prog3-1b` and its output.

```
1 #include "stdio.h"
2 #include "stdlib.h"
3
4 int main(int argc, char** argv) {
5     FILE *fp;
6     char line[1000];
7     fp=fopen("./prog2-1a.c", "r");
8     if(fp==NULL){
9         printf("Error occurs.");
10        return -1;
11    }
12    while(fgets(line, sizeof(line), fp)!
=NULL){
13        printf("%s", line);
14    }
15    fclose(fp);
16    return 0;
17 }
18
```

fundamental/04 - [master•] » ./prog3-1a

```
1 #include "stdio.h"
2 #include "stdlib.h"
3
4 int main(int argc, char** argv) {
5     FILE *fp;
6     char line[1000];
7     fp=fopen("./prog2-1a.c", "r");
8     if(fp==NULL){
9         printf("Error occurs.");
10        return -1;
11    }
12    int lineNumber = 1;
13    while(fgets(line, sizeof(line), fp)!
=NULL){
14        printf("%d:%s", lineNumber++, line
);
15    }
16    fclose(fp);
17    return 0;
18 }
19
```

fundamental/04 - [master•] » ./prog3-1b

```
1 #include "stdio.h"
2 #include "stdlib.h"
3
4 int main(int argc, char** argv) {
5     FILE *fp;
6     char line[1000];
7     fp=fopen("./prog2-1a.c", "r");
8     if(fp==NULL){
9         printf("Error occurs.");
10        return -1;
11    }
12    int lineNumber = 1;
13    while(fgets(line, sizeof(line), fp)!
=NULL){
14        printf("%d:%s", lineNumber++, line
);
15    }
16    fclose(fp);
17    return 0;
18 }
19
```

NORMAL > PASTE > prog3-1a.c

NORMAL > PASTE > prog3-1b.c[+]

ValueError: ...ompletions yet.

Figure 1: from left to right: `prog3-1a.c` and its output, `prog3-1b.c` and its output.

Referring to Figure 2, the left half shows the source code of `prog3-1c` and the right half shows the `prog2-1a-line.txt` file.

### Subquestion 4

Referring to Figure 3, the left half shows the source code of `prog3-2a.c` (at the top) and its output (at the bottom). The right half shows the important part

```

1 #include "stdio.h"
2 #include "stdlib.h"
3
4 int main(int argc, char** argv) {
5     FILE *fpr, *fpw;
6     char line[1000];
7     fpr=fopen("./prog2-1a.c", "r");
8     fpw=fopen("./prog2-1a-line.txt", "w");
9     if(fpr==NULL || fpw==NULL){
10         printf("Error occurs.");
11         return -1;
12     }
13     int lineNumber = 1;
14     while(fgets(line, sizeof(line), fpr)!=NULL){
15         fprintf(fpw, "%d:%s", lineNumber++, line);
16     }
17     fclose(fpr);
18     fclose(fpw);
19     return 0;
20 }

```

```

|fundamental/04 - [master•] » less prog2-1a-line.txt
|1:#include "stdio.h"
|2:#include "stdlib.h"
|3:
|4:int main(int argc, char** argv) {
|5:
|6:     int max = 5000, iter = 1000;
|7:     int *array1 =
|8:         (int*) malloc(sizeof(int) * max);
|9:     int *array2 =
|10:         (int*) malloc(sizeof(int) * max);
|11:
|12:     // create arrays
|13:     for (int i = 0; i < max; i++) {
|14:         array1[i] = 2 * i + 1;
|15:         array2[i] = 2 * i;
|16:     }
|17:
|18:     // exchange arrays elements
|19:     for (int j = 0; j < iter; j++) {
|20:         for (int i = 0; i < max; i++) {

```

Figure 2: from left to right: prog3-1c.c and prog2-1a-line.txt.

of the source code of prog3-2b.c (at the top) and it's output (at the bottom). The hidden part of prog3-2b.c is the same as prog3-2b.c. In the prog3-2b program, I bubble sorted the array using when each new structure were added to the array.

```

1 #include "stdio.h"
2 #include "stdlib.h"
3 #include "math.h"
4 struct _pref {
5     char name[50];
6     int population;
7 };
8 typedef struct _pref PREF;
9 int main(int argc, char** argv) {
10     PREF pref[47];
11     FILE *fpr;
12     char line[1000];
13     fpr=fopen("./population.csv", "r");
14     if(fpr==NULL){
15         printf("Error occurs.");
16         return -1;
17     }
18     int prefNumber = 0, max;
19     double mean = 0.0, variance = 0.0;
20     while(fscanf(fpr, "%s %d", pref[prefNumber].name, &pref[prefNumber].population)
21 )!=EOF) {
22     mean += pref[prefNumber].population;
23     prefNumber++;
24 }
25 max = prefNumber;
26 mean /= max;
27 while(prefNumber >= 0) {
28     variance += sqrt(abs(pref[prefNumber].population - mean));
29     prefNumber--;
30 }
31 variance = sqrt(variance);
32 printf("mean: %e, variance: %e", mean, variance);
33 fclose(fpr);
34 return 0;
35 }

```

```

16     return -1;
17 }
18 int prefNumber = 0;
19 while(fscanf(fpr, "%s %d", pref[prefNumber].name, &pref[prefNumber].population)
20 )!=EOF) {
21     for (int i = prefNumber; i > 0; i--) {
22         if (pref[i].population > pref[i - 1].population) {
23             PREF temp = pref[i];
24             pref[i] = pref[i - 1];
25             pref[i - 1] = temp;
26         }
27     }
28     prefNumber++;
29 }
30 for (int i = 0; i < 20; i++) {
31     printf("%s %d\n", pref[i].name, pref[i].population);
32 }
33 fclose(fpr);
34 return 0;
35 }

```

```

NORMAL > PASTE master > prog3-2b.c c < 55% : 19: 1 ! trailing[18]
ValueError: Still no compile flags, no completions yet.

```

```

fundamental/04 - [master] » ./prog3-2b
|Tokyo-to 12577
|Osaka-fu 8817
|Kanagawa 8792
|Aichi 7255
|Saitama 7054
|Chiba 6056
|Hokkaido 5628
|Hyogo 5591
|Fukuoka 5050
|Shizuoka 3792
|Ibaraki 2975
|Hiroshima 2877
|Kyoto-fu 2648
|Niigata 2431
|Miyagi 2360
|Nagano 2196
|Gifu 2107
|Fukushima 2091
|Gumma 2024
|Tochigi 2017
fundamental/04 - [master] »

```

```

fundamental/04 - [master] » ./prog3-2a
mean: 2.718489e+03, variance: 4.376838e+01
fundamental/04 - [master] »

```

Figure 3: from left to right: prog3-2a.c (top) and it's output (bottom), prog3-2b.c (top) and it's output (bottom).