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 it's output, and the right half shows the source code of `prog3-1b` and it's 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
~
NORMAL > PASTE >> prog3-1a.c                <<
```

```
fundamental/04 - [master•] » ./prog3-1a
#include "stdio.h"
#include "stdlib.h"

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

  int max = 5000, iter = 1000;
  int *array1 =
    (int*) malloc(sizeof(int) * max);
  int *array2 =
    (int*) malloc(sizeof(int) * max);

  // create arrays
  for (int i = 0; i < max; i++) {
    array1[i] = 2 * i + 1;
    array2[i] = 2 * i;
  }

  // exchange arrays elements
  for (int j = 0; j < iter; j++) {
    for (int i = 0; i < max; i++) {
      int k = array1[i];
```

```
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 }
NORMAL > PASTE >> prog3-1b.c[+]             <<
ValueError: ...ompletions yet.
```

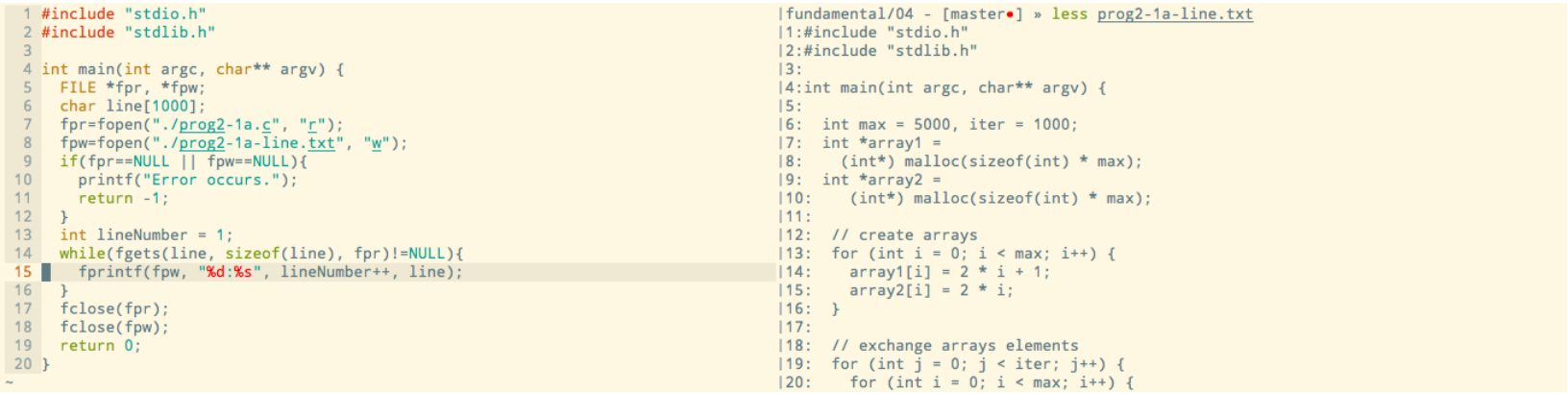
```
|~ » cd sit/fundamental/04
|fundamental/04 - [master•] » ./prog3-1b
|1:#include "stdio.h"
|2:#include "stdlib.h"
|3:
|4:int main(int argc, char** argv) {
|5:
|6:  int max = 5000, iter = 1000;
|7:  int *array1 =
|8:    (int*) malloc(sizeof(int) * max);
|9:  int *array2 =
|10:   (int*) malloc(sizeof(int) * 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 1: from left to right: `prog3-1a.c` and it's output, `prog3-1b.c` and it's 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 it's output (at the bottom). The right half shows the important part



```
#include "stdio.h"
#include "stdlib.h"

int main(int argc, char** argv) {
  FILE *fpr, *fpw;
  char line[1000];
  fpr=fopen("./prog2-1a.c", "r");
  fpw=fopen("./prog2-1a-line.txt", "w");
  if(fpr==NULL || fpw==NULL){
    printf("Error occurs.");
    return -1;
  }
  int lineNumber = 1;
  while(fgets(line, sizeof(line), fpr)!=NULL){
    fprintf(fpw, "%d:%s", lineNumber++, line);
  }
  fclose(fpr);
  fclose(fpw);
  return 0;
}
```

```
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.

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