Folder roteiro-10/include

3 printable files

(file list disabled)

roteiro-10/include/sort-module.h

```h
#ifndef SORT_MODULE_H
#define SORT_MODULE_H

#include <stdio.h>
#include <stdlib.h>

#include "./time.h"

/*
#define NAME_LENGTH 50

typedef struct Person {
    char name[NAME_LENGTH + 1];
    int age;
} Key;
*/

typedef int Key;

typedef struct {
    long long comps, swaps;
    Time start, end;
} Statistics;

void getElements(Key* arr, int quantity);
int compare(Key a, Key b, int reverse, Statistics* statistics);
void swap(Key* a, Key* b, Statistics* statistics);
void print(Key* arr, int n);

#endif
```

roteiro-10/include/sort.h

```h
#ifndef SORT_H
#define SORT_H

#include "./sort-module.h"

// Sorting essential functions
Statistics* createStatistics();
void destroyStatistics(Statistics** s);
void selectionSort(Key* arr, int n, int reverse, Statistics* statistics);
void insertionSort(Key* arr, int n, int reverse, Statistics* statistics);
void bubbleSort(Key* arr, int n, int reverse, Statistics* statistics);
void getOptions(void (**sortFunction)(Key *, int, int, Statistics *), int *order, int *quantity);

#endif
```

roteiro-10/include/time.h

```h
#ifndef TEMPO_H
#define TEMPO_H

#include <stdio.h>
#include <stdlib.h>

typedef long double Time;

Time getCpuTime();
Time formatTime(long int sec, long int usec);
void printElapsedTime(Time start, Time end);

#endif
```

Folder roteiro-10/src

4 printable files

(file list disabled)

roteiro-10/src/main.c

```c
#include <stdio.h>
#include <stdlib.h>

#include "../include/sort-module.h"
#include "../include/sort.h"
#include "../include/time.h"

int main() {
    void (*sortFunction)(Key *, int, int, Statistics *);
    int order, quantity;
    getOptions(&sortFunction, &order, &quantity);

    Key *arr = (Key *)malloc(quantity * sizeof(Key));
    getElements(arr, quantity);

    Statistics *s = createStatistics();
    s->start = getCpuTime();
    (*sortFunction)(arr, quantity, order, s);
    s->end = getCpuTime();

    print(arr, quantity);

    printf("\n== Statistics ==\n");
    printf("Elapsed time = ");
    printElapsedTime(s->start, s->end);
    printf("\nComparations = %Ld\n", s->comps);
    printf("Swaps = %Ld\n", s->swaps);

    destroyStatistics(&s);
    free(arr);

    return 0;
}
```

roteiro-10/src/sort-module.c

```c
#include "./sort-module.h"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Get elements from user input
void getElements(Key* arr, int quantity) {
    for (int i = 0; i < quantity; i++) {
        scanf("%d", &arr[i]);
    }
}

// Compare elements (a < b)
int compare(Key a, Key b, int reverse, Statistics* statistics)
{
    if (statistics != NULL) statistics->comps++;
    return (!reverse ? (a < b) : (a > b));
}

// Swap elements
void swap(Key* a, Key* b, Statistics* statistics) {
    if (statistics != NULL) statistics->swaps++;
    Key aux = *a;
    *a = *b;
    *b = aux;
}

// Print all elements in the array
void print(Key* arr, int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

/*
// Get elements from user input
void getElements(Key *arr, int quantity) {
    char name[NAME_LENGTH + 1];
    for (int i = 0; i < quantity; i++) {
```

```
41          printf("Name (max length %d): ", NAME_LENGTH);
42          setbuf(stdin, NULL);
43          fgets(name, NAME_LENGTH, stdin);
44          if (name[strlen(name) - 1] == '\n') name[strlen(name) -
    1] = '\0';
45          strcpy(arr[i].name, name);
46
47          printf("Age: ");
48          scanf("%d", &arr[i].age);
49
50          printf("\n");
51      }
52  }
53
54  // Compare elements (a.name < b.name) and (a.age < b.age)
55  int compare(Key a, Key b, int reverse, Statistics* statistics)
    {
56      if (statistics != NULL) statistics->comps++;
57
58      int cmp = strcmp(a.name, b.name);
59      if (cmp == 0)
60          return (!reverse ? (a.age < b.age) : (a.age > b.age));
61      return (!reverse ? (cmp < 0) : (cmp > 0));
62  }
63
64  // Swap elements
65  void swap(Key* a, Key* b, Statistics* statistics) {
66      if (statistics != NULL) statistics->swaps++;
67
68      Key aux = *a;
69      *a = *b;
70      *b = aux;
71  }
72
73  // Print all elements in the array
74  void print(Key* arr, int n) {
75      for (int i = 0; i < n; i++) {
76          printf("%s - %d\n", arr[i].name, arr[i].age);
77      }
78      printf("\n");
79  }
80  */
81
```

roteiro-10/src/sort.c

```c
#include "../include/sort.h"

#include <stdio.h>
#include <stdlib.h>

// Allocate a new statistics pointer
Statistics* createStatistics() {
    Statistics* new = (Statistics*)malloc(sizeof(Statistics));
    new->comps = 0;
    new->swaps = 0;
    new->start = 0;
    new->end = 0;
    return new;
}

// Destroy statistics allocated pointer
void destroyStatistics(Statistics** s) {
    if (*s == NULL) return;
    free(*s);
    *s = NULL;
}

// Selection Sort Function
void selectionSort(Key* arr, int n, int reverse, Statistics* statistics) {
    for (int i = 0; i < n; i++) {
        int min = i;
        for (int j = i + 1; j < n; j++) {
            if (compare(arr[j], arr[min], reverse, statistics)) min = j;
        }
        if (compare(arr[min], arr[i], reverse, statistics)) swap(&arr[min], &arr[i], statistics);
    }
}

// Insertion Sort Function
void insertionSort(Key* arr, int n, int reverse, Statistics* statistics) {
    for (int i = 0; i < n; i++) {
        for (int j = i; j >= 1; j--) {
```

```
38              if (compare(arr[j], arr[j - 1], reverse,
    statistics))
39                  swap(&arr[j], &arr[j - 1], statistics);
40              else
41                  break;
42          }
43      }
44  }
45
46  // Bubble Sort Function
47  void bubbleSort(Key* arr, int n, int reverse, Statistics*
    statistics) {
48      for (int i = 0; i < n; i++) {
49          for (int j = 0; j < n - i - 1; j++) {
50              if (compare(arr[j + 1], arr[j], reverse,
    statistics)) {
51                  swap(&arr[j + 1], &arr[j], statistics);
52              }
53          }
54      }
55  }
56
57  // Function to exit the program if there is an error
58  void throwError(char* message) {
59      printf("%s - Aborting...\n", message);
60      exit(1);
61  }
62
63  // User select the sorting options
64  void getOptions(void (**sortFunction)(Key*, int, int,
    Statistics*), int* order, int* quantity) {
65      int option;
66
67      // User selects the algorithm
68      printf("1 - Selection Sort\n");
69      printf("2 - Insertion Sort\n");
70      printf("3 - Bubble Sort\n");
71      printf("Sorting Algorithm: ");
72      scanf("%d", &option);
73      switch (option) {
74          case 1:
75              *sortFunction = &selectionSort;
76              break;
77          case 2:
78              *sortFunction = &insertionSort;
```

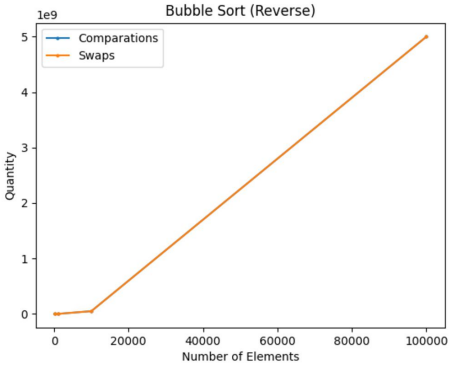
```
            break;
        case 3:
            *sortFunction = &bubbleSort;
            break;
        default:
            throwError("Invalid algorithm option");
    }

    printf("\n");

    // User selects the order
    printf("0 - Ascending\n");
    printf("1 - Descending\n");
    printf("Order: ");
    scanf("%d", &option);
    if (option != 0 && option != 1)
        throwError("Invalid order option");

    *order = option;

    printf("\n");

    // User enters the quantity
    printf("Number of Elements: ");
    scanf("%d", quantity);
    if (quantity < 0)
        throwError("Invalid number");

    printf("\n");
}

```

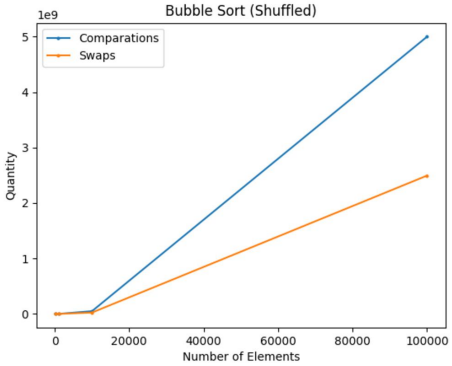
roteiro-10/src/time.c

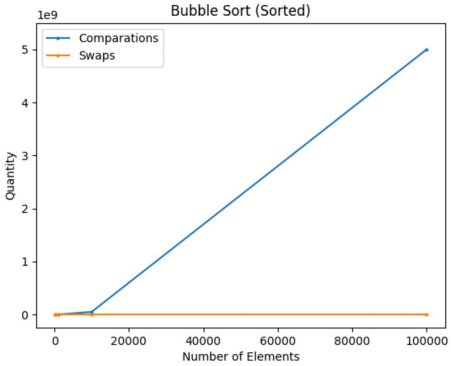
```
#include "../include/time.h"

#include <sys/resource.h>

// Returns CPU time in that moment
Time getCpuTime() {
    struct rusage usage;
    getrusage(RUSAGE_SELF, &usage);
```

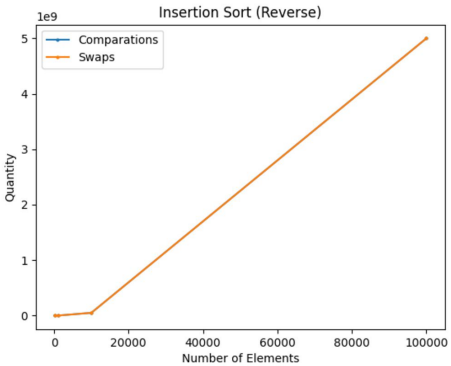
```
    return formatTime(usage.ru_utime.tv_sec, usage.ru_utime.tv_usec);
}

// Join time in seconds and microseconds into a long double variable
Time formatTime(long int sec, long int usec) {
    Time totalTime = sec + ((Time)usec / 1000000.0L);
    return totalTime;
}

// Prints the difference between start to end
void printElapsedTime(Time start, Time end) {
    Time elapsedTime = end - start;
    printf("%Lf", elapsedTime);
}

```

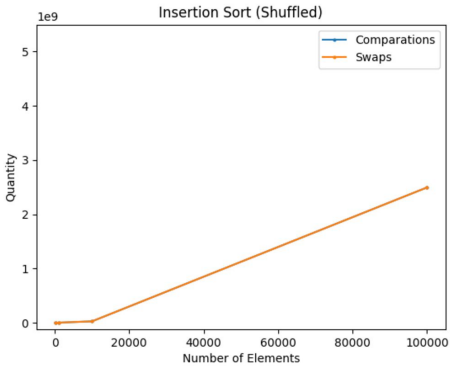
```
gabriel-dp@gabriel-dp: ~/Desktop/dev/c/lab2/roteiro-10
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$ ./bin/out
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
Sorting Algorithm: 1

0 - Ascending
1 - Descending
Order: 0

Number of Elements: 5

3 1 2 5 4
1 2 3 4 5

Statistics
Elapsed time = 0.000002
Comparations = 15
Swaps = 3
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$ ./bin/out
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
Sorting Algorithm: 2

0 - Ascending
1 - Descending
Order: 1

Number of Elements: 5

1 2 3 4 5
5 4 3 2 1

Statistics
Elapsed time = 0.000003
Comparations = 10
Swaps = 10
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$ ./bin/out
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
Sorting Algorithm: 3

0 - Ascending
1 - Descending
Order: 0

Number of Elements: 5

2 5 1 3 4
1 2 3 4 5

Statistics
Elapsed time = 0.000000
Comparations = 10
Swaps = 4
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$
```

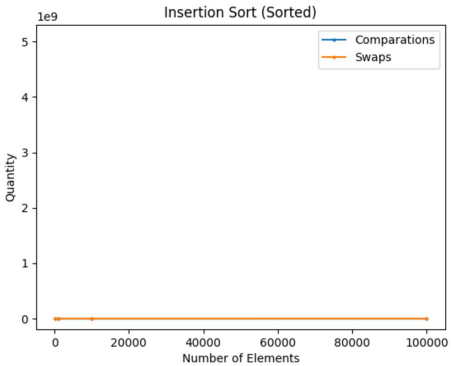

Bubble Sort (Reverse)
1e9
Comparations
Swaps
Quantity
Number of Elements

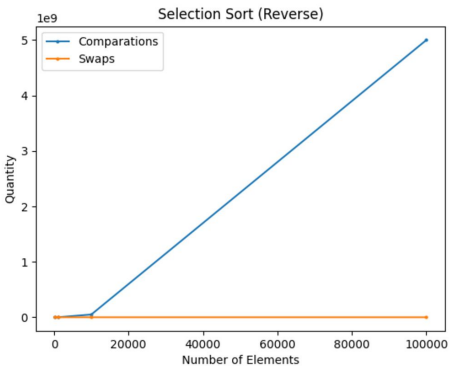

Bubble Sort (Shuffled)
1e9
Comparations
Swaps
Quantity
Number of Elements

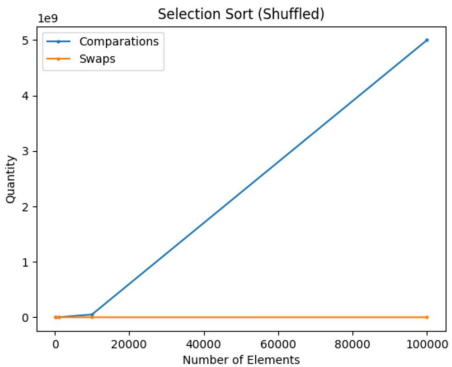

Bubble Sort (Sorted)
1e9
Comparations
Swaps
Quantity
Number of Elements

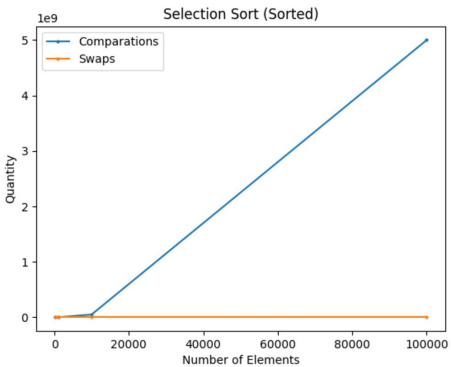

Insertion Sort (Reverse)
1e9
Comparations
Swaps
Quantity
Number of Elements

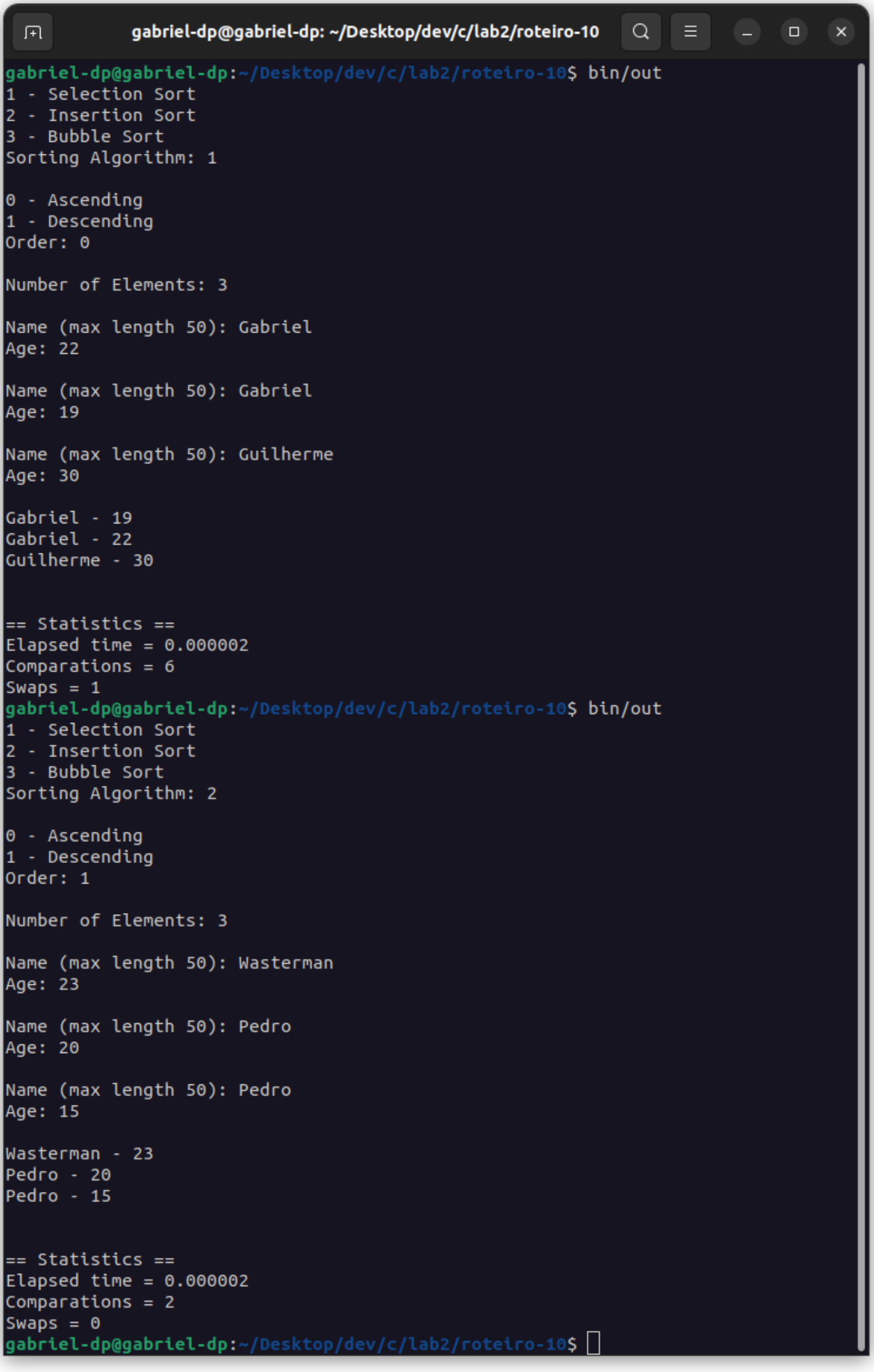

Insertion Sort (Shuffled)
1e9
Comparations
Swaps
Quantity
Number of Elements


Insertion Sort (Sorted)
1e9
Comparations
Swaps
Quantity
Number of Elements


Selection Sort (Reverse)
1e9
Comparations
Swaps
Quantity
Number of Elements


Selection Sort (Shuffled)
1e9
Comparations
Swaps
Quantity
Number of Elements


Selection Sort (Sorted)
1e9
Comparations
Swaps
Quantity
Number of Elements


gabriel-dp@gabriel-dp: ~/Desktop/dev/c/lab2/roteiro-10
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$ bin/out
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
Sorting Algorithm: 1
0 - Ascending
1 - Descending
Order: 0
Number of Elements: 3
Name (max length 50): Gabriel
Age: 22
Name (max length 50): Gabriel
Age: 19
Name (max length 50): Guilherme
Age: 30
Gabriel - 19
Gabriel - 22
Guilherme - 30
== Statistics ==
Elapsed time = 0.000002
Comparations = 6
Swaps = 1
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$ bin/out
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
Sorting Algorithm: 2
0 - Ascending
1 - Descending
Order: 1
Number of Elements: 3
Name (max length 50): Wasterman
Age: 23
Name (max length 50): Pedro
Age: 20
Name (max length 50): Pedro
Age: 15
Wasterman - 23
Pedro - 20
Pedro - 15
== Statistics ==
Elapsed time = 0.000002
Comparations = 2
Swaps = 0
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-10$