MSC - INF101B
C Programming Language
Travaux Pratiques (TP) 3

Panagiotis PAPADAKIS

1 Introduction

The practical session TP 3 proposes coding exercises that will allow you to strengthen your understanding of the notions presented during Course 3. In the beginning of each TP, you should create a separate folder that will contain your work. For the third TP 3, create a folder named TP3 from a terminal and go into this directory.

2 Exercises

2.1

Using a text editor, create a text file named "function_data" with the following contents:

2.700000 2.543769
2.740000 2.845237
2.780000 3.131921
2.820000 3.392771
2.860000 3.617007
2.900000 3.794857
2.940000 3.918256
2.980000 3.981452
3.020000 3.981452
3.060000 3.918256
3.100000 3.794856
3.140000 3.617005
3.180000 3.392769
3.220000 3.131921
3.260000 2.845236

Considering that the 2 columns correspond to the x and y coordinates respectively of the samples of a function, try to visualize / plot these samples by typing the following command:

```
gnuplot --persist -e 'plot "function_data" '
```

Write a program that can plot data samples by calling the `gnuplot` tool from within the code (you should use the function `system()`). In particular, your program should receive as argument from the command line the mean μ, standard deviation σ and the number of samples of a 1D Gaussian distribution function and should plot samples taken uniformely within the $[-3\sigma, +3\sigma]$ range (hint: to use standard math functions you need to include `math.h` and also provide the '-lm'gcc option to link with the math library). As an example, the output of your program after running the command:

```
./a.out 3 0.6 20
```

should be like the following figure:

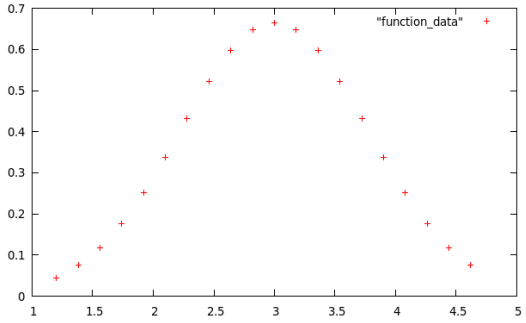

Figure 1: Example output of your program.

2.2

Consider a binary file that contains a sequence of (float) x coordinates followed by the corresponding sequence of (float) y coordinates of a set of points sampled from a mathematical function. Write a C function that reads such a binary file and produces a text file in the same format as the one used earlier by `gnuplot`.

Use this function in a new program which plots the data provided in the following file:

`https://formations.telecom-bretagne.eu/fad/pluginfile.php/66429/mod_folder/content/0/TP3/function_data_bin?forcedownload=1`

2.3 Exercise 3

2.3.1

By using the following code fraction:

```
#define MAX_FNAME_LENGTH 256

struct filestructure
{
char filename[MAX_FNAME_LENGTH];
unsigned int num_of_bytes;
};

typedef struct filestructure filestruct;
```

create a program that can *pack* and *unpack* file collections. You should create two respective functions, namely:

```
void file_packing(char *packed_file_name, int num_of_files, char
**array_of_names);

void file_unpacking(char *packed_file_name);
```

that can be called from your program depending on whether we want to pack a collection of files, or unpack a packed file to obtain its contained files. The string `packed_file_name` denotes the name of the file to be packed / unpacked. The `num_of_files` is the number of files that will be packed and the array of strings `array_of_names` holds each individual file name. The files to be packed can be of arbitrary type and should be provided to your program by the command line.

NOTE-1: You will need to use standard file I/O functions functions as well as functions such as `strlen, strcpy` etc. NOTE-2: To prevent overwriting already existing files after unpacking, make your program create a new folder where it should put the unpacked files (e.g. using function `system()`).

2.3.2

Create a source file named "packing_yourLASTNAME.c" and the corresponding header file "packing_yourLASTNAME.h", that will include the definitions and declarations of both functions respectively. Compile the "packing_yourLASTNAME.c" to obtain the object file ".o". Send the header and the object file to another team and ask the same team to send you their own. Include their header file and link their object file to your program and test that they work as intended. Try different number and types of files.