C - Pool - Tek1

Subject Assignment 2 - Display Screen

C Pool Managers
looneytunes@epitech.eu

Contents

Instructions

- The team leader (first login of the line) has to sign up his group for the defense.
- Any request for precisions on a subject will complicate it.
- It is forbidden to modify the sources of your project after 10 AM Sunday.
- The subject may change until one hour before turn-in.
- The assignment exercises are to be carried out by groups of 2.
- Only the team leader's turn-in directory will be collected.
- You will find the list of the imposed groups and your assigned subject in files group_promotion_city.txt.
- You will have to carry out the indicated subject with your imposed partners and to present yourselves at your defense Sunday, at the right time, with all your partners.
- For the defense, the project should be finished. Defenses are used to present and explain your work in the slightest detail.
- Every member of the group should be fully aware of the achieved work. Each member will be questioned, the mark of the group is based on the worst explanations.
- Obviously, you will have to do everything possible to contact your partners: Look at their intranet profile, facebook, etc... No excuse will be accepted in regard to group problems.
- If after have tried everything one of your partners is still unreachable, send an email to your DPR (Regional Education Director) as soon as possible.
- You can optionally carry out several subjects to get a potential bonus.

> ⚠ It is absolutely mandatory to have the mandatory subject perfectly carried out to claim the bonus subjects.

- Respect the norm takes time, but is good for you. This way your code will respect the norm since the first written line.
- We have been very indulgent for defenses of the first assignment, we will be much more rigorous this time.

> ⚠ A Segfault, Bus Error, Floating Exception or equivalent is eliminatory!

Turn-in details

- Turn-in directory:
  `Piscine_C_colles-Semaine_02`

> *Hints* `Remember it is always better to create your repository at the beginning of the day and to turn-in your work on a regular basis`

> *Hints* `Only the project turned in by your project leader will be picked up`

- Binary name:
  `a.out`
- Your source code will be compiled with the following command:
  `cc *.c`
- Technical indications for the display on the terminal:
  - To make the cursor return at the beginning of the line, use :

    ```
    my_putchar(’\r’);
    ```

  - To make the cursor come back one character, use :

    ```
    my_putchar(’\b’);
    ```

  - To delay the display, `man usleep`.
- Good luck...

Indications for all subjects

- Turn-in directory:
  `Piscine_C_colles-Semaine_02`

> 💡 *Hints* `Remember it is always better to create your repository at the beginning of the day and to turn-in your work on a regular basis`

- Technical indications for the display on the terminal:
  - To make the cursor return at the beginning of the line, use :

    ```
        my_putchar('\r');
    ```

  - To make the cursor come back one character, use :

    ```
        my_putchar('\b');
    ```

  - To delay the display, `man usleep`.

- Good luck...

Subject 01

- Binary name:
  `colle2-1`
- Your source code will be compiled with the following command:
  `cc *.c -o colle2-1`
- You have to develop a program that takes in parameters n character strings and that displays them alternately, making them appear in two manners:
  - The characters come one by one from the left.
  - The characters come one by one from the right.
- When the program reaches the last character string, it restarts with the first one, so it will never stop.
- An optional parameter can be passed to modify the speed, as shown in the example below.
- A test binary is available on the intranet with the subject
- Example:
  `./colle2-1 "Hi everybody" "Go work!"`
- The scrolling speed can be given in parameter:
  `./colle2-1 -500 "Hi everybody" "Go work!"`
- The previous program does not execute at the same speed as:
  `./colle2-1 -50000 "Hi everybody" "Go work!"`
- If the speed parameter is not provided, you shall attribute a default speed of your choice.
- Note that the program must be able to take n character strings and is not limited at 2 strings (as in the examples).
- Of course, the various parameters can be combined.

Subject 02

- Binary name:

  ```
  colle2-2
  ```

- Your source code will be compiled with the following command:

  ```
  cc *.c -o colle2-2
  ```

- You have to develop a program that takes a character string in parameter.

- On a space of twice the size of this string, the string moves to the right then to the left while doing a rotation of the text in the direction of the movement...

- The program will never stop.

- A test binary is available on the intranet with the subject

- Example:

  ```
  ./colle2-2 "Enjoy the conf of the LoOsEr:)"
  ```

- The scrolling speed can be given in parameter:

  ```
  ./colle2-2 "Enjoy the conf of the LoOsEr:)" 5000
  ```

- The previous program does not execute at the same speed as:

  ```
  ./colle2-2 "Enjoy the conf of the LoOsEr:)" 90000
  ```

- If the speed parameter is not provided, you shall attribute a default speed of your choice.

Subject 03

- Binary name:
  `colle2-3`
- Your source code will be compiled with the following command:
  `cc *.c -o colle2-3`
- You have to develop a program that takes a character string in parameter and that will display this string, making the text "rotate" in the style of the information panel of the airports, character by character starting from the middle.
- If the word has an even number of characters, you must start with the two characters in the middle.
- A test binary is available on the intranet with the subject
- Example:
  `./colle2-3 "The Plane To Paris Will Be Delayed"`
- The scrolling speed can be given in parameter:
  `./colle2-3 "The Plane To Paris Will Be Delayed" 500`
- The previous program does not execute at the same speed as:
  `./colle2-3 "The Plane To Paris Will Be Delayed" 500000`
- If the speed parameter is not provided, you shall attribute a default speed of your choice.

Subject 04

- Binary name:
  `colle2-4`

- Your source code will be compiled with the following command:
  `cc *.c -o colle2-4`

- You have to develop a program that takes a character string in parameter and that will display this string by adding spaces to it and then by removing them.

- The spaces are added between the alphanumerical characters.

- However you must add a space between an alphanumerical character and a ponctuation character, but not between 2 ponctuation characters.
  Punctuation characters are:
  - .
  - ,
  - ;
  - '
  - ?
  - !
  - :

- You must remove the spaces without removing the leading spaces.

- A test binary is available on the intranet with the subject

- Example:
  `./colle2-4 "Hello everybody...  "`

- The scrolling speed can be given in parameter:
  `./colle2-4 "Hello everybody...  " -speed 5000`

- The previous program does not execute at the same speed as:
  `./colle2-4 "Hello everybody...  " -speed 500000`

- If the speed parameter is not provided, you shall attribute a default speed of your choice.

- We shall change the size of the project area:
  `colle2-4 "Hello everybody...  " -size 80`

- If the size parameter is not provided, you shall attribute a default size of your choice.
- Of course, all these parameters can be combined.

Subject 05

- Binary name:
  `colle2-5`

- Your source code will be compiled with the following command:
  `cc *.c -o colle2-5`

- You have to develop a program that takes a character string and a number in parameter. This number represents the number of characters between the side of the terminal and the "wall".

- The character string will be between the side of the terminal and the wall and will be pushed by a ball.

- When the text touches the wall, the last character is destructed, the ball changes of side and it restarts until complete destruction of the text.

- The ball must be a star "*".

- A test binary is available on the intranet with the subject

- Example:
  `colle2-5 "Our Asteks are good!" 42`

Think about all the possible errors and handle them.

That's all Folks!