README - CSC3002 Assignment1

The assignment guide for CSC3002 (23Spring) assignment1.

You may use `.txt` file to test individual problem on your own computer before submitting to OJ, and the command lines to compile and run:

Compile

- for making all the problems to the corresponding executable programs:

```
make all
```

- for making an individual problem to an executable program (problem 1 is `p1` , problem 2 is `p2` ... for instance):

```
make p1
```

Run & Test

Then you use following command to run the program, with the given test input in `p1.txt` (we use `p1` for instance).

The `<` in the command suggests the input file name, choose your desired test input file is also okay.

macOS

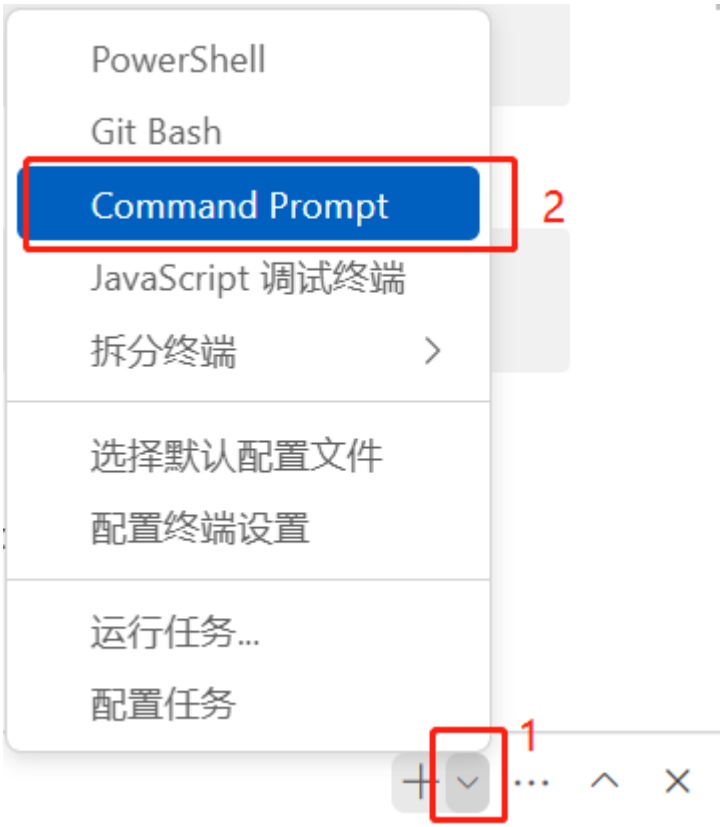
```
./p1 < p1.txt
```

Windows

```
p1.exe < p1.txt
```

NOTE: for win, you can only use "cmd" to run this command, "powershell" may lead to failures.

If your default terminal in VS Code is powershell, you can open a new cmd terminal by this:



Clean

To clean the intermediate `.o` object code and final executable files, you can run:

macOS

```
make clean
```

Windows

The delete files command on the Windows is not `rm`. So you need to change the last block of `Makefile`, from

```
clean:
    rm -f *.o *.a $(PROGRAM)
```

to

```
clean:
    del *.o *.a *.exe
```

And then run the `make clean` at the terminal.

Submit

Register your account and submit your codes following `Blackboard/Content/Assignments/Online Judge Guideline`.

On OJ, you only need to copy and paste the content from corresponding `.cpp` file for each individual question. For instance, in Problem 1, you need to paste the **whole content** of file "Combinatorics.cpp" onto the submission part in the OJ system, then click "Judge" to get your program run.

Miscellaneous

- Remember, **DO NOT** modify anything except the `TODO` part in the `.cpp` file in your "OJ_templates", otherwise your program will not be able to compile, and leads to zero mark.
- Notice that the score after pre-test is **NOT** your final score. Please check the score after our assignment due date.

Problem 2 Supplementary Instruction

For a certain input, there are 3 possible results for the DNA matches: 1 match, multiple mathes, and no matches. The outputs templates to be followed are all in the folder `./sample_outputs`, named `p2_sample_output.txt`, `p2_sample_output_2.txt` and `p2_sample_output_3.txt`, respectively.

Problem 3 Supplementary Instruction

NOTE: The goal of this task is to remove comments in the context. Other things (include whitespace) need to be kept as before. There are 3 cases to be considered

1. single-line `//` comment

   You need to remove the comments, but for the remaining things, **keep the same**, including whitespace.

   ```
   struct wait_opts wopts; // do_wait() args
   int status;// child process status
   ```

   should be converted as

   ```
   struct wait_opts wopts;
   int status;
   ```

   Note that there is a whitespace after ; in the first line but not in the last line, because the original input has one only in the first line.

2. single-line `/**/` comment

   Same as the last example: you need to remove the comments, but for the remaining things, **keep the same**, including whitespace.

   ```
   int pid, /*this pid*/ ppid; /*parent pid*/
   ```

should be converted as

```
int pid,  ppid;
```

Note that there are two spaces between `,` and `ppid`, and a whitespace after `;` because of the input.

3. multi-line `/**/` comment

In this case, you need to remove the comments but keep one single empty line. No code exists in the line.

```
/*
 * Ok, this is the main fork-routine.
 *
 * It copies the process, and if successful kick-starts
 * it and waits for it to finish using the VM if required.
 *
 * args->exit_signal is expected to be checked for sanity by the caller.
 */
pid_t kernel_clone(struct kernel_clone_args *args){}
```

should be converted as

```


pid_t kernel_clone(struct kernel_clone_args *args){}
```