Gebze Technical University

Computer Engineering Department

Introduction to Programming

CSE102 HW09

Fall 2014

Due Date

07.12.2014, 23:59

1. **(20 pts) string_rev:** Implement a recursive function that prints given string in reverse order. Do not use any helper functions.

   Input format:
   <string>
   number example
   Output
   elpmaxe rebmun

2. **(30 pts) maze_run:** Write a recursive function that finds the escape route from given maze. Starting point coordinates will be given as input parameters.

   Input format:
   <maze>
   <x><y>

   ```
   111101
   100101
   100101
   101101
   100001
   111111
   1 1
   ```

   Output

   ```
   1111.1
   1.01.1
   1.01.1
   1.11.1
   1....1
   111111
   ```

   Maximum maze size is 40x40.

3. **(50 pts) test_set_operations:** Reimplement all the functions presented in the 'Operations on Sets' case study in your text book in an iterative way (page 529-537). Also, write several functions to test set operations. Test all set operations in a selection loop. Possible commands are:
    a. em: Tester for is_empty. Is the current set empty?
    b. el <char>: Tester for is_element. Is given element in the current set?
    c. as <char> , …. , <char>: Adds elements to the current set (Until user hits enter).
    d. ps: Tester for print_set. Prints the current set.
    e. psc: Tester for print_set_with_commas. Prints the current set with commas.
    f. su <char> , …. , <char>Tester for is_subset. Is the given set is subset of the current set?
    g. un <char> , …. , <char>Tester for set_union. Finds the union of the given set and the current set. Also, the union set is assigned to the current set.
    h. q: breaks selection loop.

    You have to perform all input validity checks (whether the input is a valid set or nor, etc.).

    You should also write tester functions for all parts.

Your main function must be in following format;

```
int main(){
     //You have to complete the code and correct all kind of errors

     ////////////////////////////////////////
     puts("-------------------------------");
     printf("testing the function xxx \n");

     ...call test_part1...

     puts("-------------------------------");
     ////////////////////////////////////////


     ////////////////////////////////////////
     puts("-------------------------------");
     printf("testing the function xxx \n");

     ...call test_part2...

     puts("-------------------------------");
     ////////////////////////////////////////


     ////////////////////////////////////////
     puts("-------------------------------");
     printf("testing the function xxx \n");

     ...call test_part3...

     puts("-------------------------------");
     ////////////////////////////////////////
```

```
}
```

Sample Input (Your program should accept whole input data even if you cannot complete some parts of homework):

```
number example
111101
100101
100101
101101
100001
111111
1 1
em
as t k i o l
ps
as p o u
em
el t
el .
su y u o
un 4 u 8 h
psc
q
```

Notes:

- You should submit 1 file;
  - main.c
- Add all **files into a folder and compress it** for submission. The folder names will be restricted to the following format:
  HW#_studentid_studentname.
  - Example:
    HW01_121044001_Abdullah_Akay
- Upload soft copy of your homework to Moodle course web page
- **DON'T submit hard copy of your assignment**.
- Don't forget to test your code in the provided Linux virtual machine.
- Obey good programming rules (Indentation, Documenting, Well Commenting, Avoidıng magic numbers, Non-ascii characters etc.)
- **Strictly follow submission and file, folder naming rules.**