COMP1511 18s2

Sample Exam Questions for Week-05 Lab Practical Exam

You will be using the exam environment for your first lab exam. Your tutor will inform you how to login and start using your exam environment. There are three questions in this exam, and you need to answer all three questions. The required files will be available in the directories named "q1", "q2" and "q3' in your (exam) home directory. You need to submit the required files as specified in each exercise.
Important: in order to receive marks for an exercise, your program must compile successfully without any errors and pass "new" auto-tests. You will be awarded marks only if your solution is correct and properly solves the problem (no hard-coded solutions!). If your algorithm/solution is incorrect, even if you pass auto-tests, you will NOT be awarded marks.

Question 01

The following two files will be available in the directory named "q1":

- `isLeapYear.c`
- `test_isLeapYear.c`

You need to implement a C function `isLeapYear`, in the given file `isLeapYear.c`, that takes a year as an argument and calculates whether that year is a leap year or not. The function must return 1 if the year is a leap yer, 0 otherwise.

```
int isLeapYear(int year)
```

For example,

```
isLeapYear(2018)          returns   0
isLeapYear(2000)          returns   1
isLeapYear(1900)          returns   0
```

For this question, you need to use the provided file `test_isLeapYear.c` to test your function. Please open the file `test_isLeapYear.c` using an available text editor, try to understand how your function is tested.

You can use the following commands to compile and run few tests already provided in the file `test_isLeapYear.c`. Please make sure that you also test your function extensively by adding more test cases in `test_isLeapYear.c`. You will be submitting only one file `isLeapYear.c`. To run the simple tests available in `test_isLeapYear.c`, execute the following commands:

```
$ dcc -o test_isLeapYear    isLeapYear.c    test_isLeapYear.c
$ ./test_isLeapYear
```

Again, please make sure that you test your function extensively by adding more test cases in `test_isLeapYear.c`.

You will be submitting only one file `isLeapYear.c` for this question. Submit your file `isLeapYear.c` with the following command, from "q1" directory:

```
$ submit q1
```

Question 02

Write a program called `outliers.c` that reads integer values from standard input until the end of input stream or first unsuccessful read. The program finds number of values that are outside the range of 5 to 25 (inclusive), and prints the number of such outliers.

Make your program match the examples below exactly.

```
$ ./outliers
Enter number: 12
Enter number: 4
Enter number: 25
Enter number: 15
Enter number: 27
Enter number: Ctrl + D
Outliers: 2


$ ./outliers
Enter number: 0
Enter number: 26
Enter number: 5
Enter number: -2
Enter number: Ctrl + D
Outliers: 3
```

When you think your program is working you can use `autotest` to run some simple automated tests. From the directory "q2", run the following command:

```
$ ./autotest
```

Please make sure that you test your program extensively by considering more test cases.

You will be submitting only one file `outliers.c` for this question. Submit your file `outliers.c` with the following command,

Question 03

The following two files will be available in the directory named "q3":

- arrayPositiveMin.c
- test_arrayPositiveMin.c

You need to implement the following function that returns minimum positive value from a given array `a` (of type `int`). A value is a positive value if it is greater than zero. If there are no positive values in a given list, the function should return zero.

```
int arrayPositiveMin(int a[], int size)
```

For example,

```
arrayPositiveMin([4, 2, 9], 3)                       returns  2
arrayPositiveMin([5, -6, 3 , 20], 4)                 returns  3
arrayPositiveMin([5, -6, -12 , 5, 0, -2 , 8], 7)     returns  5
arrayPositiveMin([-2, -4, -8, -2], 4)                returns  0
```

For this question, you need to use the provided file `test_arrayPositiveMin.c` to test your function. Please open the file `test_arrayPositiveMin.c` using your favourite text editor, try to understand how your function is tested. If you have any questions, please ask your tutor.

You can use the following commands to compile and run few tests already provided in the file `test_arrayPositiveMin.c`. Please make sure that you also test your function extensively by adding more test cases in `test_arrayPositiveMin.c`. To run the simple tests available in `test_arrayPositiveMin.c`, execute the following commands:

```
$ dcc -o test_arrayPositiveMin   arrayPositiveMin.c   test_arrayPositiveMin.c
$ ./test_arrayPositiveMin
```

Again, please make sure that you test your function extensively by adding more test cases in `test_arrayPositiveMin.c`.

You will be submitting only one file `arrayPositiveMin.c` for this question. Submit your file `arrayPositiveMin.c` with the following command, from "q3" directory:

```
$ submit q3
```

-- end --