The assignment will be graded out of 100 points.

Due: 11:59pm Monday, July 23, 2018

Submission Guidelines:

Assignment will be submitted via Blackboard. If there are multiple files that you are submitting then you should zip your files into a single file and submit it. The name of this file should be in following format: lastname_firstname_UTANetID. If you are submitting a single file, make sure it is in the following format: word, txt or pdf and it should be named as mentioned above.

Make sure your name and your student ID are listed in your assignments.

If your assignment is not completed by the deadline, send it anyway and review it with the TA for partial credit. Do not take a zero or excessive late penalties just because it isn't working yet. We will make an effort to grade you on the work you have done.

Note: Please do not forget to zip all the files in ONE folder

Example:

Task1.txt
Task2.c
Task3.c
Task4.c
Task5.c
Task6.c
Task7.c

After that, copy all the C files into one folder and compress(zip) the folder. Then, submit the compressed (zipped) folder on Blackboard.

Assignment Specification:

For ALL the Tasks please get the values from user as input.

Task 1 (10 pts.)

In a file called task1.txt, write 2 differences between malloc() and calloc() functions in c.

Task 2 (15 pts.)

In a file called task2.c, write a C program to create memory (dynamic) for int, char and float variable at run time. Then, get their values from user and print them on the screen. Finally, free allocated memory.

Hint: You can use either malloc() or calloc() for this program.

Example:

Input:
Enter an Integer value: **2**
Enter a character: **A**
Enter a float value: **3.20**

Output:
Inputted values are: **2, A, 3.20**.

Task 3 (15 pts.)

In a file called task3.c, write a C program to input and print text using Dynamic Memory Allocation.

Hint: You can use either malloc() or calloc() for this program.

Example:

Input:
Enter the length(limit) of the text: **100**
Enter your text: **I am writing this program to learn dynamic memory allocation.**

Output:
The Text which you entered: **I am writing this program to learn dynamic memory allocation.**

Task 4 (15 pts.)

In a file called task4.c, write a C program to read a one-dimensional int array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

Hint: You can use either malloc() or calloc() for this program.

Example:

Enter the number of elements: **5**

Enter the element 1: **24**
Enter the element 2: **37**
Enter the element 3: **21**
Enter the element 4:**11**
Enter the element 5: **66**

Array Elements are: 24 37 21 11 66
Sum of all the elements: 159

Task 5 (15 pts.)

In a file called task5.c, write a C program to find the largest element using Dynamic Memory Allocation.

Hint: You can use either malloc() or calloc() for this program.

Example:

Enter total number of elements: **10**

Enter Number 1: 1.44
Enter Number 2: -11.63
Enter Number 3: 52.12
Enter Number 4: 12.45
Enter Number 5: 17.64
Enter Number 6: -9.25
Enter Number 7: 5.45
Enter Number 8: 19.76
Enter Number 9: -5.33
Enter Number 10: 34.19

output:
The largest element is: 52.12

Task 6 (15 pts.)

In a file called task6.c, write a C program to sort numbers in ascending and descending order by using calloc() function. (At the end use free() to release memory).

Example:
Enter total number of elements: **10**

Enter Number 1: 1.44
Enter Number 2: -11.63
Enter Number 3: 52.12
Enter Number 4: 12.45
Enter Number 5: 17.64
Enter Number 6: -9.25
Enter Number 7: 5.45
Enter Number 8: 19.76
Enter Number 9: -5.33
Enter Number 10: 34.19

Output:

Ascending Order:
-11.63
-5.33
1.44
5.45
12.45
17.64
19.76
34.19
52.12

Descending Order:
52.12
34.19
19.76
17.64
12.45
5.45
1.44
-5.33
-11.63

Task 7 (15 pts.)

In a file called task7.c, write a C program to reverse a string using **Dynamic memory allocation**.

Example:

Input: Programming
Output: gnimmargorP

Another Example:
Input: CSE1320
output: 0231ESC

Assignment Guidelines:

There will be several programming assignments in this course, typically assigned on a weekly basis. All assignments will have equal weight. No assignment scores will be dropped. The following class policies regarding assignments will be followed:

- All assignments must be submitted via Blackboard.
- No extra credit will be provided.
- If you make multiple submissions to Blackboard for the same assignment, only the latest submission will be graded.

Late submission policy:

- All assignments are graded out of 100 points. Assignments submitted late will be penalized, at a rate of 4 penalty points per hour. The submission time will be the time shown on Blackboard. Any assignment submitted more than 25 hours late will receive no credit.
- Exceptions to late submission penalties will only be made for emergencies documented in writing, in strict adherence to UTA policy. For all such exception requests, the student must demonstrate that he or she made all efforts to notify the instructor as early as possible.
- Computer crashes, network crashes, and software or hardware failure will NOT be accepted as justification for late submissions. If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline.
- Sometimes students submit the wrong files on Blackboard. Unfortunately, no credit or waiver of late penalties can be provided in such cases.
- If you find yourself in an emergency situation and cannot deliver homework on time, immediately inform the instructor and teaching assistant. Even if you have a valid reason for delivering late an assignment, you must make a convincing case that you have notified the instructor and teaching assistant as early as possible.

If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline (maximum 3 attempts).