

**Recep Tayyip Erdogan University**

**Faculty of Engineering and Architecture**

**Computer Engineering**

CE103- Algorithms and Programming I

**Homework-4 (Week-11)**

**Fall Semester, 2020-2021**

| Instructor | Asst. Prof. Dr. Uğur CORUH |
| --- | --- |
| Contact Information | [ugur.coruh@erdogan.edu.tr](mailto:ugur.coruh@erdogan.edu.tr) |
| Google Classroom Code | **ouw44uk** |
| Publish Date | **23.12.2020** |
| Due Date | **31.12.2020 23:59** |

**Complete the following homework requirements, prepare them in the format given in the link below until the deadline and time, and upload them to the related assignment in the classroom.**

<https://drive.google.com/file/d/1yqSXZZ3346iIqotb_e_yzaryfxEXE0fR/view?usp=sharing>

**Grades:**

| Problem-1 | 50 points |
| --- | --- |
| Problem-2 | 50 points |
| **Total** | **100** points |

*You will develop the following examples in C language. The examples with C ++ will not be accepted.*

NOTE: In the code, add the following information with printf. Put the description of each application in the problem part.

*int main(void)*

*{*

*printf("Build Time: %s %s\n", \_\_DATE\_\_, \_\_TIME\_\_);*

*printf("Owner: Name Surname\n");*

*printf("Student ID: 11111111\n");*

*printf("Course: CE-103\n");*

*printf("Homework: 1\n");*

*printf("Problem: “Printing the Text Entered on the Screen in Reverse \n");*

*... your codes...*

*}*

**Problem-1**: *Matrix Multiplication (50)*

In this application you will develop a C application it will get matrix row and column size as an input and matrix values will be randomly generated by your application. After generating the matrix. You will develop 3 matrix multiplication function for your application. You will run these 3 algorithms with the same matrix input and measure the multiplication time. You will do this measurement 100 times and then you will get the average of all measurements programmatically. Also, compare the times. You will implement the following algorithms in the same application.

* Normal Multiplication
* Recursive Multiplication
* Strassen Multiplication

**Problem-2**: *Quicksort (50)*

In this application, you will develop a C application for quicksort performance measurement. You will develop, measure, and compare the following algorithms. You will create random inputs for your application.

* Quicksort with Hoare Partitioning

QUICKSORT (A, p, r)

if p < r then

q ← H-PARTITION(A, p, r)

QUICKSORT(A, p, q − 1)

QUICKSORT(A, q +1, r)

* Quicksort with Lomuto Partitioning

QUICKSORT (A, p, r)

if p < r then

q ← L-PARTITION(A, p, r)

QUICKSORT(A, p, q − 1)

QUICKSORT(A, q +1, r)

* Quicksort with Left Hoare and Right Lomuto Partioning

QUICKSORT (A, p, r)

if p < r then

q ← H-PARTITION(A, p, r)

QUICKSORT\_L(A, p, q − 1)

QUICKSORT\_R(A, q +1, r)

-------------------------

QUICKSORT\_L (A, p, r)

if p < r then

q ← H-PARTITION(A, p, r)

QUICKSORT\_L(A, p, q − 1)

QUICKSORT\_L(A, q +1, r)

-------------------------

QUICKSORT\_R(A, p, r)

if p < r then

q ← H-PARTITION(A, p, r)

QUICKSORT\_R(A, p, q − 1)

QUICKSORT\_R(A, q +1, r)

* Quicksort with One Step Hoare and Next Step Lomuto Partitioning

flag default = false

---------

QUICKSORT (A, p, r, flag)

if p < r then

if flag == true

q ← H-PARTITION(A, p, r)

flag = false

else

q ← H-PARTITION(A, p, r)

flag = true

endif

QUICKSORT(A, p, q − 1,flag)

QUICKSORT(A, q +1, r,flag)

**Note : Please provide source code of applications with your reports**