Faculty of Computers and Information – Fayoum University

Computer Architecture – Practical Project Sheet

V1.0

Course Title:

Computer Architecture

Course Team:

Dr. Hewida Youssri
Mahmoud Badry (mma18@fayoum.edu.eg)

Project Title:

A team has to choose from these projects

1. **Shape Detector**
2. **String Drawer**
3. **Binary Search**
4. **Primer Number Detector**
5. **Digital clock**

Project Submission Information

- This is a team-based project.
- Each Student must join a team of **6 members** at maximum.
- Each team should send **an email message** to Mahmoud Badry that includes the team members and the chosen project. Email must be sent before **02 Jan 2021**. Projects should be balanced on all the students.
- Project will be discussed at the logic design lab in our college and will be uploaded to the Google classroom project's section.

Submission Date:

Final Exam Day (5-02-2021)

- Each team will have an announced **delivery time** that will be public on our Google's classroom

Marking and Assessment

This assignment will be marked out of 120%. 20% of the grade is an extra credit grade (Bonus)

Learning outcomes

- Ability to write Assembly Language programs for the Intel 8086 microprocessors.
- Ability to debug and interpret machine code.
- Ability to examine and modify the contents of Memory.
- Knowledge of data transfer instructions, arithmetic instructions, logic instructions, shift instructions, and rotate instructions.
- Knowledge of Control flow and loop instructions.
- Ability to work with the MDA kit 8086 hardware.

IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED.

COPYING FROM EACH OTHER ISN'T ALLOWED.

I. Project – Introduction

This document will present 5 projects in the following order:

1. Shape Detector

2. String Drawer

3. Binary Search

4. Primer Number Detector

5. Digital clock

II. Project – Shape detector

Project Title: - "**Shape detector**"

Description: - User enter some unsigned points in one line and then the programs should predict the shape of these numbers if they represent a square, a rectangle, a triangle, or unknown. The result has to be drawn in dot matrix.

Details: -

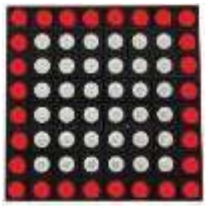
- User inputs **unsigned 2D** points separated by **Enters**.
    - Example: (1,1)
      (2,2)
      (3,3)
- Your program should read the points and the detect the shape from the list [Square – Rectangle – Triangle - Unknown]
- Points user enters has to be exact and sorted. For example a triangle should only have 3 points.
- View the result on the input screen as text and on the dot matrix as a drawing.
- Bonus: Let your shape detector detect pentagon and hexagonal and draw them as well.

Example

The input screen

```
Enter the points:
```

```
(1,1)
(3,1)
(3,3)
(1,3)
```



You will submit:

- A complete flow chart for your program
- Procedures description that contains
    - Task accomplished by the procedure.
    - List of input parameters and their usage.
    - Description of any value calculated/returned by the procedure.
    - Preconditions that must be satisfied before the procedure is called.

III. Project – String drawer

Project Title: - "**String drawer**"

Description: - User enters a string in the LCD screen and then it's viewed on the dot matrix character by character.

Details: -

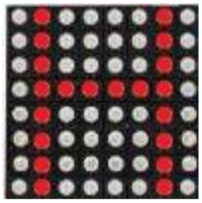
- Allowed characters are all English upper-case characters and the numbers.
- Keypad has to have the ability of entering all the English characters through the numbers from 0 to 9 like old mobiles.
- Only red LEDs will be used in the dot matrix.
- A delay between each viewed character should be presented. For example, 1 or 2 seconds.
- Bonus: Instead of delays between viewed characters, the characters should move from right to left.

Example

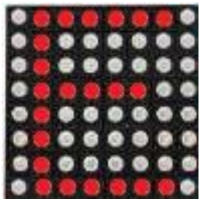
The input screen

```
HEN
```

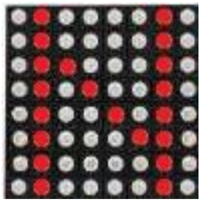
The dot matrix should view



Then after 2 sec



Then after 2 sec



You will submit:

- A complete flow chart for your program
- Procedures description that contains
    - Task accomplished by the procedure.
    - List of input parameters and their usage.
    - Description of any value calculated/returned by the procedure.
    - Preconditions that must be satisfied before the procedure is called.

IV. Project – Binary Search

Project Title: - "**Binary search**"

Description: - User enters a list of unsigned numbers and then searches if a specific number exists to present its occurrence in the entered list.

Details: -

- Maximum allowed number length is 4.
- Comma should be used to separate numbers, any keypad than the numbers can be used as comma.
- Binary algorithm must be used as the search algorithm.
- Number of occurrences an input search should be viewed on the LCD.
- Bonus: If the number occurrence of the searched number is less than 9, view it in the dot matrix.

Example

The input screen

```
Enter the list:
```

```
111, 222, 333, 444
```

```
Enter number to search:
```

```
222
```

```
Result: (1)
```

You will submit:

- A complete flow chart for your program

- Procedures description that contains
    - Task accomplished by the procedure.
    - List of input parameters and their usage.
    - Description of any value calculated/returned by the procedure.
    - Preconditions that must be satisfied before the procedure is called.

V. Project – Prime number detector

Project Title: - "**Prime number detector**"

Description: - User enters a list of unsigned numbers and then searches for prime numbers inside the list. It then outputs the numbers in a comma separated fashion.

Details: -

- Maximum allowed number length is 3.
- Comma should be used to separate numbers, any keypad than the numbers can be used as comma.
- Prime numbers should be viewed in a comma separated fashion
- Bonus: Count the number of prime numbers, if the number is less than 9, view it in the dot matrix.

Example

The input screen

```
Enter the list:
```

```
0, 1, 2, 3, 4, 5, 6
```

```
Result: (2,3,5)
```

You will submit:

- A complete flow chart for your program
- Procedures description that contains
    - Task accomplished by the procedure.
    - List of input parameters and their usage.
    - Description of any value calculated/returned by the procedure.
    - Preconditions that must be satisfied before the procedure is called.

VI. Project – Digital Clock

Project Title: - "**Digital Clock**"

Description: - A digital clock program that shows the current time. It starts with 12:00:00 AM, after that the user can change either the hours, the minutes, the seconds, or AM/PM.

Details: -

- At start the 12:00:00AM will be presented and the seconds should count. Means after one second it will be 12:00:01AM.
- Seconds calculation should be accurate.
- A user can use the A key bad to stop the clock and start in editing more.
- The user can move the cursor using the + and – key bad and then made his edits using the numbers key bad. For the AM/PM, AM=1 and PM=2
- Bonus: In edit mode, use a key that sets the current time as an alarm. If the alarm hits, the dotmatrix should be lighted with Red 10 times (Between each one and the other is 1 second).

Example

The input screen	`12:00:00AM`
After 1 Sec	`12:00:01AM`
A is pressed	`12:00:01AM`
User edited	`1:30:00PM`
After 1 Sec	`1:31:00PM`

You will submit:

- A complete flow chart for your program
- Procedures description that contains
  - Task accomplished by the procedure.
  - List of input parameters and their usage.
  - Description of any value calculated/returned by the procedure.
  - Preconditions that must be satisfied before the procedure is called.