**ASSIGNMENT – 1 REPORT**

**Group Members:**

1. Shashank Devarmani – IMT2022107

2. Kandarp Dave – IMT2022115

**Program Chosen: Sorting Algorithm**

**Explanation:**

1. IMT2022\_107\_115\_sort.asm

This assembly program takes 4 below inputs and gives out the sorted output in different lines:

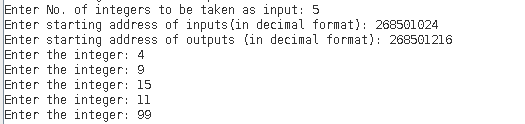
1. no. of integer inputs
2. starting address where input needs to be stored
3. starting address where output needs to be stored
4. the integer inputs

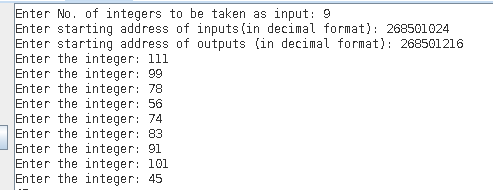
The program uses the selection sort algorithm to sort the numbers in ascending order.  
It first copies the elements from the input address to the output address.  
Then, it uses two for loops to perform the iterative version of the selection sort algorithm to sort the numbers.

To implement the for loops, various labels like outerLoop, innerLoop are used.

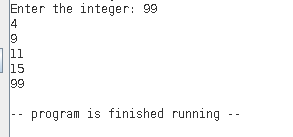
To implement the if-else conditions, labels like if\_condn and after\_if\_condn are used.

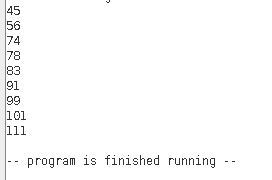
**Screenshots of input**:





**Screenshots of output**:





2. IMT2022\_107\_115\_sorting\_algorithm.asm

Contains the selection sort algorithm, without the template. The C equivalent of this code can be found in IMT2022\_107\_115\_selection\_sort.c.

3. IMT2022\_107\_115\_assembler.py

This python program reads the MIPS assembly code and converts it into binary format.

The assembly code is written in inpfile.txt. The program reads this code and writes the binary code in a file name outfile.txt.

Each instruction is split using spaces and the corresponding instruction is converted to binary using various dictionaries. The dictionaries contain mapping from register name to register number, instruction name to opcode, etc.

4. IMT2022\_107\_115\_checkBinary.py

This program can be used to verify the binary code generated by the python assembler. Make sure to first run the assembler.py program. To use this program, follow these steps:

1. Open IMT2022\_107\_115\_sort.asm in the MARS simulator.
2. Assemble the program. Click on the option ‘Dump machine code or data in an available format.
3. Select .text as the memory segment and binary text as the dump format.
4. Click on ‘Dump File’ and save the file as ‘MARS\_binary.txt’ in the project folder.
5. Open MARS\_binary.txt and delete the first 19 lines. Now, delete everything except the first 36 lines. Please verify that the file contains only 36 lines.
6. Run IMT2022\_107\_115\_checkBinary.py

The code will display mismatch found, if any.

**Screenshot after execution of both the files:**

