PROGRAMMING PROJECT 1
Due: 16 / 04 / 2021 – 23:30 (No late submission)

In this project, you are required to implement given procedures in MIPS assembly language. You will use a MIPS simulator (QTSPIM or MARS) to develop and test your code. There will be four questions in the project which are unrelated.

QUESTION 1. *(10 points)* In this program you are required to count number of occurrences of characters in a given string. Your procedure must ignore the non-alphabetic characters including numbers, punctuations etc. Upper and lower case letters must be counted as same character. Given the following string as an input, output must be printed as on the table with sorted character list and number of occurrences.

An example run:

Enter the String: This is Computer Organization course!

Program output must be:

Character	*Occurence*
o	*4*
t	*3*
i	*3*
s	*3*
r	*3*
c	*2*
u	*2*
e	*2*
a	*2*
n	*2*
h	*1*
m	*1*
p	*1*
g	*1*
z	*1*

QUESTION 2. *(12 points)* In this procedure, the input character list contains ASCII characters representing single-digit or multi-digit decimal numbers, separated by spaces. You are required to sort these numbers (*from smaller to larger*) separated by one space character and store the sorted list in the output_list argument. (The numbers may be negative, which means that there may be minus sign as the first character of a number.)

An example run:

Input: 1 6 23 -5 18
Output: -5 1 6 18 23

Note: Multi-digit decimal numbers must be converted from ascii representation and number of digits might vary up to the 32 bit data size limits. For example -98625 can be also given as an integer.

QUESTION 3. *(12 points)* Write a MIPS assembly program to calculate the **num_prime**(N) function, which is the number of prime numbers less than N. Use the sieve of Eratosthenes method to generate prime numbers in the interval [2,N]. Suppose that maximum value of N can be 1,000,000. Here is an example tabulation of **num_prime**(N):

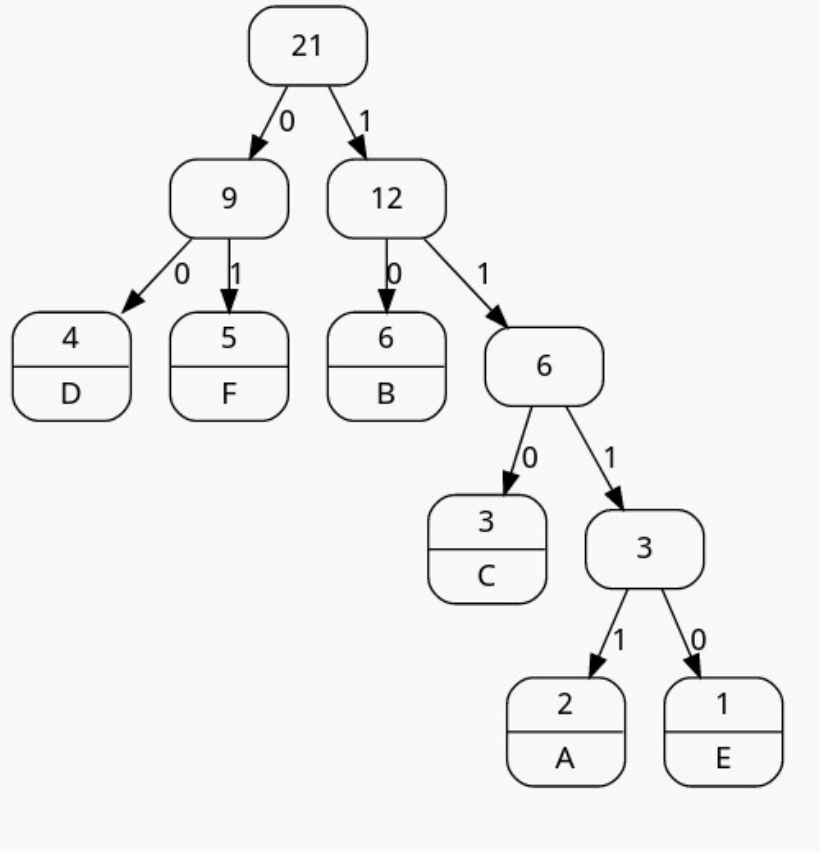
*Please enter an integer number for **num_prime**(N):*
10
prime(10) is 4.
*Please enter an integer number for **num_prime**(N):*
100
prime(100) is 25.
*Please enter an integer number for **num_prime**(N):*
1000000
prime(1000000) is 78498.

QUESTION 4. *(23 points)* Write a MIPS assembly program to construct Huffman Code tree [1]. Huffman code is used to compress data by assigning the shorter binary strings for more occurring characters in a string. As can be seen from the sample run given below, there will be two input strings. Your program will accept the first string to construct Huffman code tree (*Hint: You can use Question 1 which counts the occurrences of characters in a string*). The second input will be a string to be coded using the constructed Huffman Code. The output of the program will be binary compressed string. Assume that in the second input string there will be no characters that do not exist in the first string. Note that spaces will be ignored from both input strings and will not be printed on output strings.

An example run:

Enter the string to construct Huffman Code: AABBBCCBBDDDDEFFFFFBC
Enter the string to be converted using Huffman Code: ABC
Output:111110110

Output can be constructed using the Huffman code tree[1]. Following figure illustrates the Huffman tree for given string.



Character	Occurence	Binary Code (After Running Huffman Coding)
A	2	1111
B	6	10
C	3	110
D	4	00
E	1	1110
F	5	01

MENU *(8 points):* Your program should support a *Menu* including all questions above. A sample execution scenario given below:

```
Welcome to our MIPS project!
Main Menu:
1. Count Alphabetic Characters
2. Sort Numbers
3. Prime (N)
4. Huffman Coding
5. Exit
Please select an option: 1
```

These options must be printed inside a loop until "Exit" option is selected.
When the user select option 1, you should print the followings:

```
Enter the String: This is Computer Organization course!
Character Occurrence
o         4
t         3
i         3
s         3
r         3
c         2
u         2
e         2
a         2
n         2
h         1
m         1
p         1
g         1
z         1

Main Menu:
1. Count Alphabetic Characters
2. Sort Numbers
3. Prime (N)
4. Huffman Coding
5. Exit
Please select an option: 2
Input: 1 6 23 -5 18
Output: -5 1 6 18 23




Main Menu:
```

```
1. Count Alphabetic Characters
2. Sort Numbers
3. Prime (N)
4. Huffman Coding
5. Exit
Please select an option:  3
Please enter an integer number for prime(N):100
prime(100) is 25.

Main Menu:
1. Count Alphabetic Characters
2. Sort Numbers
3. Prime (N)
4. Huffman Coding
5. Exit
Please select an option: 4
Please enter the string to construct Huffman Code:
AABBBCCBBDDDDEFFFFFBC
Please enter the string to be converted using Huffman Code:
ABC
Output:111110110

Main Menu:
1. Count Alphabetic Characters
2. Sort Numbers
3. Prime (N)
4. Huffman Coding
5. Exit
Please select an option: 5
Program ends. Bye :)
```

Assumptions and Requirements

- The arguments to the procedures are stored in $a registers; i.e., the first one is in $a0, the second one is in $a1, and so on.
- Only valid arguments are passed into the procedures. Therefore, you do not need to check the arguments for their validity.
- When you invoke a procedure, the values of all $a registers should be preserved. Their values should be same at the end of the procedure call as they were at the time of call.
- You have to use QtSpim or MARS simulator in your implementation. Any other simulator is not allowed.
- You are required to submit a minimum 2-page report (***5 points***) explaining implementation details of your project. Your report will have four parts (one for each question) and it will also include screenshot of your sample runs, as well.
- You should submit a fully commented source code that includes details of your implementation. Note that the name of the file should include surnames of the group members. (ex: surname1_surname2_surname3_surname4.s)

- Zip your fully commented source code file and the project report into a single file and submit the zip file via Canvas.

General Policies for the Project

- ***You have to work in groups of 3 or 4***. You will select your partners and partners will not be changed throughout the semester. *It is not acceptable of a partner team to work with other teams.*
- *A portion of your project grade will be set with a Project Quiz.* Note that if you do not submit the project, you will not be allowed to attend the Project Quiz.
- Copying (partially or full) solutions from other students is a form of cheating. Copying (partially or full) solutions from Web including Github (and similar sites) is another form of cheating. It is NOT acceptable to copy (or start your) solutions from Web. **In case of any forms of cheating or copying among the groups, the penalties will be severe. Both Giver and Receiver are equally culpable and suffer equal penalties!!!**
- No late submission will be accepted!

References

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms* (3rd Edition), MIT Press, 2009.

** Note that this is an example textbook that you can find information on Huffman Code tree. You can find similar materials from other textbooks on algorithms, as well.