



**MARMARA UNIVERSITY  
FACULTY OF ENGINEERING  
COMPUTER ENGINEERING**


**CSE3038 COMPUTER ORGANIZATION  
PROJECT 1 | REPORT**

**OĞUZCAN ÇELİK | 150114062  
MEHMET ALİ SAKALLI | 150114080**

## Menu

In the first part of the program, it prints a welcome message. Then it starts a loop and prints menu message, takes an input from the user that specifies which program will be ran. After taking the input, it finds the matching question from the menu jump table and jumps to that address. If user gives 4 as input, program ends.

```

 Console
Welcome to our MIPS project!

Main Menu:
1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit
Please select an option:
```

## Question 1 – Number Series

If user gives 1 as input, question 1 executes. It gets first number of the series, number of the integers in the series and the offset between two integers in the series from the user one by one and stores each of them in \$t0, \$t1 and \$t2 registers, respectively. In addition to that, \$t4 register stores the summation of the numbers and it is initially 0.

After taking inputs from the user, it prints “The series is:” a loop starts. In each iteration, program decreases \$t1, number of integers in the series, by 1. Then it prints the number in the \$t0 and a blank space. After that, it adds \$t0 to total summation and increases \$t0 by \$t2, offset value. In the end, it compares \$t1 with zero. If it is not zero, next iteration starts. Otherwise, loop finishes and it prints a new line, output message and the summation of the numbers. After execution is done, returns to the main menu.

### Example Run - 1

```

Main Menu:
1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit
Please select an option: 1

Enter the first number in the series: 7

Enter the number of integers in the series: 5

Enter the offset between two successive integers in the series: 4

The series is: 7 11 15 19 23
The summation of the numbers is: 75
```

## Example Run - 2

Main Menu:

1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit

Please select an option: 1

Enter the first number in the series: 12

Enter the number of integers in the series: 16

Enter the offset between two successive integers in the series: 24

The series is: 12 36 60 84 108 132 156 180 204 228 252 276 300 324 348 372

The summation of the numbers is: 3072

## Question 2 – Prim's Algorithm

If user gives 2 as input, question 2 executes. It gets a graph as input from the user and stores this input in a variable called q2input. Then, it calculates the length of the input, stores it in \$t2 register and a variable called q2inpSize.

There is a variable called q2visited, which stores the visited nodes in the graph and a variable called q2visSize, which stores the number of visited nodes. After calculating the length of the input, it takes the first node and adds that node to the visited nodes. Then it increases the q2visSize by 1.

There are registers that store specific values.

\$s0: stores the value of "counter mod 3"

\$s1: stores if the first node of the current edge is visited

\$s2: stores if the second node of the current edge is visited

\$s5: stores the current least weight, initially -1.

\$s6: stores the first node of the current edge

\$s7: stores the second node of the current edge

\$t2: stores length of the input

\$t3: stores the ASCII value of the "space"

\$t7: stores the first node of the edge with least weight

\$t8: stores the second node of the edge with least weight

\$t9: stores the total weight

Then it starts a loop and reads the given input byte by byte. If it matches with space, continues to the next iteration. There is a counter that checks if the current value is the first node of the edge, second node of the edge or the weight between nodes. Program compares this value with 0, 1 and 2 to detect current value. If it is 0 or 1, it starts a loop and checks if the current node is visited and sets the \$s1 and \$s2 registers. If the counter is 2, it resets the counter to 0 and checks \$s1 and \$s2 registers. If only one of them is set to 1, that means current edge is an available edge to choose. If it is the first edge, that means if \$s5 = -1, directly sets \$s5 to the current weight. Otherwise compares the current weight with \$s5, if it is less than \$s5, sets \$s5 to the current weight again.

After finishing the first loop, we have the chosen edge in our hand. Program prints this edge to the screen in this order: first\_node second\_node weight. Then adds this weight to the total weight, adds nodes to visited nodes, increases visited nodes size by 2. When adding current weight to the total weight, it decreases the weight by 48 because it stores the ASCII value of the integer and decreasing by 48 gives us the true value of that integer.

Then it starts to read the input byte by byte again. If \$s5 is still set to -1 when the loop is done, that means tree is generated and it's time to terminate the question. In the end, it prints the total weight. Before returning to menu, it clears the user input and visited nodes and sets the length of the visited nodes to 0 again. Then it returns to the main menu.

### Example Run – 1, 2

Main Menu:

1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit

Please select an option: 2

Enter the graph: 0 1 8 0 2 6 0 3 3 1 2 2 1 5 5 2 4 2 3 4 1 4 5 3

Minimum Spanning Tree:

0 3 3

3 4 1

2 4 2

1 2 2

4 5 3

Total weight is : 11

Main Menu:

1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit

Please select an option: 2

Enter the graph: 0 1 2 0 2 7 0 5 8 1 2 4 2 3 2 2 4 3 2 6 4 3 4 9 3 6 1 5 6 6

Minimum Spanning Tree:

0 1 2

1 2 4

2 3 2

3 6 1

2 4 3

5 6 6

Total weight is : 18

### Question 3 – Encrypt/Decrypt

If user gives 3 as input, question 3 executes. Takes an input string and an offset value as input and stores input string in a variable called q3input. Firstly, it calculates the length of the input. After calculating the length, it prints the source string to the screen. But when printing the source, it converts all lowercase letters to uppercase. To detect if a letter is in lowercase, it makes some calculations.

If the value of the character is less than 'A' or greater than 'z', that means it's not a letter. In addition to that, values between 'Z' and 'a' also does not represents a letter. So, when printing the source, program prints all characters except letters directly and if it is a lowercase letter, converts it to uppercase by decreasing the value by 32.

After that, it checks if the offset is positive or negative. If it is positive, makes encryption. Otherwise, decrypts the string.

There is a variable called q3str5 and it stores the alphabet. Its value is "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ". It stores the alphabet two times to make shifting easier.

To encrypt the string, it starts to read input byte by byte in a loop. In each iteration, it finds the letter in the alphabet, than adds the offset to that value. Matching letter with that number in the q3str is the corresponding letter to the current letter. It prints that letter and continues to the next letter in the next iteration.

Decryption works the same as the encryption. The only difference is after finding the letter in the alphabet it adds 26 to that value then subtracts the offset. Matching letter in the alphabet gives us the decrypted version of that letter. Then prints the letter and moves to the next iteration.

Both encryption and decryption converts lowercase letters to uppercase, too. After each letter is printed to screen, it clears the user input and returns to the main menu.

#### Example Runs – 1

Main Menu:

1. Number Series
2. Prims Algorithm
3. Encrypt/Decrypt
4. Exit

Please select an option: 3

Enter an input string: I am an engineer!

Enter an offset value: 25

SOURCE: I AM AN ENGINEER!

PROCESSED: H ZL ZM DMFHMDQ!

## Example Runs – 2, 3

```
Main Menu:
 1. Number Series
 2. Prims Algorithm
 3. Encrypt/Decrypt
 4. Exit
Please select an option: 3

Enter an input string: I am an engineer!

Enter an offset value: -1

SOURCE: I AM AN ENGINEER!
PROCESSED: H ZL ZM DMFHMDDQ!

Main Menu:
 1. Number Series
 2. Prims Algorithm
 3. Encrypt/Decrypt
 4. Exit
Please select an option: 3

Enter an input string: vdkbnld sn lzqlzqz tmhudqrhsx :)

Enter an offset value: -25

SOURCE: VDKBNLD SN LZQLZQZ TMHUDQRHSX :)
PROCESSED: WELCOME TO MARMARA UNIVERSITY :)
```

## EXIT

If user gives 4 as input, program prints exit message and terminates.

## Example Run - 1

```
Main Menu:
 1. Number Series
 2. Prims Algorithm
 3. Encrypt/Decrypt
 4. Exit
Please select an option: 4

Program ends. Bye...|
```