

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

FUNDAMENTALS OF PROGRAMMING I GROUP PROJECT I

GROUP MEMBERS ID Number

1. Tewuhbo Mihret ETS1554/14
2. Tinsae Daniel ETS1560/14
3. Tsion Samuel ETS1576/14
4. Tsion Getachew ETS1581/14
5. Tsion Kassahun ETS1585/14
6. Tsiyon Gashaw ETS1588/14

Section: E

Submitted to: Miss Eleni

Submission date: April 10, 2023

Introduction

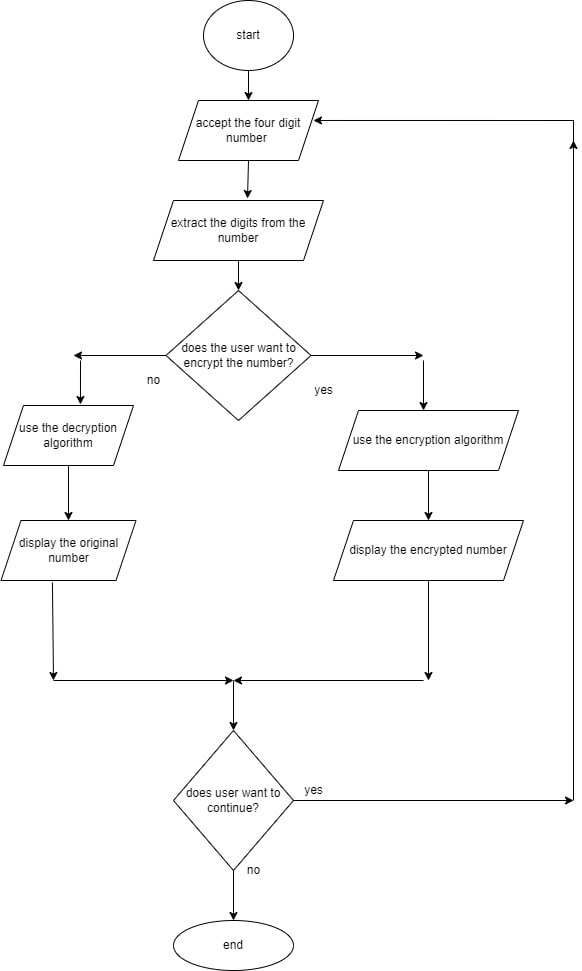
This document comprises of the pseudo codes and flowcharts of the questions given to us in the project. While we do the code in another file, we have found it necessary to include the pseudo codes and flowcharts of the questions, in order for us to understand a little bit of what we are doing, and to see a clear outline in how our actual code works. And since our code follows the flowcharts, we will be able to evaluate whether our code was written properly i.e according to the pseudo code and flowchart we described above. So without further we do, let’s get further into them.

Project I

The project is about a company wanting to transfer data by using four digit integers. And since entering the numbers directly might lead to their phones being tapped, there is a potential risk of data loss. Therefore, we need to encrypt the four digit numbers before they are sent, and after they get there, we need to decrypt the number to retain the original number. The plan is simple, actually. As the pseudo code suggests, the steps are the following.

1. Start the program.
2. Accept the number from the user.
3. Ask the user if he/she wants to encrypt or decrypt the number.
4. If yes, use an encryption algorithm to secure the data.
5. Display the encrypted number.
6. If no, use a decryption algorithm to retrieve the original number.
7. Display the original number.
8. End the program.

The flowchart is as follows.



Pattern 1

This pattern has a shape of a X -shape and we use letters from I to A to fill in the shape of the X shape figure.

The interesting thing here is that the letters increase up to some point, and they return back to where they started. For example, ABCBA, the letters increased from A to C, then as soon as C was displayed, it decreased back to A. This shows that the letters increase until the middle letter. Then they return back to their starting point i.e A.

The pseudo code of the program looks as follows:

Start

1. Initialize i with ‘H’

1.1) Check if i is greater than or equal to ‘A’

2) Initialize j with ‘A’

2.1) Check if j is less than or equal to ‘I’

Print j and then space

Update(increment) j by one

go to step 2.1 and check

Then assign the value of j to ‘x’

If step 2.1

is false go to condition 3

3) initialize character k with x minus one

3.1) Check if k is greater than or equal to ‘A’

If true

Print k and space

Then update by subtracting one from the value of ‘k’

Go to step 3.1 and check

If false

Print new line and update (decrease) the value of i with one

4) If variable l is equal to character ‘H’

4.1) Check if l is greater than or equal to ‘i’

And then print new line

Check condition 4.1

Update the the value of l( decrease by one)

If false

Go to the next statement

// the lower part of the pyramid

Print 3 backward space(\b)

1. Initialize character i with ‘A’

1.1) check if ‘i ‘ is less than or equal to ‘G’

1.1.1) initialize character ‘j’ with ‘A’

Check if ‘j’ is less than or equal to ‘i’ plus one

Print ‘j’ and space

Assign the value of ‘j’ to some variable y

Update the value of ‘j’ (increase by one)

Check if j is less than or equal to ‘i’ plus one

1.2) If false go to condition 2

2) initialize character ‘k’ with y minus one

2.1 check if ‘k’ is greater than or equal to ‘A’

If true

Print k and then space

Update by subtracting one from the value of ‘k’

If false

start with new line and move to condition 3

3) initialize character ‘m’ with ‘E’

3.1) check if ‘m’ is greater than or equal to ‘i’

If true Print space

Update the value of ‘m’ (decrease by one)

Check condition 3.1

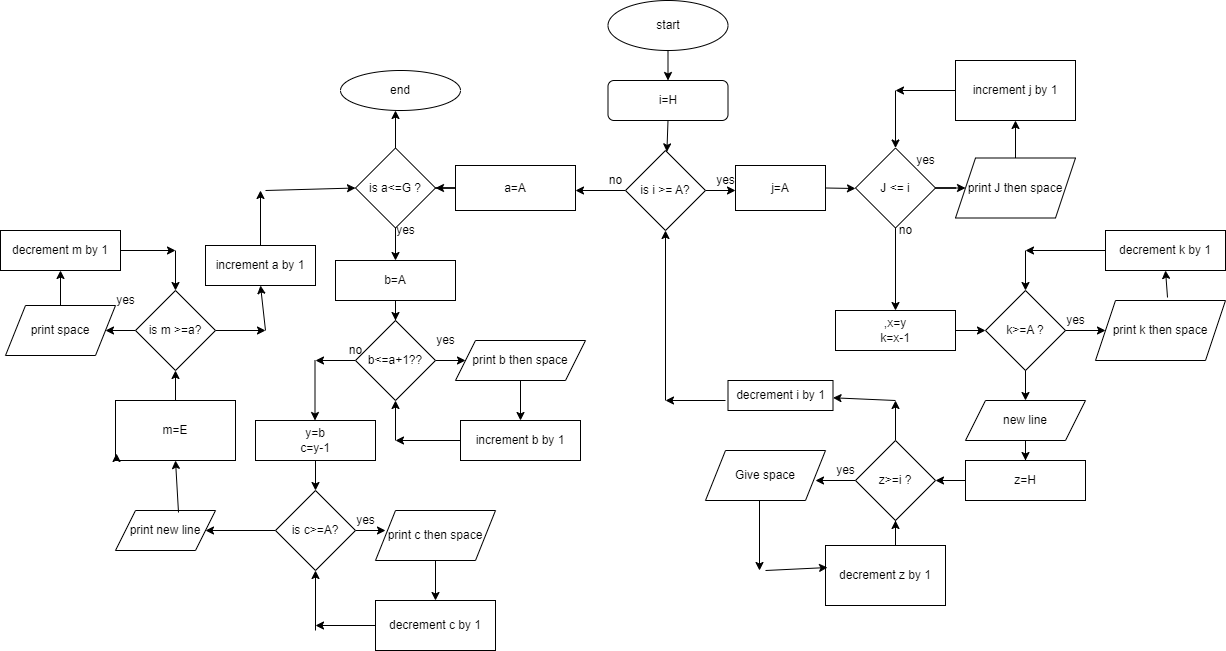
If false

Update the value of ‘i’ (increase by one)

And if condition 1.1 is false

End the program.

The flowchart is as follows:



Pattern 2

This pattern is very similar to a rhombus, but the key differences from a rhombus is that:

1. We shade the outside part of the rhombus. In other words, the shape of the rhombus is only visible by the blank spaces made by the stars we use to draw the body of the rhombus.
2. At some points of the rhombus, the symmetry isn’t the same. So it isn’t necessarily a rhombus, but a slight modification is done to it.

But since it is very similar to a rhombus, we can use some ideas from how to draw a rhombus using loops, and edit some parts to our code so that the shape given to us appears on the screen.

The pseudo code looks as follows:

Start

1. Initialize ‘i’ with H

1.1 check if ‘i’ less than or equal to 10

If ‘i’ is equal to one o0r ‘i’ is equal to two or ‘i’ equal to three

Assign k is equal to one

Print ‘\*’ as long as ‘k’ is less than or equal to 8

Update ‘k’ (increase by one)  
check if ‘k’ is less than or equal to 8

If false otherwise

2) Initialize ‘j’ is equal to one

2.1) Check if ‘j’ is less than or equal to 10 minus ‘i’ plus one

Print ‘\*’ and space

Then if ‘i’ is equal to 2

Print space

3.1) initialize i is equal to and check if ‘i’ is less than or equal to 2 multiplied by i-2

Print space

Assign the value of n to variable ‘b’

Update n(increase by one)

If condition 3.1 is false go to condition 4

4) initialize ‘m’ with one

Check if ‘m’ is less than or equal to 11 minus ‘i’

Print ‘8’ and space

Update m (increase by one)

go to condition 4.1 if its false start with ew line

// for the lower part

1. Initialize ‘z’ with one
2. Check if ‘z’ is less than or equal to 10

If ‘z’ is equal to 8 or ‘z’ is equal to 9

Initialize a variable ‘a’ with 1

Then print ‘\*’ and space as long as ‘a’ is less than or equal to 8

Update a and if false

Move to next statement

3) initialize x with one and check if x is less than z plus one

Print ‘\*’ and space

Update x

If false go to next

Initialize c with 2 multiplied by 9 minus (z+1)

Check if c >= 1

Update c(decrease by one)

Print space

If zis less than or equal to 8

Print space

Initialize d= 1

Check if d is <= z plus one

Print ‘\*’ and space

Update d(increase by 1)

If false

Update z (increase by 1)

If false print new line

End the program.

The flowchart is as follows:

