ADDIS ABABA SCIENCE AND TECHNOLOGY

UNIVERSITY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

Course: Fundamentals of programming 1

GROUP 3: Group Project

GROUP MEMEBERS

ID NO



TADIYOS DEJENE

ETS1522/14



YOHANNA BETSIHA ETS1701/14



YOHANNES TIGISTU ETS1703/14



YONATHAN ASHENAFI ETS1740/14



YONA

S WORKINEH ETS1725/14



YONATHAN ALAMREW

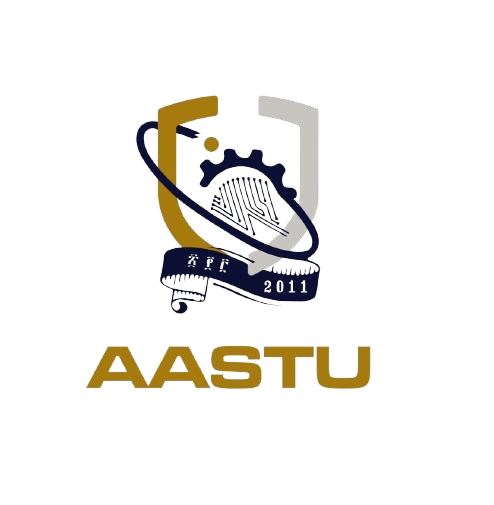
ETS1739/14



YOHANNES MESAY ETS1704/14

SUBMITTED TO:

Mrs. Eleni Teshome



Contents

[Introduction: 3](#_Toc132039887)

[Part I: 4](#_Toc132039888)

[Problem analysis 4](#_Toc132039889)

[Pseudo code 6](#_Toc132039890)

[Flowchart 8](#_Toc132039891)

[Problem analysis 9](#_Toc132039892)

[Pseudo code 10](#_Toc132039893)

[Flow chart 12](#_Toc132039894)

[PART II 13](#_Toc132039895)

[Conclusion: 14](#_Toc132039896)

# Introduction:

The objective of this assignment is to develop and demonstrate various skills related to problem solving, programming, modern development tools and teamwork. The assignment is divided into four objectives. The first objective requires analyzing a given problem description and developing a flow chart and pseudo code to demonstrate problem solving skills. The second objective involves designing, writing and debugging a small-scale computer program using basic programming constructs and good practices. The third objective requires the use of modern development and collaboration tools and techniques to demonstrate proficiency in these areas. Finally, the fourth objective emphasizes teamwork, collaboration, and presentation skills.

The codes found in this repository

https://github.com/yoni-clef/Group-project

Part I: **Problem Analysis, Program Design and Implementation Instruction: Analyze the problem given below, and (b) draw flow chart and write a pseudocode to describe the algorithm that you apply to solve the following problem and then transform the algorithms into program.**

8. okebe Tsibah secondary and preparatory prepare school closing function (gathering) and invite various concerned body. The people who attend the school function categorized in to four groups and given a code. Accordingly, 1 represents an infant, 2 represents a child, 3 represents a teenager, and 4 represents an adult. No other integer value should be accepted as valid input, and data entry should stop when a negative value is entered. Write an application that read entrance ticket number, person code as 1, 2, 3 or 4, and entrance fee. The fee schedule is 5ETB for child, 10ETB infant, and 20ETB for teenager and adults. The program should compute and display total numbers of people attend the function and total amount of ticket sales. Also the program should display a list of how many infants, children, teenagers, and adults were at the school function along with category codes. You should write the program without using array.

Codes for this problem

https://github.com/yoni-clef/Group-project/blob/main/kokebe%20Tsebah2.cpp

## Problem analysis

**Problem Statement:**

The program is designed to calculate the total number of attendees and ticket sales for an event at Kokebe Tsibah Secondary and Preparatory School. The program should prompt the user to enter a series of ticket numbers and person codes. The program should keep track of the unique ticket numbers entered and count the number of infants, children, teenagers, and adults attending the event based on the person codes entered. The program should calculate the total sales by adding up the entrance fees for each attendee. Finally, the program should output the total number of attendees, total sales, and the number of infants, children, teenagers, and adults attending the event.

**Input:**

The input to the program consists of a series of ticket numbers and person codes entered by the user. The ticket numbers are non-negative integers and the person codes are integers between 1 and 4 inclusive.

**Output:**

The output of the program should include the total number of attendees, total sales in Ethiopian Birr (ETB), and the number of infants (Code 1), children (Code 2), teenagers (Code 3), and adults (Code 4) attending the event.

**Solution Approach:**

To solve this problem, the program can use a set data structure to keep track of the unique ticket numbers entered by the user. When a new ticket number is entered, the program can check if it is already in the set. If it is not, the ticket number can be added to the set and the program can continue to process the ticket. If the ticket number is already in the set, an error message can be printed and the user can be prompted to enter a different ticket number.

After a valid ticket number has been entered, the program can prompt the user to enter a person code. The program can use a switch statement to determine the entrance fee based on the person code entered. The entrance fee for infants (Code 1) is 10 ETB, for children (Code 2) is 5 ETB, and for teenagers (Code 3) and adults (Code 4) is 20 ETB.

The program can use separate count variables to keep track of the number of infants, children, teenagers, and adults attending the event. After calculating the entrance fee for a ticket, the appropriate count variable can be incremented based on the person code entered.

The program can also keep track of the total number of attendees and total sales. After calculating the entrance fee for a ticket, the total attendees can be incremented by one and the entrance fee can be added to the total sales.

Finally, after all tickets have been processed, the program can output the total number of attendees, total sales in ETB, and the number of infants (Code 1), children (Code 2), teenagers (Code 3), and adults (Code 4) attending the event.

## Pseudo code

The pseudo code for the project is written below.

1.Initialize total\_attendees, total\_sales, infant\_count, child\_count, teenager\_count, adult\_count to 0

2.Initialize ticket\_numbers as an empty set

3.Print welcome message

4.Prompt user to enter ticket number

While ticket number is not negative:

If ticket number is already in ticket\_numbers:

Print error message

Else:

Add ticket number to ticket\_numbers

Prompt user to enter person code

While person code is not valid:

Print error message

Prompt user to enter person code

If person code is 1:

entrance\_fee = 10

Increment infant\_count

Else if person code is 2:

entrance\_fee = 5

Increment child\_count

Else if person code is 3 or 4:

entrance\_fee = 20

If person code is 3:

Increment teenager\_count

Else:

Increment adult\_count

Increment total\_attendees by 1

Add entrance\_fee to total\_sales

5.Prompt user to enter next ticket number

6.Print total number of attendees

7.Print total sales

8.Print number of infants

9.Print number of children

10.Print number of teenagers

11.Print number of adults

## owchart

**9.** A prime number is an integer greater than one and divisible only by itself and one. Write an application that reads an integer number N and check if it is prime number or not. If the number is prime then the program prints out the average value of prime numbers between 1 and N. Otherwise the program should find and prints out all prime factors of inputted number N. (Hint: 1 is a prime number. For each number from greater than or equal to 2, find Remainder = N % k, where k ranges from 2 to sqrt(N). If k is greater than sqrt(number), the number is not equally divisible by k. And if any Remainder equals 0, the number is not a prime number.

The correct code found here

https://github.com/yoni-clef/Group-project/blob/main/primecheck.cpp

# Problem analysis

Problem: Check if an integer N is prime or not. If N is prime, find the average value of all prime numbers between 1 and N. If N is not prime, find and print all prime factors of N.

Input: An integer number N.

Output: The average value of prime numbers between 1 and N if N is prime. All prime factors of N if N is not prime.

Approach:

Check if N is a prime number or not by iterating over all numbers from 2 to sqrt(N) and checking if N is divisible by any of them.

If N is prime, calculate the average value of all prime numbers between 1 and N by iterating over all numbers from 1 to N and checking if each number is prime. If a number is prime, add it to a running total and count the number of prime numbers found. The average value can then be calculated by dividing the running total by the number of primes found.

If N is not prime, find all prime factors of N by dividing N by all numbers from 2 to sqrt(N) that divide N without leaving a remainder. Add each factor to a list of prime factors and continue until N is no longer divisible by any factor, at which point N will either be 1 or a prime number itself.

## Pseudo code

1. Start

2. Read an integer number N from user

3. Define a boolean function isPrime(N) to check whether N is prime or not:

a. If N is less than or equal to 1, return false

b. For each integer i from 2 to sqrt(N), check whether N is divisible by i

c. If N is divisible by i, return false

d. Otherwise, return true

4. Define a function printPrimeFactors(N) to print all prime factors of N:

a. Initialize a loop variable i to 2

b. While N is greater than 1:

i. If i is a factor of N, print i and divide N by i

ii. Otherwise, increment i

5. If isPrime(N) returns true, do the following:

a. Initialize a variable sum to 0 and a variable count to 0

b. For each integer i from 1 to N:

i. If isPrime(i) returns true, add i to sum and increment count

c. Calculate the average value of prime numbers between 1 and N as sum/count

d. Print the average value

6. Otherwise, print all prime factors of N using the printPrimeFactors(N) function

7. End

## Flow chart

PART II**: Pattern Printing Write a program that accepts a positive integer, n, or a letter entered by the user and prints the following shape. But if the value entered is less than one, the program prints nothing.**

**h. the program will be found in the following repository**

**https://github.com/yoni-clef/Group-project/blob/main/Pattern%20Main.cpp**

# Conclusion:

In conclusion, this assignment serves as an excellent opportunity to develop and showcase a range of essential skills required in today's world. The first objective enables the students to demonstrate their problem-solving abilities by analyzing a given problem description and developing a flow chart and pseudo code. The second objective emphasizes the importance of good programming practices and the ability to design, write and debug a small-scale computer program. The third objective focuses on the use of modern development and collaboration tools and techniques, essential skills that are necessary in today's fast-paced world. Finally, the fourth objective underscores the importance of teamwork, collaboration, and presentation skills, which are essential for success in any field. Overall, the successful completion of these objectives will help students prepare for future challenges in their academic and professional careers.