CELESTIAL CODEX REPORT

ABSTRACT

The following document contains a detailed report of the works carried out by Celestial Codex members. The aim was to develop solutions to the given challenges using the C language. The group successfully achieved this by leveraging various control structures and programming techniques.

The group members are as follows:

* Benedict Xavier
* Gabriel Opiyo
* Debra Gitonga
* Victor Gachogu
* Brian Ireri
* Owen Ouna
* Tracy
* Prince Michael
* Felix Mwangi
* Bongo Sydney

# Introduction

## Objective

This report is objectively written to promote team work ensuring each solution is well-reasoned thus avoiding any loopholes and that the results for the test data are accurate and properly documented to enable even new users get a gist of the programs before working on them

## Scope

It covers the expectations for all four questions given in the assignment as well as their solutions.

# Question One

## Problem Statement

A program is required for a computer game. The user keys in the number of rounds he wishes to play. For each round the user enters his lucky number. The program takes the number and divides it with a secret number. If the remainder of the division is zero, it is considered a draw for the round and the total score is incriminated by 1. Otherwise, if it is any other even number, it is considered a win for the round and the total score is incremented by 3. However, if it is an odd number, it is considered a loss for the round and the total score is decremented by 3. This is done until he completes his rounds. He wins if the total score at the end is a positive number. Write a C program to accomplish this.

## Problem Analysis

The program is for a computer game. A lucky number (by user) is divided by a secret number (by program) and results awarded accordingly. This continues the number of rounds specified by user at the beginning. User with positive results at the end wins else loses.

Inputs: Number of rounds; Lucky number for each round

Process: Division of lucky number by secret number; Looping rounds; Awarding points accordingly; Checking final results.

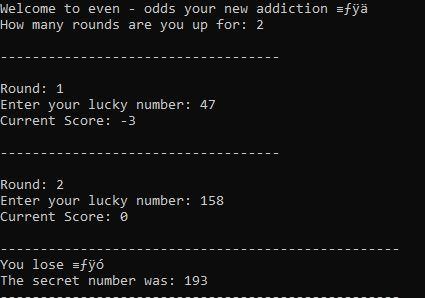
Output: Round result; Final result.

## Testing and Results

The test data is as follows:

* Number of rounds – 2
* Lucky number round 1 – 47
* Lucky number round 2 – 158

The results of the program are as follows:



Discussion: The remainder of the first division is 47, 47 is an odd number thus the score awarded is (-3) but the remainder of the second division is 158 which is an even number thus the score awarded is (+3). The final score is 0 which is neither positive nor negative thus a loss.

## Learning Outcomes

From the question in discussion, the group learnt about seeding.

Seeding refers to the process of initializing the random number generator (for the secret number) with a starting value called a “seed”. This ensures that a different secret number is generated each time the program runs thus logically efficient as the game is made free from predictions

# Question Two

## Problem Statement

Write a program that accepts the student admission number, student surname and three subject marks and displays the same along with the student average mark. The program should also display the student grade depending on the average marks the student obtains. The grading system is shown in the table below

| Average Mark | Grade |
| --- | --- |
| 70-100 | A |
| 60-69 | B |
| 50-59 | C |
| 40-49 | D |
| 0-39 | FAIL |

The program should also prompt the user if to continue. If the user enters “Y”, for yes then program computes for another student and “N”, for no then program exits

## Problem Analysis

The program is for a school grading system. The sum of the subject marks is computed and an average is calculated. The program then grades the student according to the average mark. The grade, average mark and input are displayed. This continues for as long as the user decides.

Input: Admission number; Surname; Subject 1 mark; Subject 2 mark; Subject 3 mark.

Process: Computing sum of the subject marks; Finding the average.

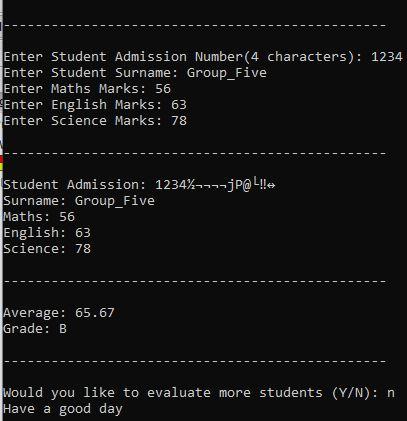
Output: Average mark along with the ***input***

## Testing and Results

The test data is as follows

* Admission number – 1234
* Surname – Group\_Five
* Subject 1 mark – 56
* Subject 2 mark – 63
* Subject 3 mark – 78

The results of the program are as follows



## Learning Outcomes

From the question in discussion, the group learnt how to tie a choice to a loop. This ensures the program continues as long as the choice is y for yes and terminates the loop if the choice is n for no.

# Question Three

## Problem Statement

A computer program is required to evaluate the credit worthy of a client. The program reads the credit limit, price and quantity of the item to be purchased by the client. If the value of the goods is more than the credit limit, the program displays “Sorry you cannot purchase goods worthy such a value on credit” and allows the customer to re-enter the quantity, otherwise, displays “Thank You for purchasing from us” and the value of the purchase. This should be repeated for n customers.

## Problem Analysis

This is a program that compares one’s credit limit to the value of items on purchase then determines whether the customer is eligible to purchase on credit. An appropriate output is then displayed according to the results. If purchase is not successful, the program allows the user to enter a new quantity. The program repeats this for n number of customers.

Input: Credit limit; Quantity of items; Price per item; Number of customers.

Process: Product of quantity and price per item; Comparison between credit limit and value of purchase.

Output: “Sorry you cannot purchase goods worthy such a value on credit” or “Thank You for purchasing from us”.

## Flowchart

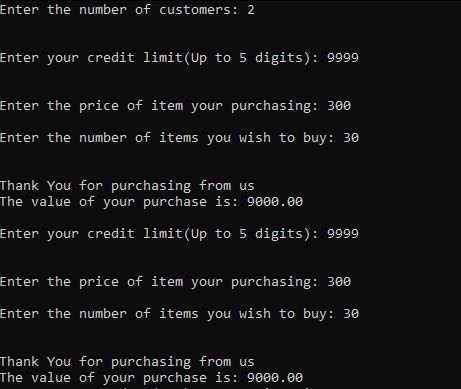
## Testing and Results

The test data is as follows

Case One:

* Credit limit – Ksh 9999
* Quantity of items – 300
* Price per item – Ksh 30
* Number of customers – 2

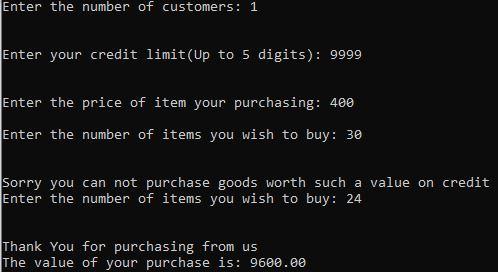
The result of case one is as follows



Case Two:

* Credit limit – Ksh 9999
* Quantity of items – 400
* Price per item – Ksh 30
* Number of customers – 1

The result of case two is as follows



## Learning Outcomes

From the question in discussion, the group learnt how to limit the number of digits one can input in the program using the strlen and strspn functions under the string.h C library.

# Question Four

## Problem Statement

Study the following table used to compute the tax payable by employees in certain organization

| Gross Pay | Fewer than three dependants | Three dependants or more |
| --- | --- | --- |
| Ksh 10,000 or less | Tax rate = 0 | Tax rate = 0 |
| More than KSh10, 000 and less than or equal to KSh20, 000 | Tax rate = 15% | Tax rate = 10% |
| Over KSh20, 000 | Tax rate = 35% | Tax rate = 25% |

A program is required to input the gross pay and number of dependants of an employee and then computes the tax payable and net pay. The program should output gross pay, tax payable and the net pay of an employee in a suitable format.

**Hint:**

**Tax payable = Gross pay \* Tax rate**

**Net pay = Gross pay- Tax payable**

## Problem Analysis

This is a program that calculates individual tax payable according to various factors. It reads the gross pay and number of dependents then outputs the same with the net pay.

Input: Gross pay; Number of dependants.

Process: Setting tax rate depending on gross pay and number of dependants; Calculating tax payable.

Output: Gross pay; Number of dependants.

## Flowchart

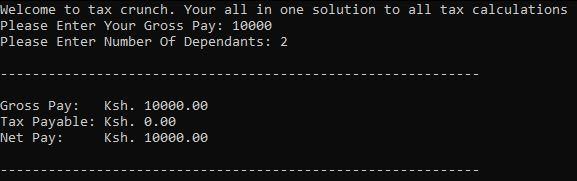
## Testing and Results

The test data is as follows

Case One

* Gross pay – Ksh 10,000
* Number of dependants – 2

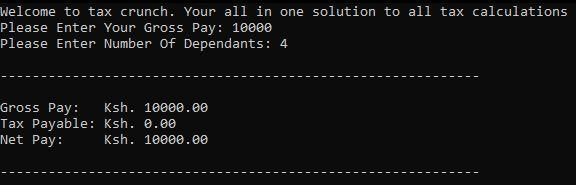
The result of Case One is as follows



Case Two

* Gross pay – Ksh 10,000
* Number of dependants – 4

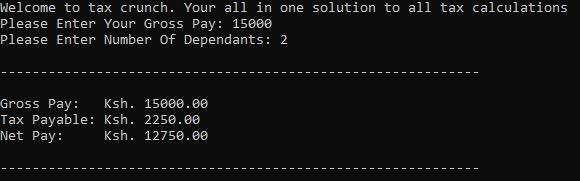
The result of Case Two is as follows



Case Three

* Gross pay – Ksh 15,000
* Number of dependants – 2

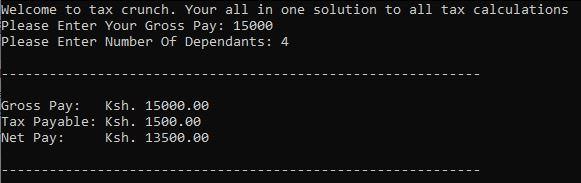
The result of Case Three is as follows



Case Four

* Gross pay – Ksh 15,000
* Number of dependants – 4

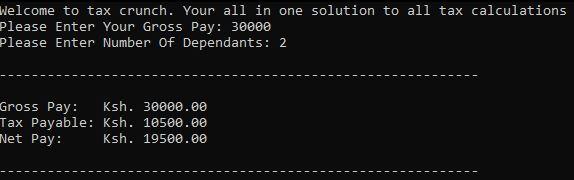
The result of Case Four is as follows



Case Five

* Gross pay – Ksh 30,000
* Number of dependants – 2

The result of Case Five is as follows



Case Six

* Gross pay – Ksh 30,000
* Number of dependants – 4

The result of Case Six is as follows

