

**CSC126 : FUNDAMENTALS OF ALGORITHMS AND**

**COMPUTER PROBLEM SOLVING**

**REPORT**

|  |  |
| --- | --- |
| PROJECT TITLE | G BURGERS |

|  |
| --- |
| GROUP MEMBERS : |
| MUHAMMAD AMIR BIN KHAIRUL NIZAR (2022832564) |
| MUHAMMAD IRFAN SYAFIQ BIN ZAIDI (2022488964) |
| AHMAD AIMAN ARIF BIN AHMAD ALIAS (2022881236) |

Table of Content

1. Executive Summary
   1. Project Background
   2. Objectives
   3. Description

3.0 Pseudocode & Flowchart

4.0 Source code

5.0 Sample Output

6.0 Reference

**Executive Summary**

* 1. **Project Background**

The owner of a G burgers is facing a problem of keeping track of their daily stock of buns, eggs, and meat. They need to keep updating their available stock regularly to ensure they never run out of ingredients. At the same time, their customers are eager to know what the current stock is and what items are available on the menu. To solve this problem, the owner can consider implementing a computerized system to manage their inventory and sales. The system can keep track of the daily inflow and outflow of ingredients and provide real-time information on the available stock. The owner can also use the system to record the total sales for each day and analyse the data to make informed decisions about inventory management. Additionally, the owner can set up a display screen or a website where customers can view the current stock status and the menu items that are available. This will not only provide transparency to customers but also help the owner to manage their stock efficiently. By implementing these changes, the burger stall owner can streamline their operations and provide a better experience for their customers.

1.2 **Objectives**

1. Keep track of the available stock of buns, eggs, and meat on a daily basis and ensure that the ingredients are always in sufficient quantity.

2. Provide real-time information to customers on the current stock status and available menu items.

3.Streamline the sales and inventory management process by using a computerized system.

4.Record and analyse the daily sales data to make informed decisions about inventory management.

5.Provide a better experience for customers by ensuring the availability of ingredients and transparent information about the current stock status.

**Description**

Welcome to the world of burger stall management “The G Burgers”! As the owner of a busy burger stall, it's important to keep track of the daily inflow and outflow of ingredients such as buns, eggs, and meat. Keeping the customers informed about the current stock status and available menu items is also crucial to providing a good experience. In this scenario, we will explore how to overcome the challenges of managing inventory and sales, and how to provide a seamless experience for customers. By implementing a computerized system and keeping track of daily sales data, we can streamline the operations and ensure that the burger stall is always fully stocked and ready to serve its customers. We are proud to announce that our burger stall now offers a special burger with an egg! This delicious addition is sure to satisfy even the biggest appetites and is sure to become a customer favourite. And for those who are looking for a more classic option, our standard burger is still available and reasonably priced. With our juicy, succulent beef patty and soft, fluffy buns, our standard burger is a classic staple that never fails to impress. So come on down and try one today! Whether you're in the mood for a classic burger or a new twist on an old favourite, our menu has something for everyone. And with our affordable prices, you can enjoy a delicious meal without breaking the breaking the bank. So come on in and treat yourself to a burger today!

**3.0 Pseudocode**

**Design Algorithm (Pseudocode)**

**Psuedocode(Main)**

Start

const int maxItem=4

int count=0 , stocks[maxItem] , burgerType

double price , taxAmount , totalPrice , totalSales

const double tax = 0.06

char meatOptn

string burgerName , stockName[maxItem] = "Buns = " , "Chicken Meat = " , "Beef Meat = " , "Egg = "

ownerStock()

char repeatOrder='Y'

WHILE ( repeatOrder == ‘Y’ || repeatOrder ‘y’)

IF stocks [0]>0

IF (stocks [1]>0 || stocks [2]>0)

menuNstock()

brgOptn()

burgerName=burgerNaming()

priceCalc()

custBill()

DISPLAY “Make a new order? (Y to proceed):”

INPUT repetOrder

ELSE

repeatOrder=’N’

DISPLAY “Sorry! Our buns ran out of stock”

DISPLAY “Thank You Come Again!”

ELSE

DISPLAY “Sorry! Our buns ran out of stock”

DISPLAY “Thank You Come Again!”

DISPLAY ENDLINE

grossNstock()

END WHILE

END IF ELSE

END

**Psuedocode\_(ownerStock)**

ownerStock()

char stockInput='Y'

DISPLAY "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

DISPLAY “Good Morning Owner!”

WHILE (stockInput=='Y' || stockInput=='y')

DISPLAY "Please enter todays stock:"

count=0

FOR (count ; count<maxItem ; count++)

DISPLAY stockName[count]

INPUT stocks[count]

DISPLAY ENDLINE

IF (stocks[0] <0)

DISPLAY "INVALID Buns Stock!"

IF (stocks[1] <0)

DISPLAY "INVALID Chicken Meat Stock!"

IF (stocks[2] <0)

DISPLAY "INVALID Beef Meat Stock!"

IF (stocks[3] <0)

DISPLAY "INVALID Egg Stocks!"

IF (stocks[0]>0 && (stocks[1]>0 || stocks[2]>0))

DISPLAY "Want to re-enter stocks? (Y to proceed) : "

INPUT stockInput

DISPLAY “Happy Selling!”

DISPLAY "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

**Pseudocode(menuNstock)**

void menuNstock()

DISPLAY "==================================================="

DISPLAY “MENU”

DISPLAY "Chicken Burger (RM3.50) Stocks:"

IF (stocks[1]>0)

DISPLAY stocks[1]

ELSE

DISPLAY "OUT OF STOCK"

DISPLAY "Beef Burger (RM4.50) Stocks:"

IF (stocks[2]>0)

DISPLAY stocks[2]

ELSE

DISPLAY "OUT OF STOCK"

DISPLAY "~ Egg add-ons (RM1.00) Stocks:"

IF "~ Egg add-ons (RM1.00) Stocks:"

IF (stocks[3]>0)

DISPLAY stocks[3]

ELSE

DISPLAY "OUT OF STOCK"

DISPLAY "==================================================="

DISPLAY “----ORDER----"

DISPLAY "1) Special (Egg add-ons)"

DISPLAY "2) Standard "

**Pseudocode\_brgOptn()**

void brgOptn()

Price =0

DISPLAY "Enter Burger Type (1/2) : "

INPUT burgerType

DISPLAY ENDLINE

SWITCH (burgerType)

case 1:

stocks[3]--

price++

case 2:

meatSelector()

BREAK

DEFAULT :

DISPLAY "INVALID BURGER TYPE"

brgrOptn()

**Pseudoce\_meatSelector**

Void meatSelector

DISPLAY "Enter Burger Meat (C/B) : "

INPUT meatOptn

IF (meatOptn=='C' || meatOptn=='c')

stocks[0]--

stocks[1]--

price=price+3.5

ELSE IF (meatOptn=='B' || meatOptn=='b')

stocks[0]--

stocks[2]--

price=price+4.5

ELSE

price=0;

DISPLAY “INVALID BURGER MEAT";

DISPLAY ENDLINE

meatSelector();

string burgerNaming()

IF (burgerType==1)

IF (meatOptn=='C' || meatOptn=='c')

RETURN "Special Chicken Burger"

ELSE IF (meatOptn=='B' || meatOptn=='b')

RETURN "Special Beef Burger"

ELSE IF (burgerType==2)

IF (meatOptn=='C' || meatOptn=='c')

RETURN "Chicken Burger"

ELSE IF (meatOptn=='B' || meatOptn=='b')

RETURN "Beef Burger"

**Pseudocode\_priceCalc()**

void priceCalc()

taxAmount=price\*tax

totalPrice=price+taxAmount

totalSales=totalSales+totalPrice

**Pseudocode\_custBill()**

void custBill()

DISPLAY "###################################################"

DISPLAY “Bills"

DISPLAY burgerName

DISPLAY "Price :RM ", price

DISPLAY "Tax(6%) :RM ", taxAmount

DISPLAY "Total Price :RM ", totalPrice

DISPLAY "###################################################"

Pseudocode\_grossNstock()

DISPLAY "---------------------------------------------------"

DISPLAY “-Todays Statistic|-"

DISPLAY "Total Sales: RM", totalSales

DISPLAY "Stocks Left:"

Count=0

FOR (count ; count<maxItem ; count++)

DISPLAY stockName[count]

DISPLAY stocks[count]

DISPLAY "---------------------------------------------------"

**Chart

Description automatically generated3.0 FLOWCHARTChart, diagram

Description automatically generated**

**Chart, funnel chart

Description automatically generatedChart, diagram, funnel chart

Description automatically generatedChart, diagram

Description automatically generatedDiagram

Description automatically generatedTimeline

Description automatically generatedTimeline

Description automatically generatedDiagram

Description automatically generated**

**4.0 SOURCE CODE**

**6.0 REFERENCE**