

Final Project Report
Online Shop Management

Created by

Phakapol Maneesopa 65070503459

Thanaphat Ngoennet 65070503449

Thanakorn Soonjaw 65070503450

Parunchai Kochseni 65070503456

Kanchai Lerdksrisakulrat 65070503475

CPE100: Computer Programming

Semester 01/2022

King Mongkut's University of Technology Thonburi

Final Project Report
Online Shop Management

Created by
Phakhapol Maneesopa 65070503459
Thanaphat Ngoennet 65070503449
Thanakorn Soonjaw 65070503450
Parunchai Kochseni 65070503456
Kanchai Lerdsrisakulrat 65070503475

Summit to
Dr. Natasha
CPE100: Computer Programming
Semester 01/2022

King Mongkut's University of Technology Thonburi

Abstract

This project aims to give clients of physical stores the benefits of internet shopping. Any store in the neighborhood or international brands with retail outlet chains can use this approach. If retailers offer a website through which customers may easily shop from any location. Using an android mobile to shop for goods online from any location is helpful. As a result, the customer will receive his favorite store's online purchasing and delivery services. Since the program is accessible through a smartphone and is constantly available, the shops won't be losing any more clients to the popular online retailers.

Acknowledgement

This undertaking is over. We really appreciate the organizing committee's assistance in making the task go well. I appreciate you sharing your understanding about the project assembly, Dr. Natasha. I'd like to express my gratitude to the assistant professors for their project assistance and suggestions for fixing flaws.

We would like to thank all of our friends who helped us test the program and provide feedback to the developers so that we could fix and enhance it for ourselves.

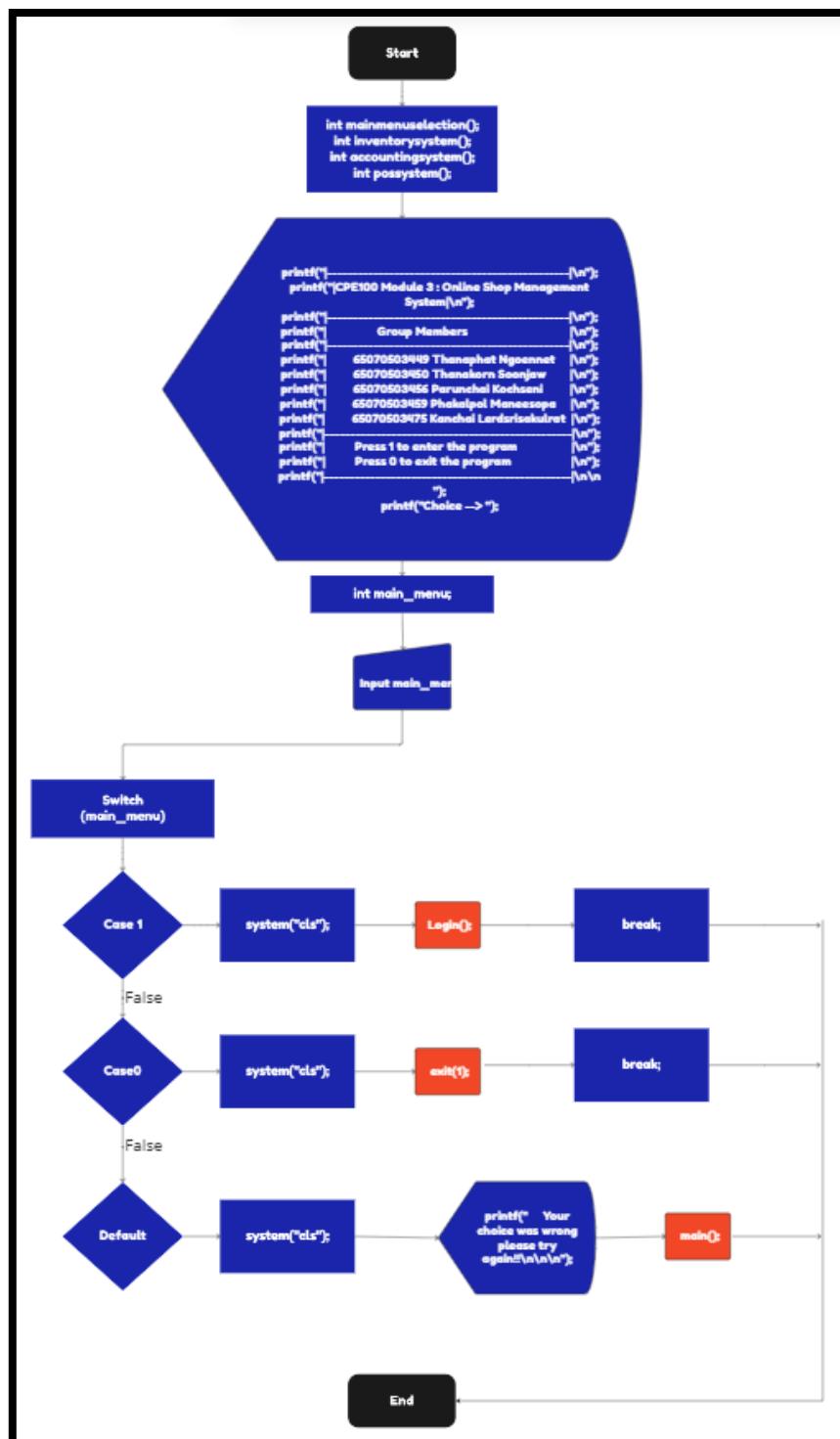
We can create our program more quickly and effectively because of the fantastic program that Visual studio code's developers have created.

Table Of Contents

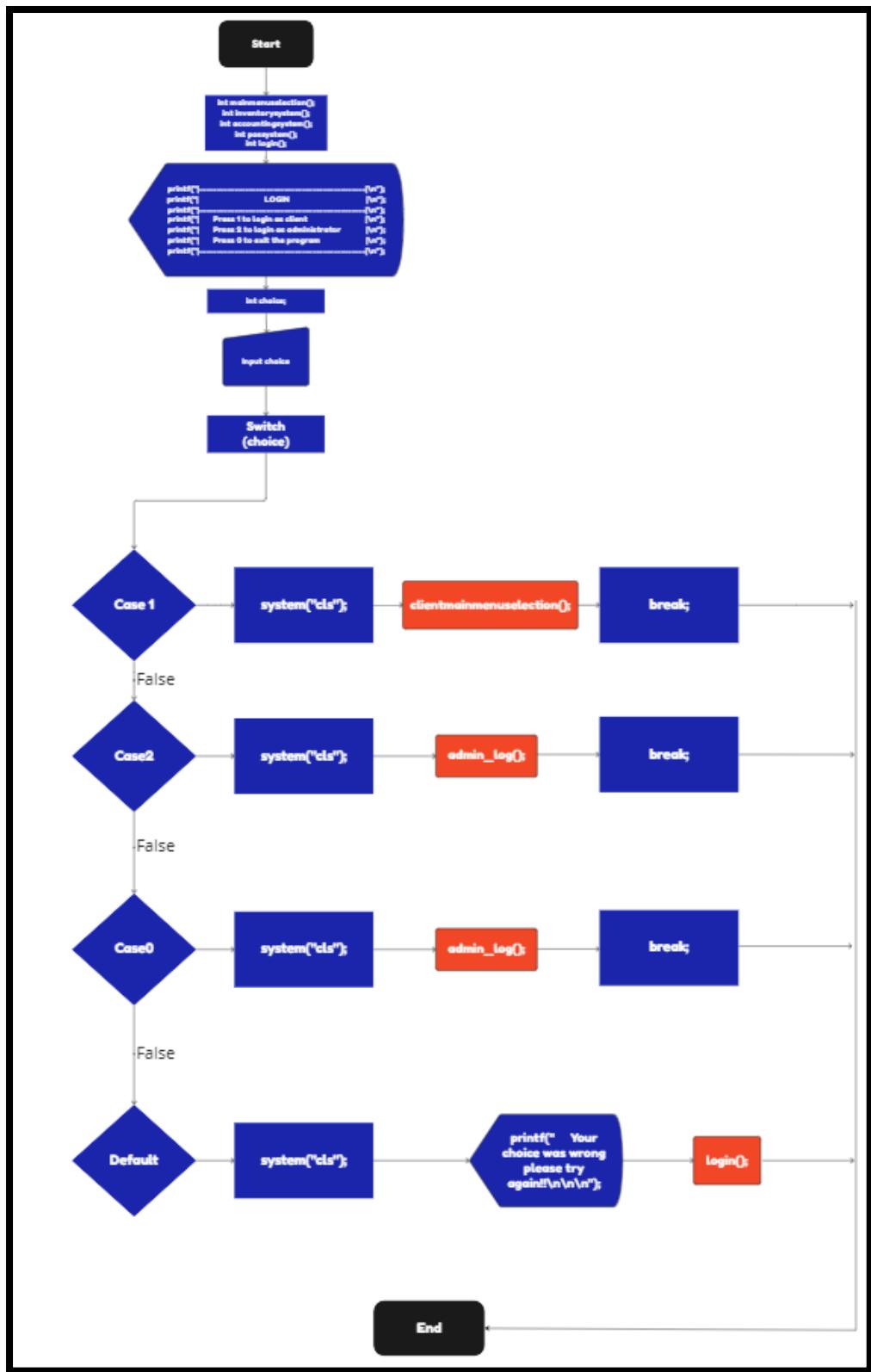
Contents	Pages
Flowchart & Operation of Program	6
How program work?	64
Function	68
Workload	104

Section 1

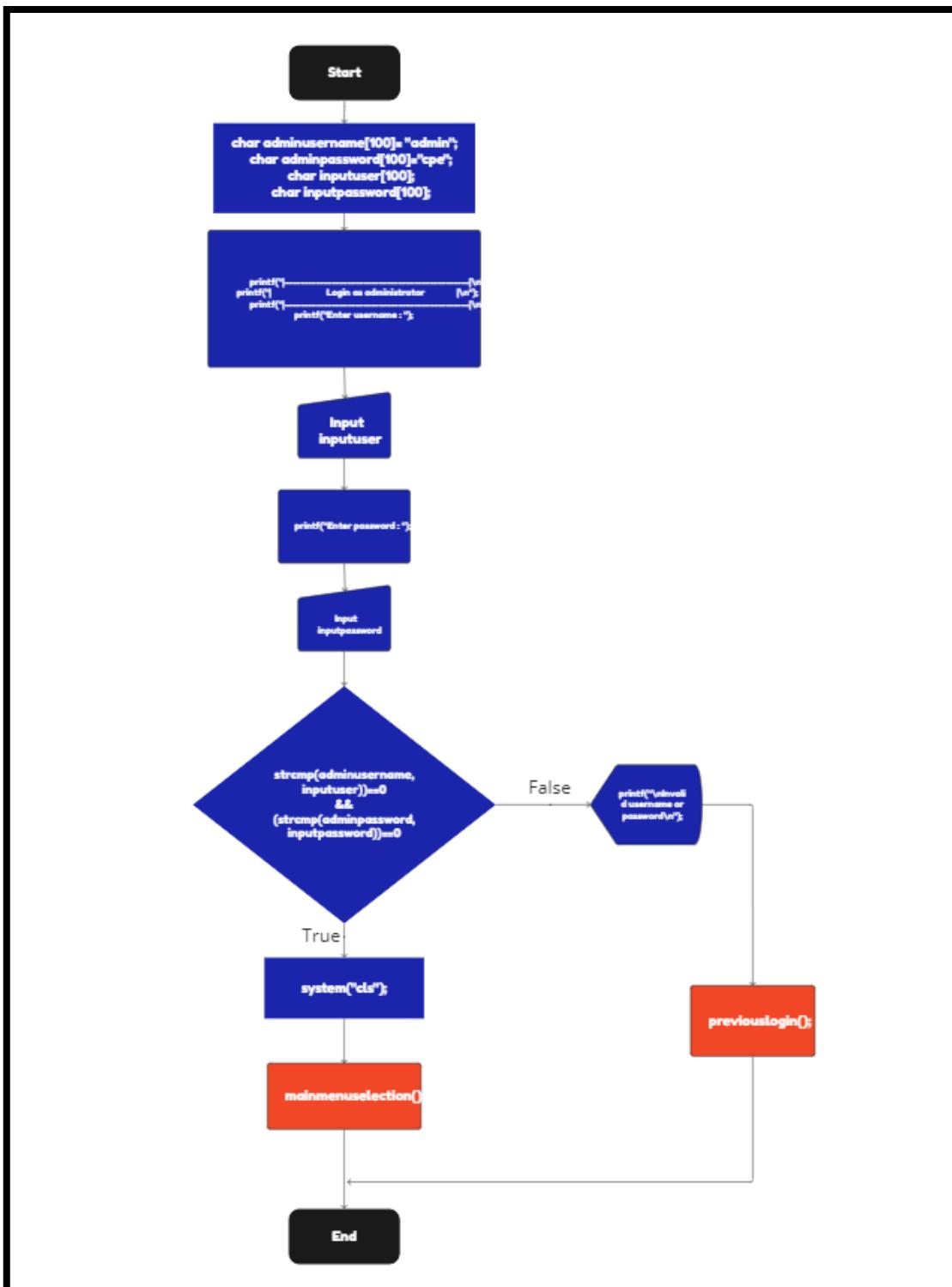
Flowchart & Operation of Program



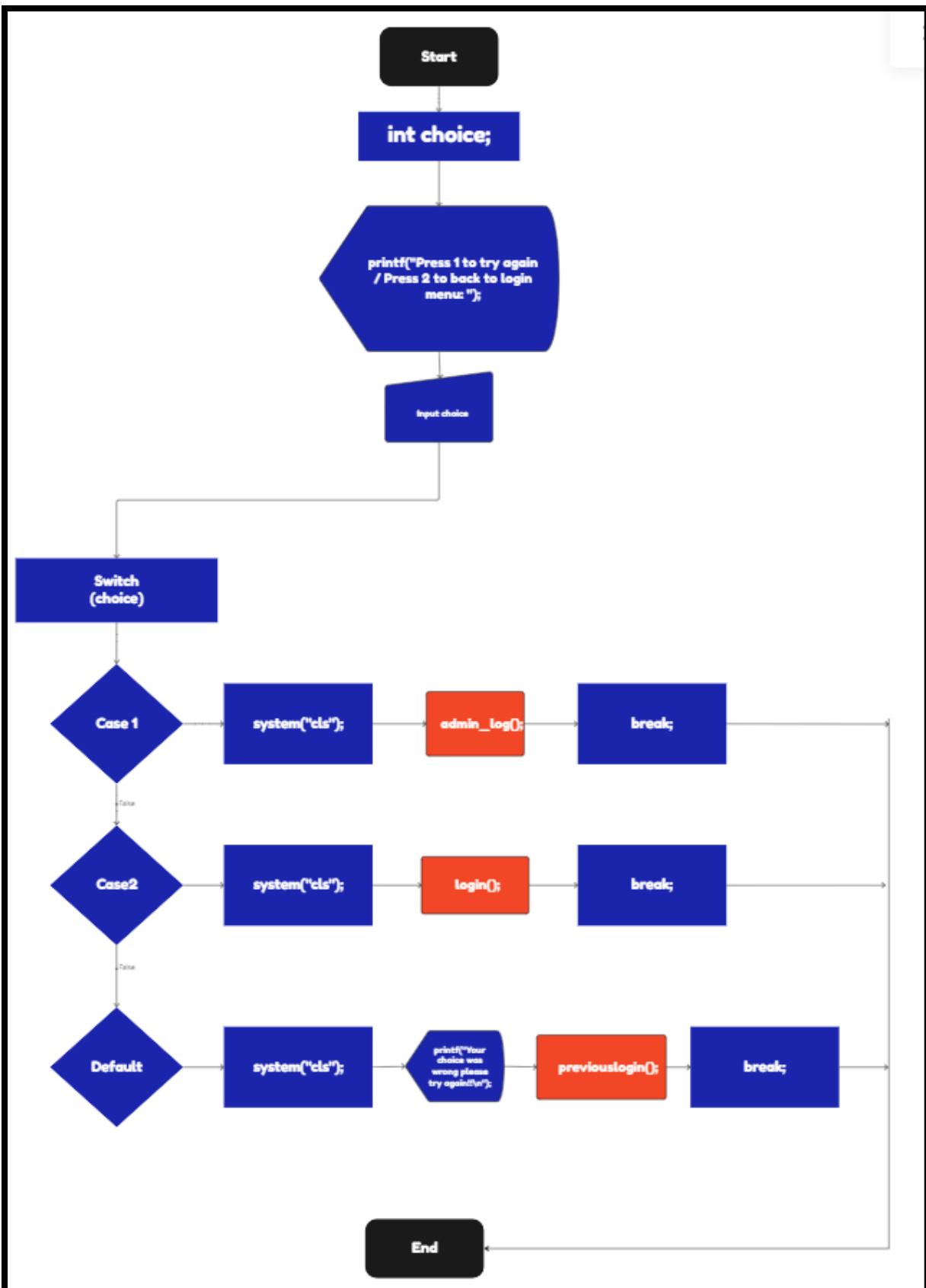
Int main



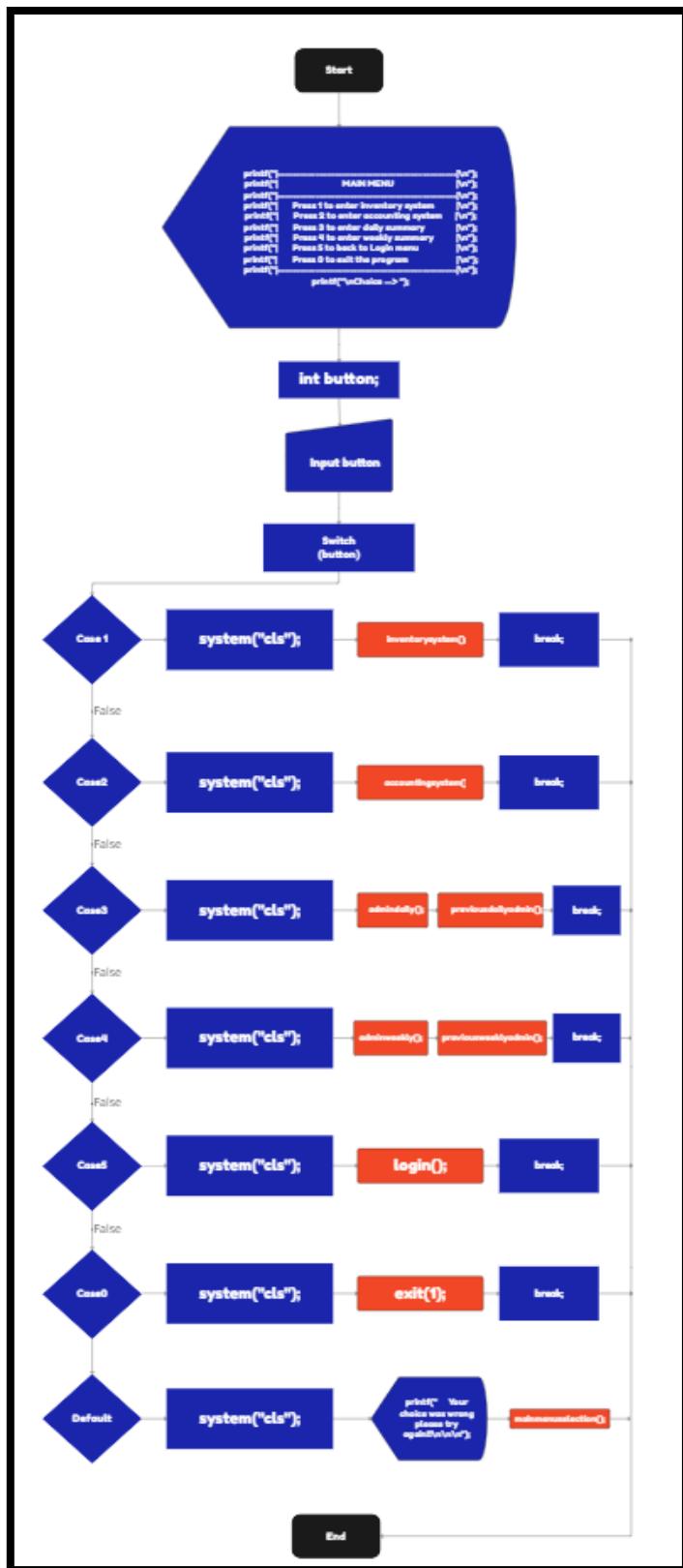
Int login



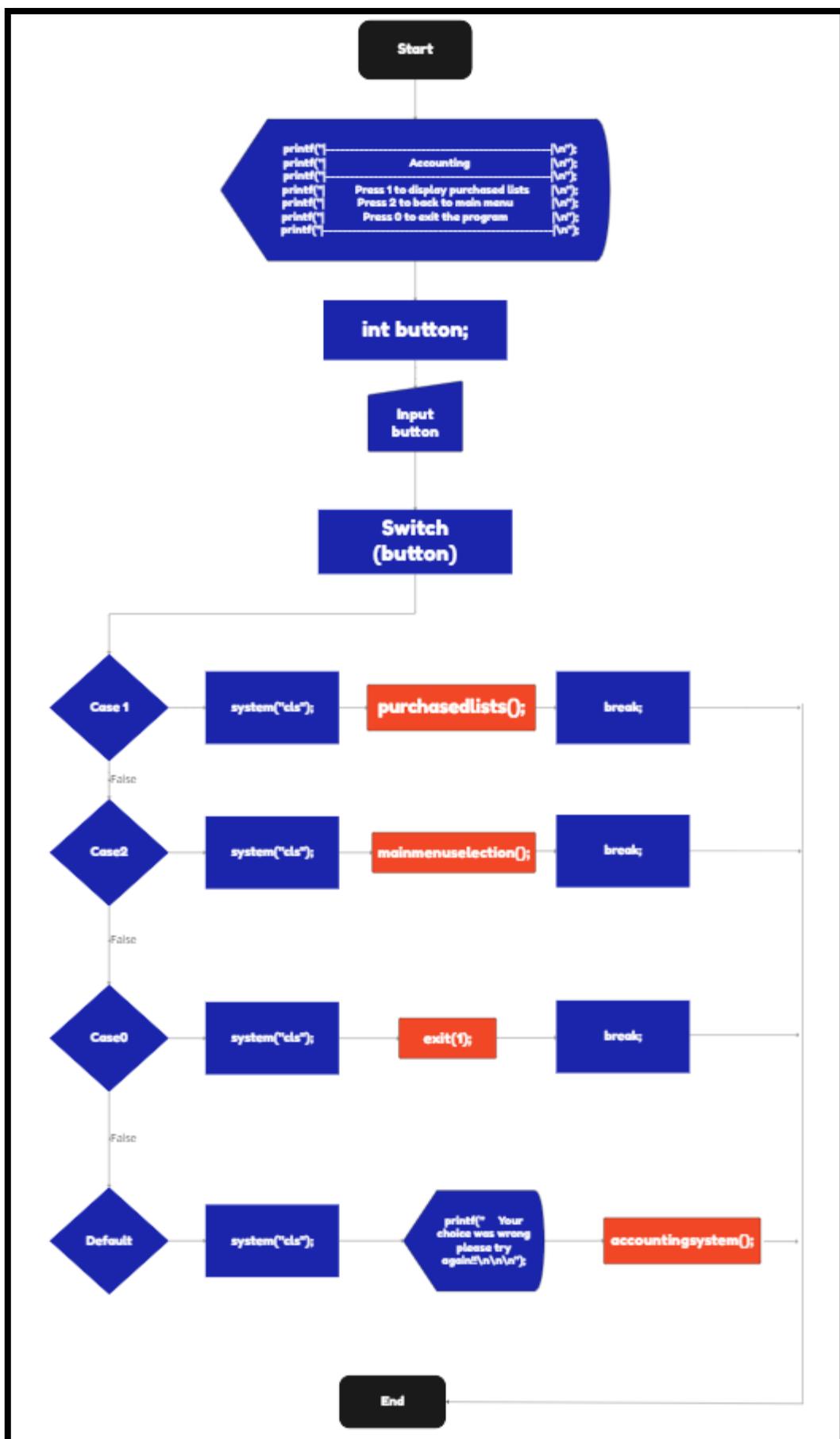
Void Admin_log



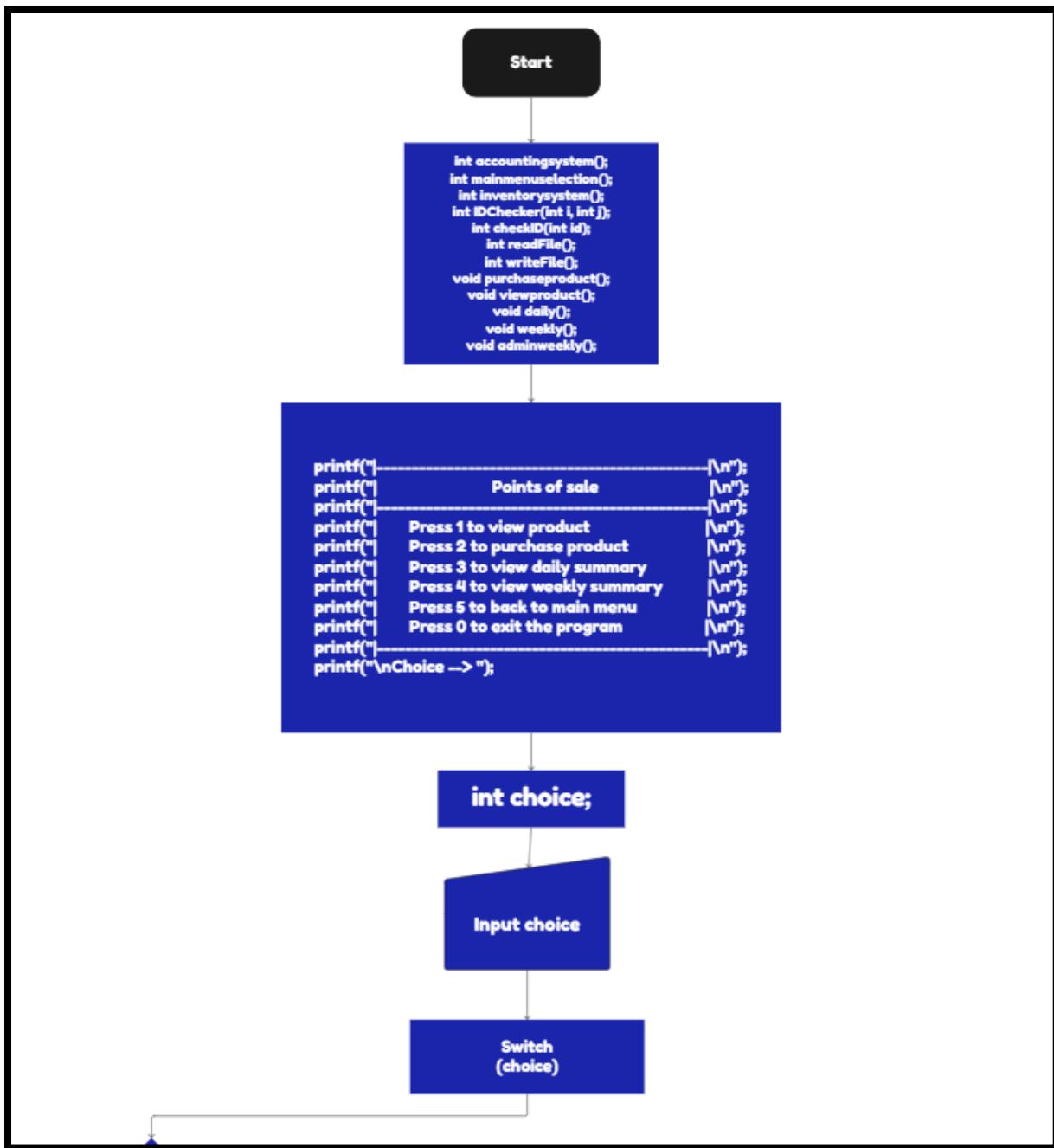
Void PreviousLogin



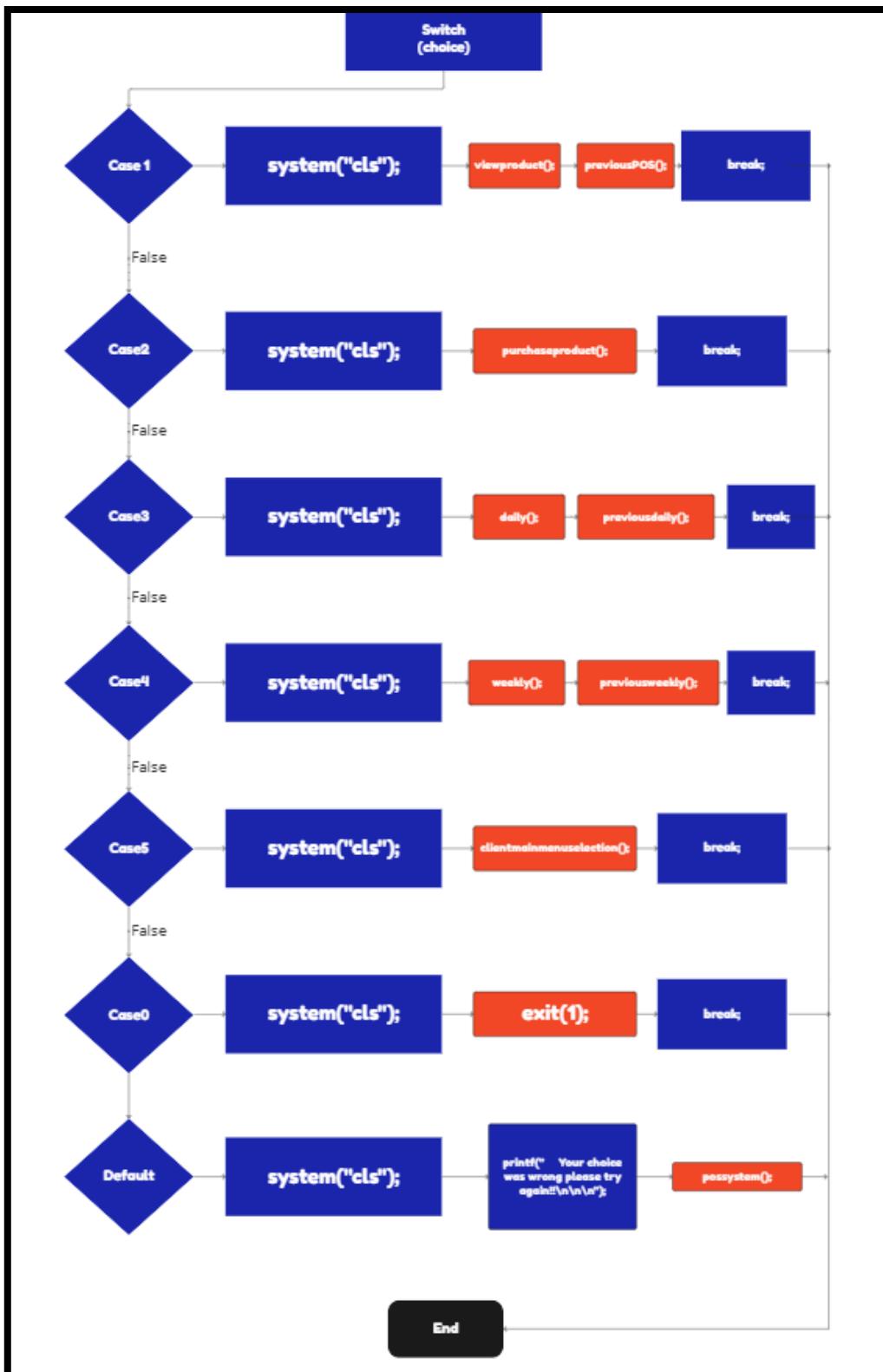
Int MainMenuSelection



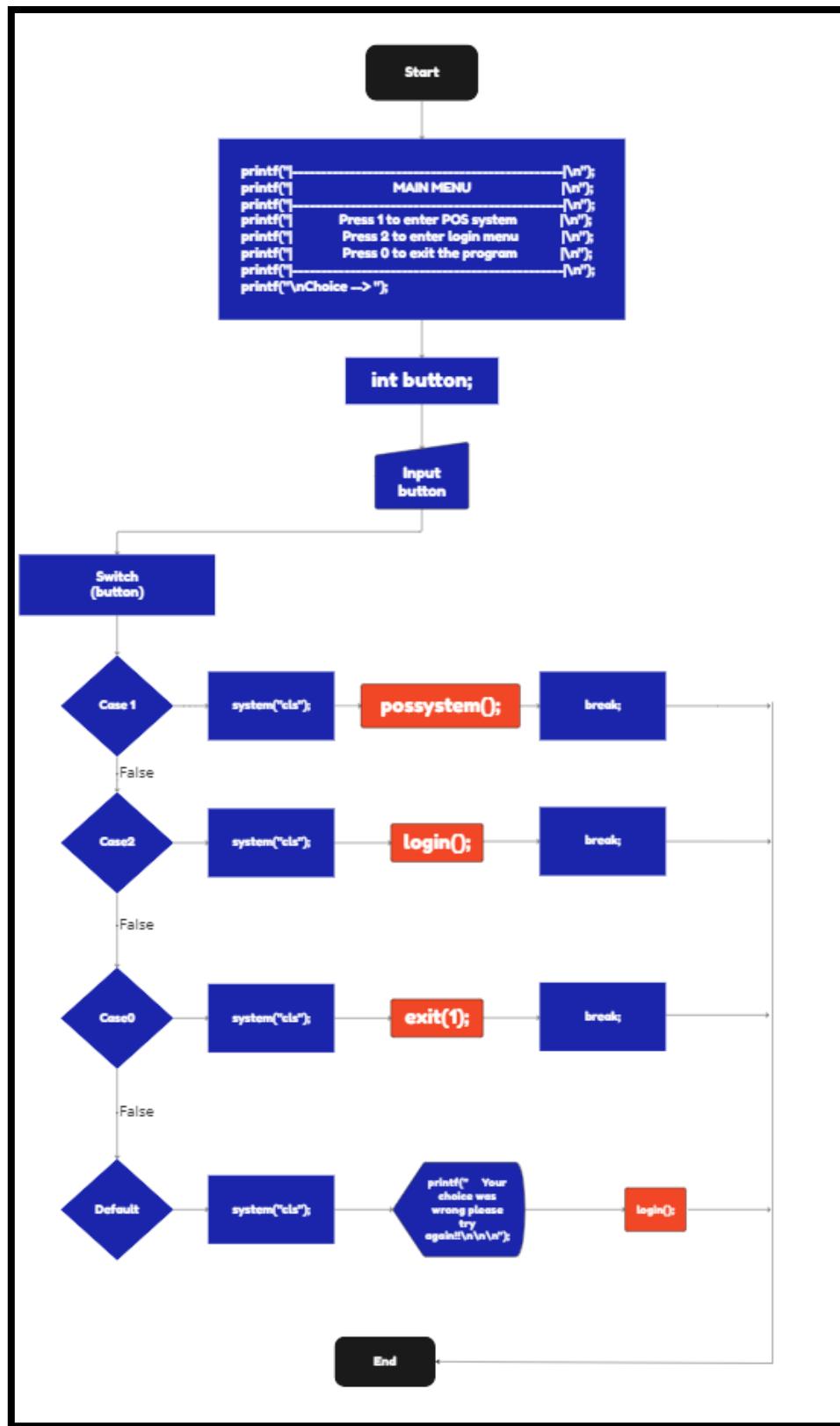
Int Accountingsystem



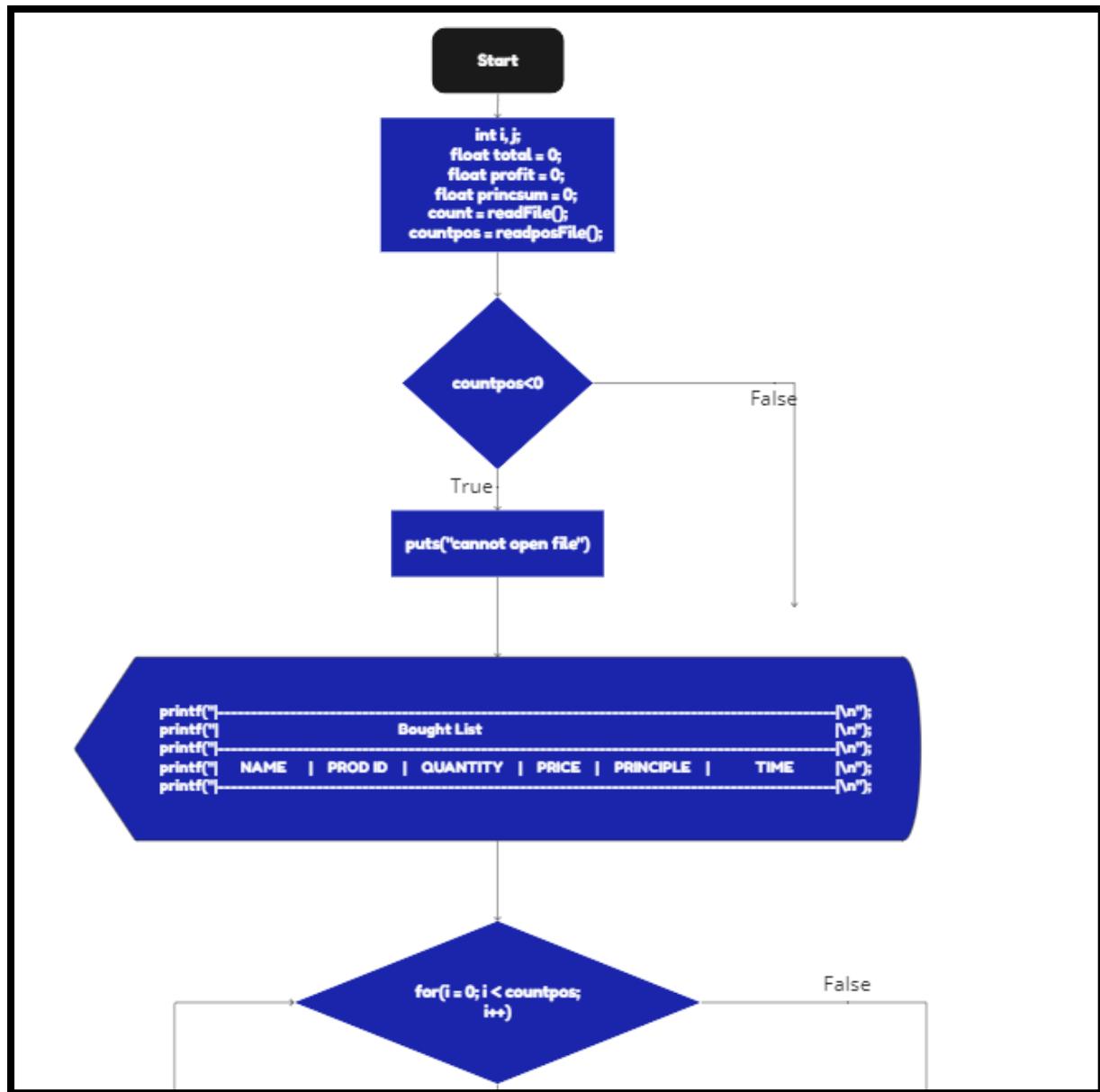
Int Possystem (1)



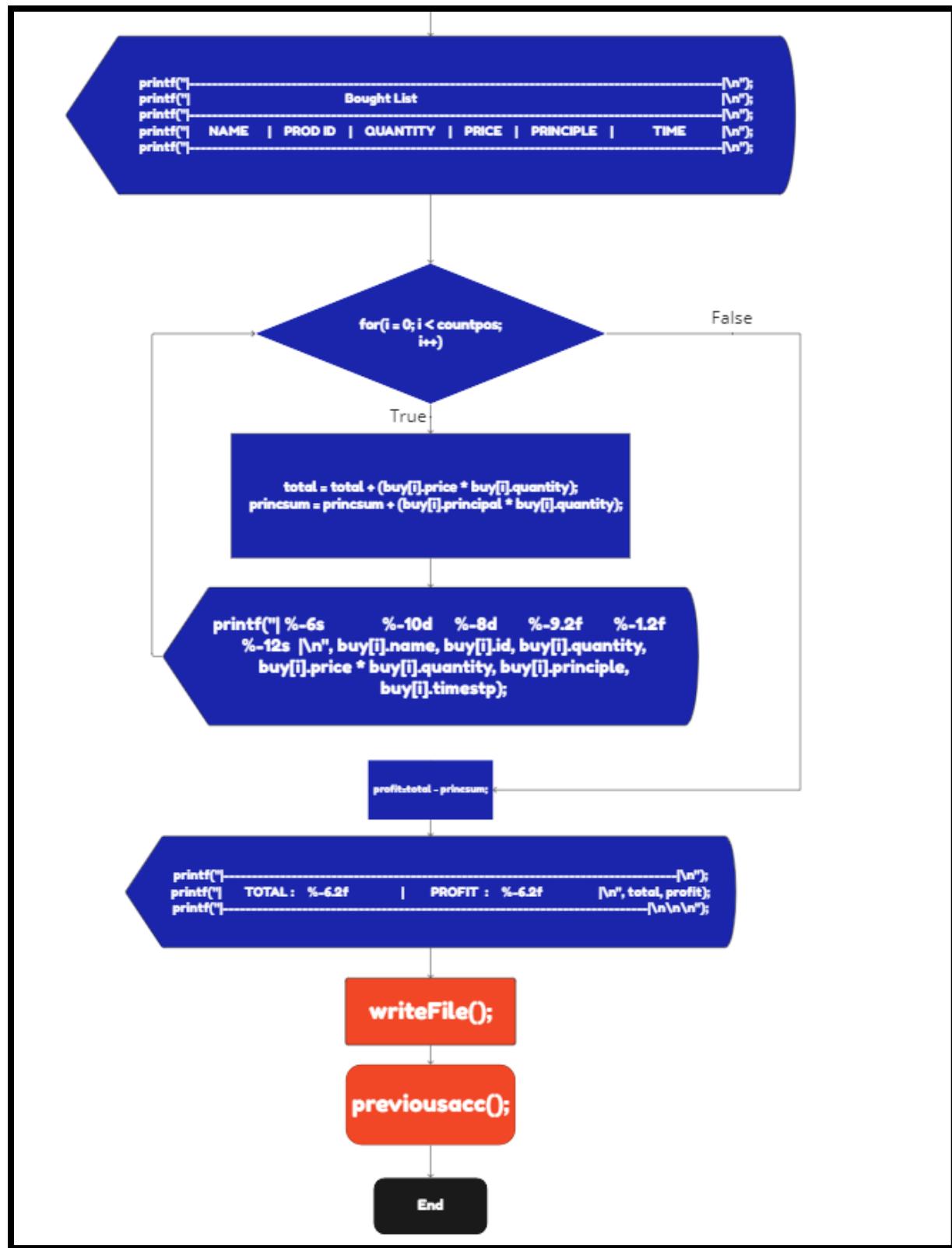
Int Possystem (2)



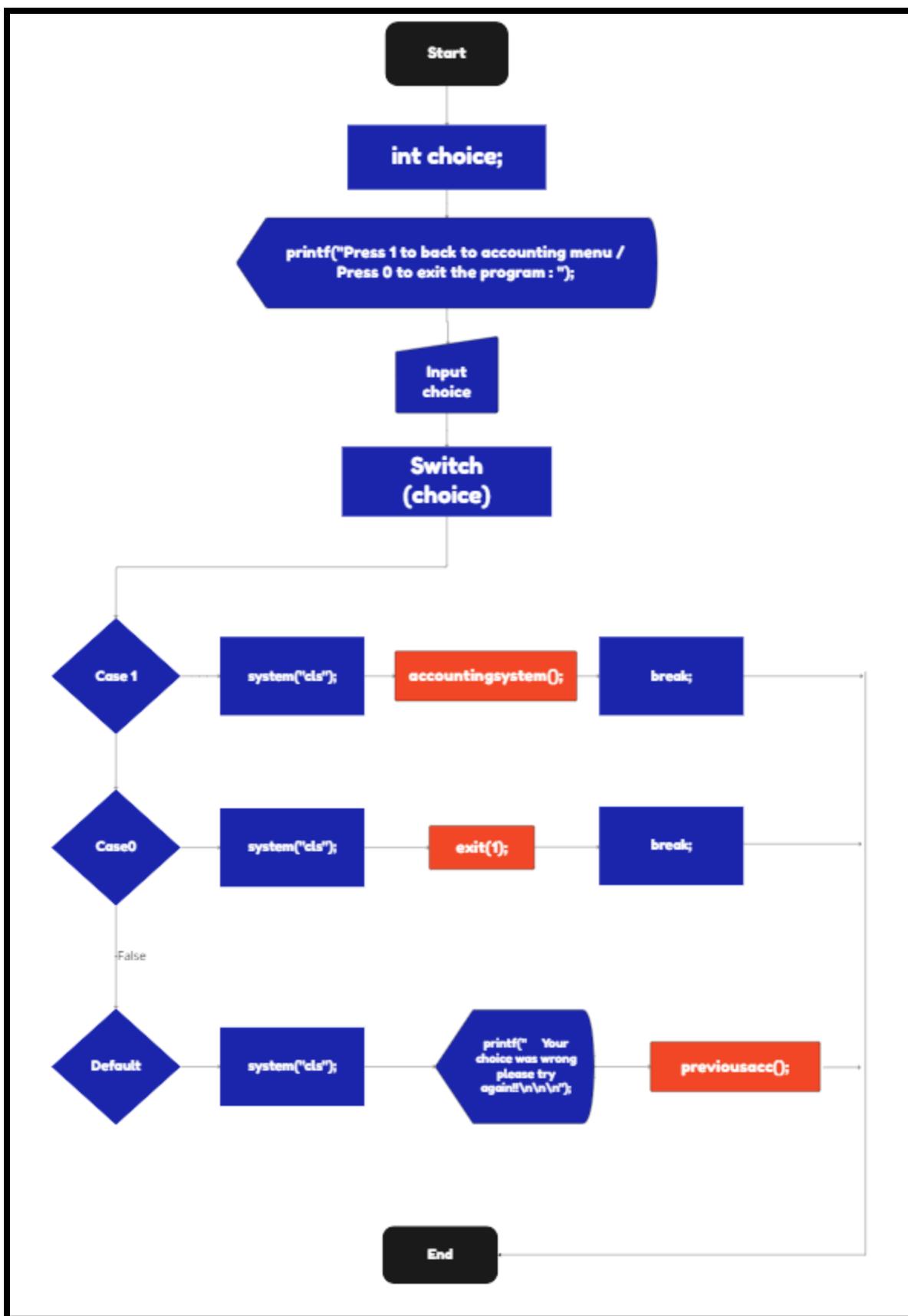
Int Clientmainmenuselection



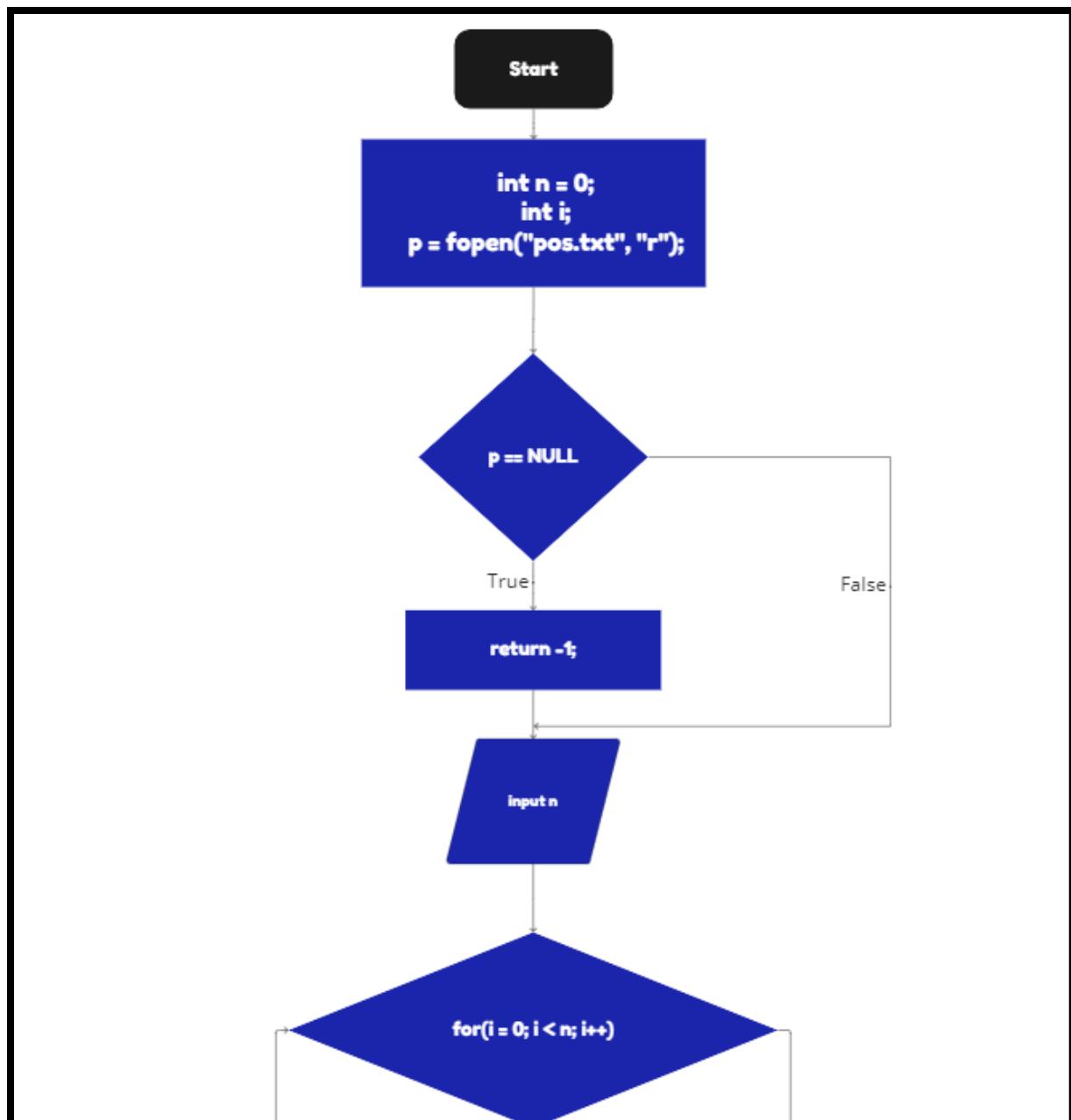
Int Purchasedlists (1)



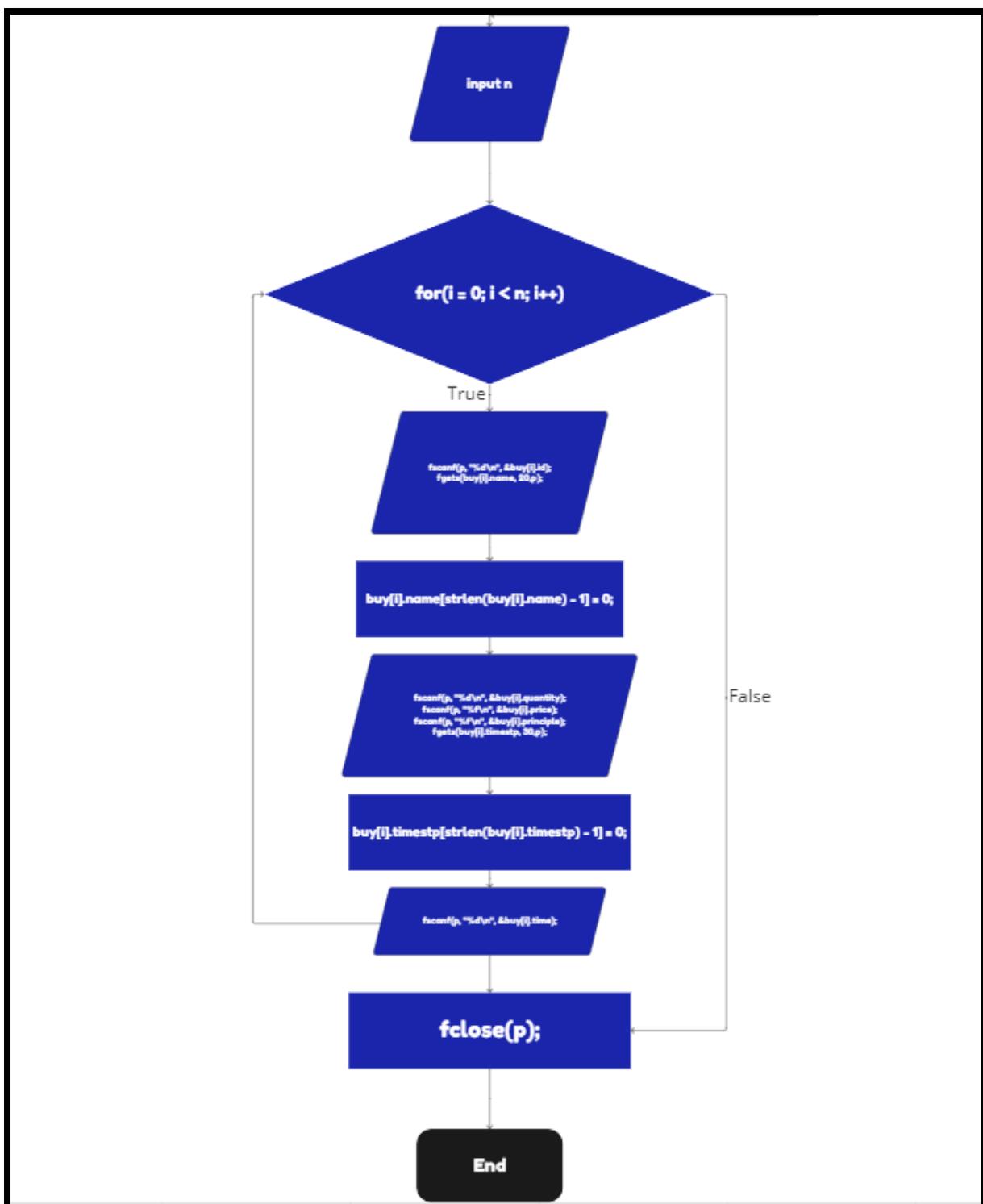
Int Purchasedlists (2)



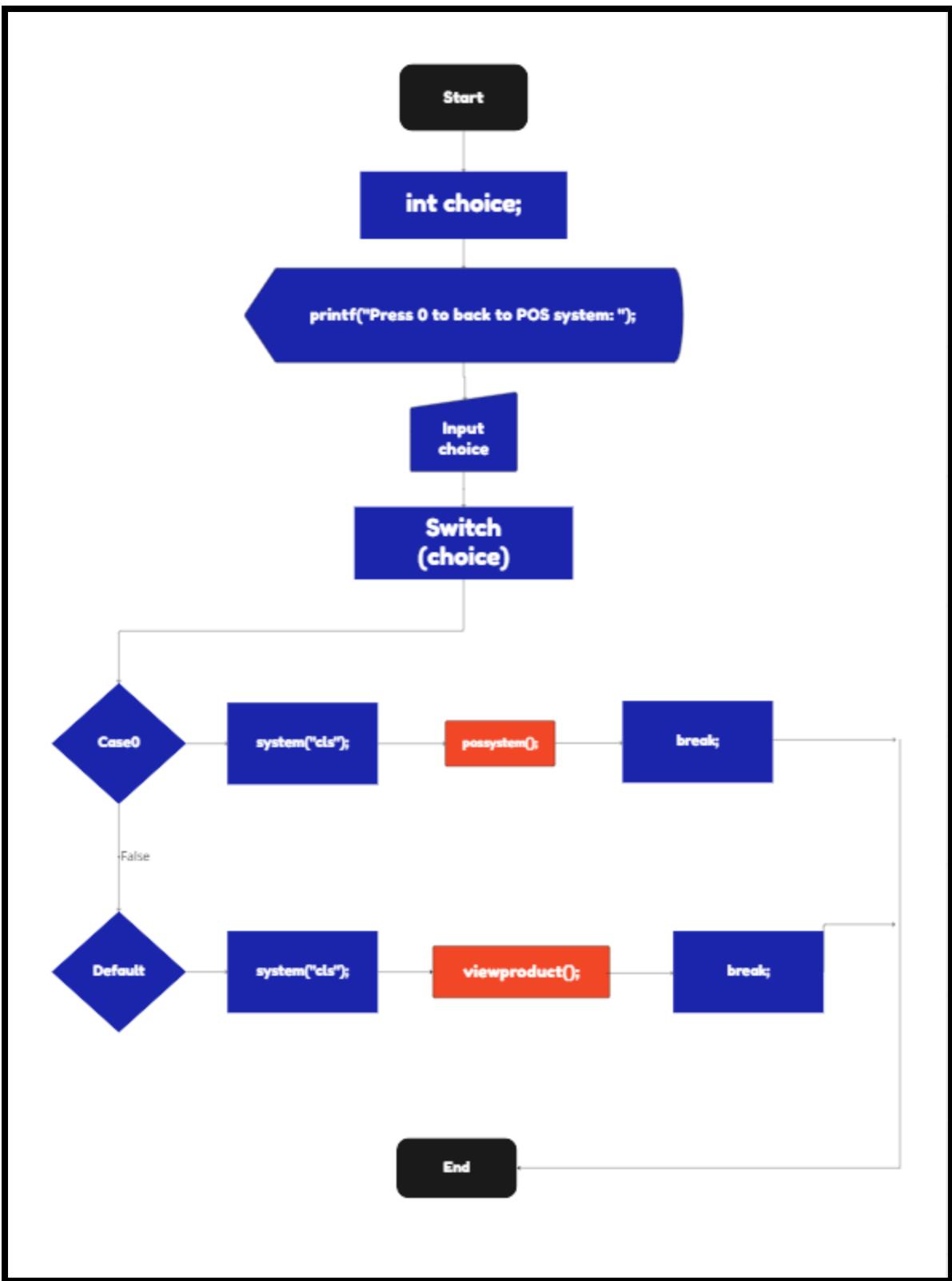
Int Previousacc



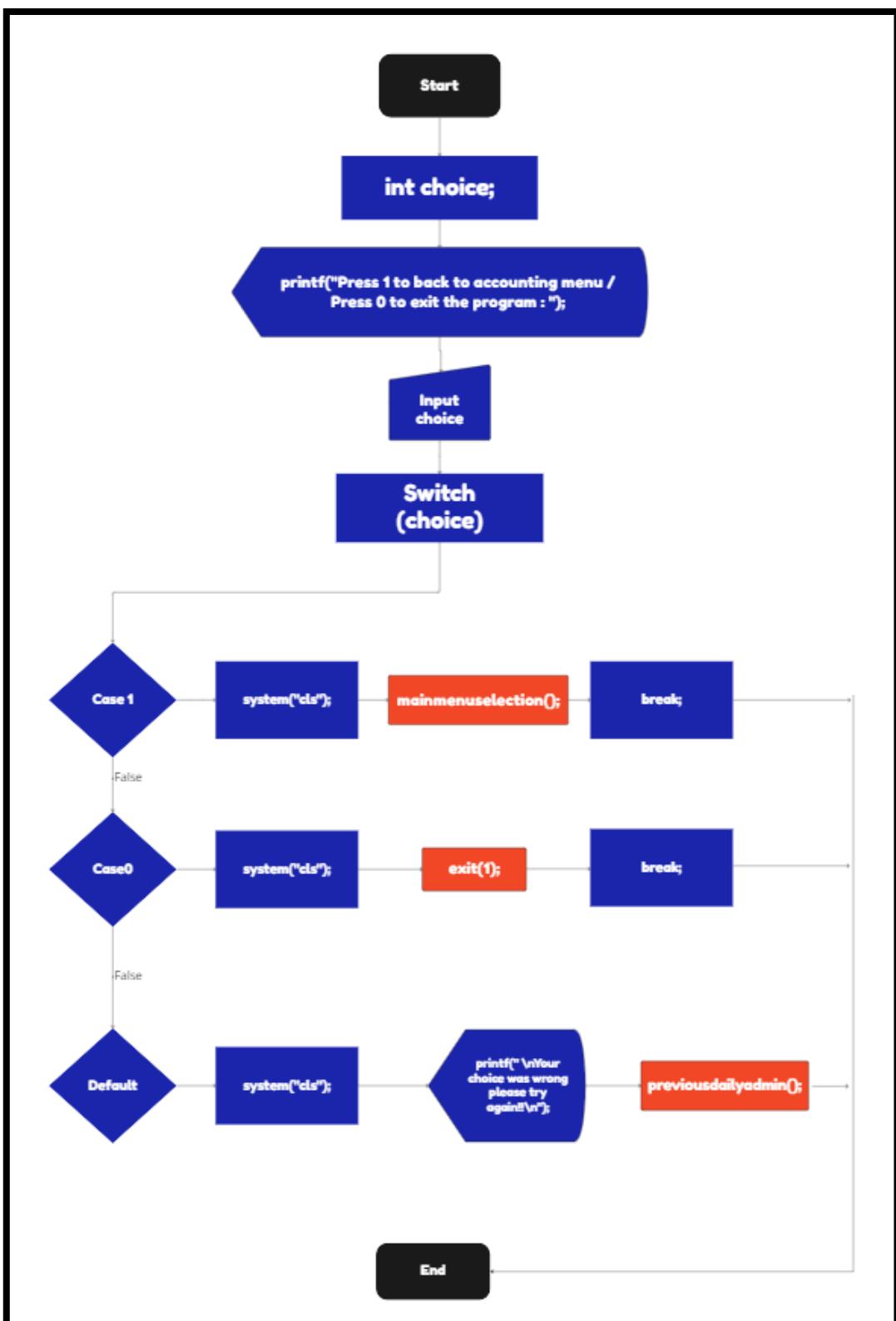
Int readposfile (1)



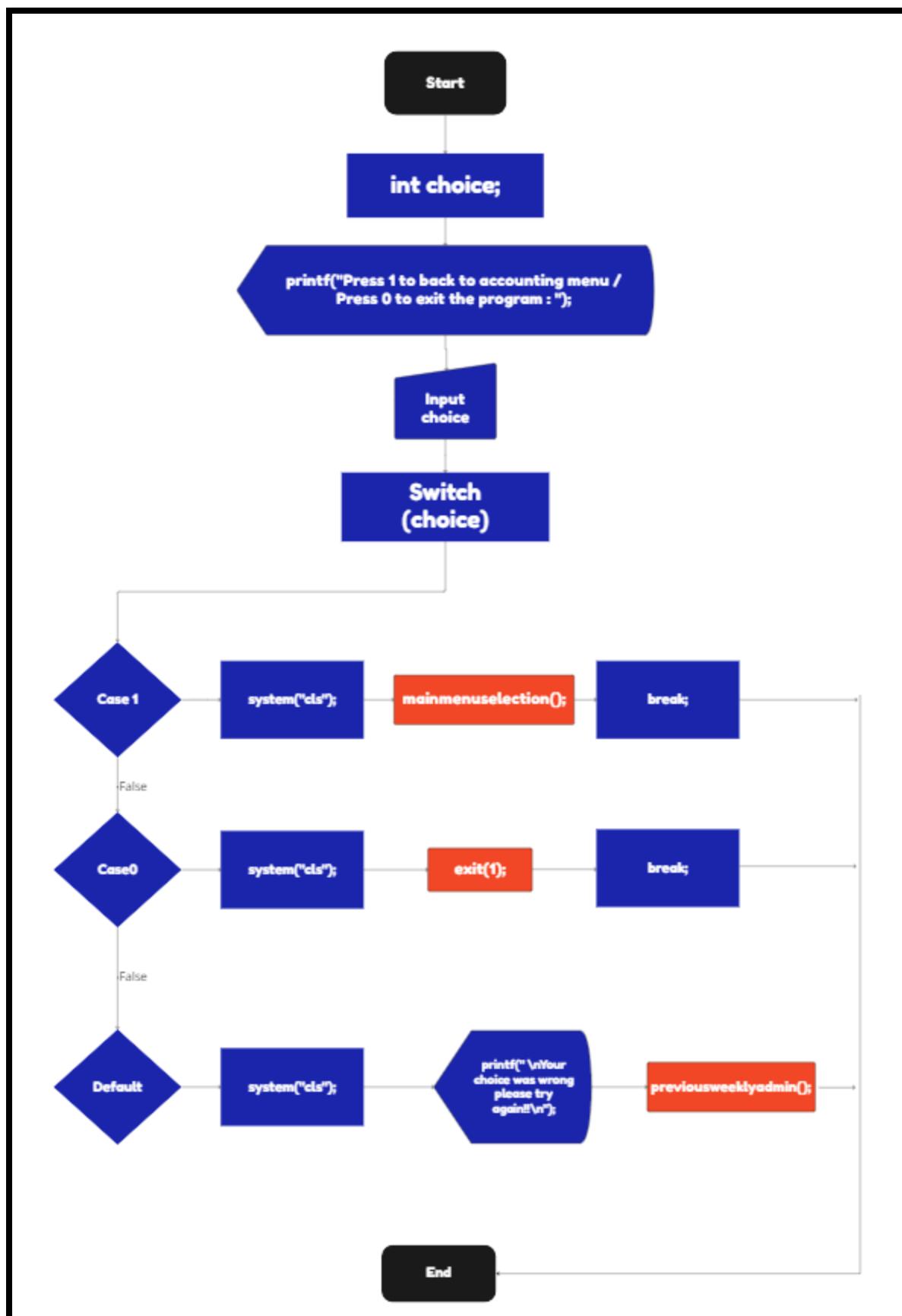
Int readposfile (2)



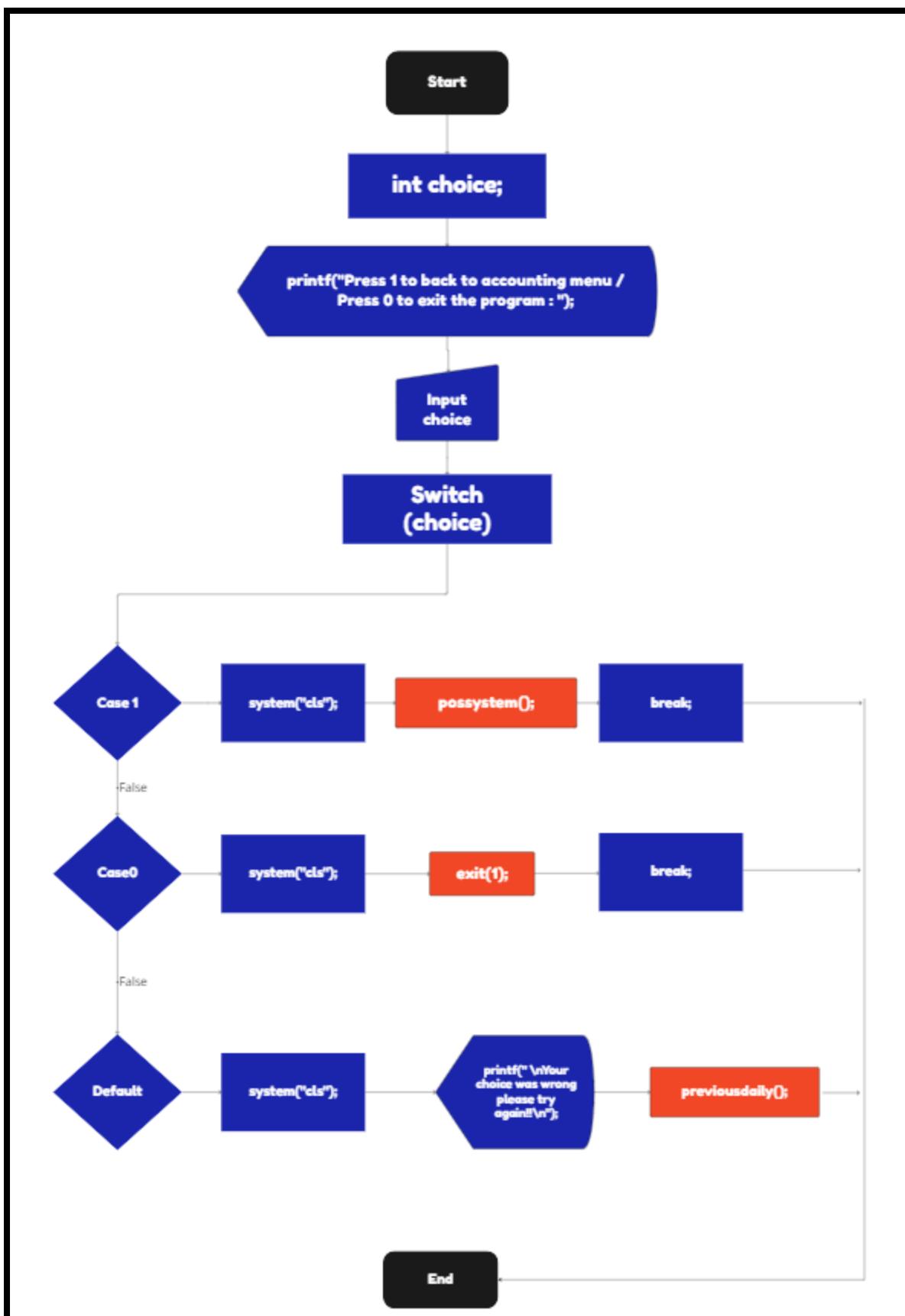
Void Previouspos



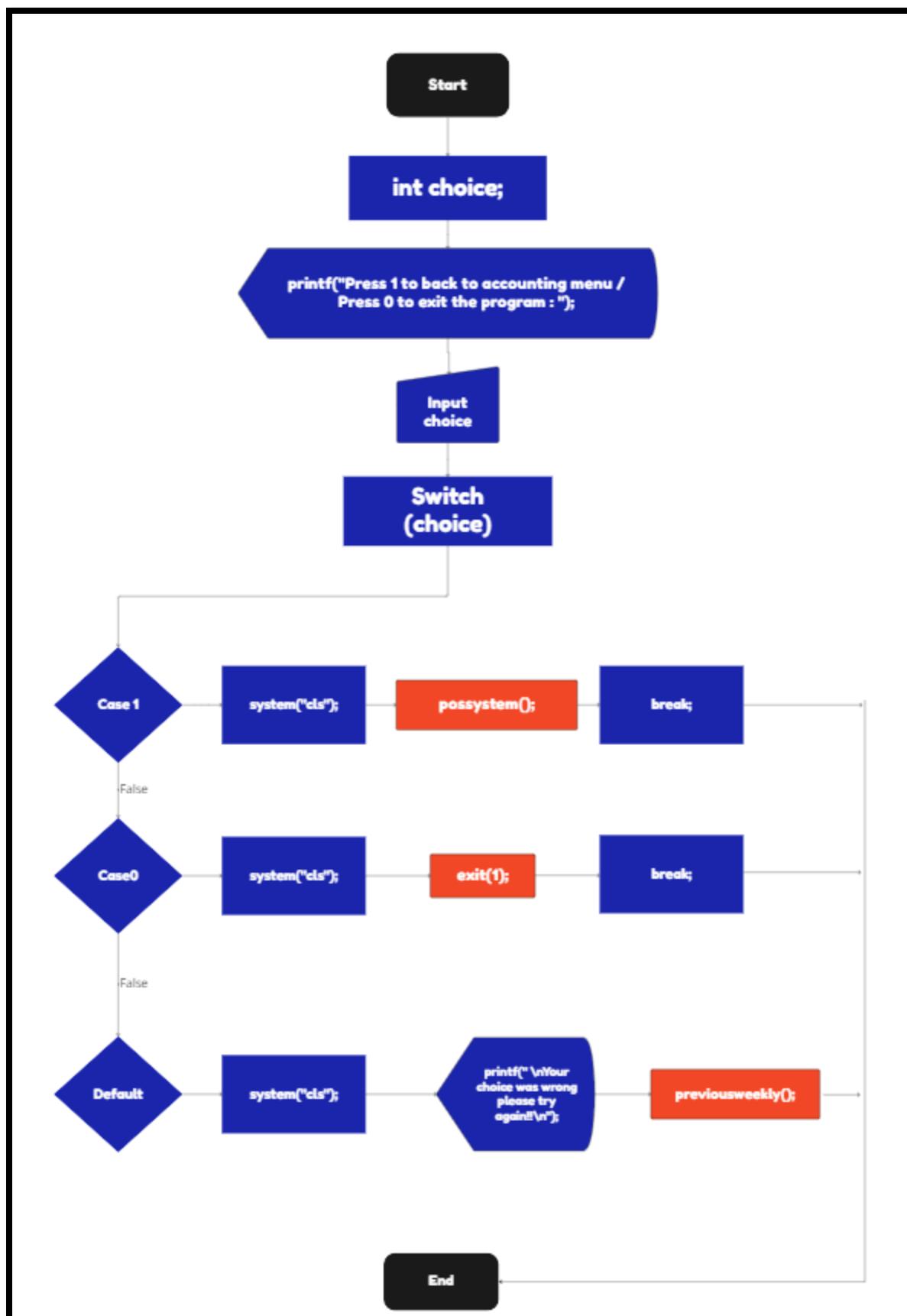
Void Previousdailyadmin



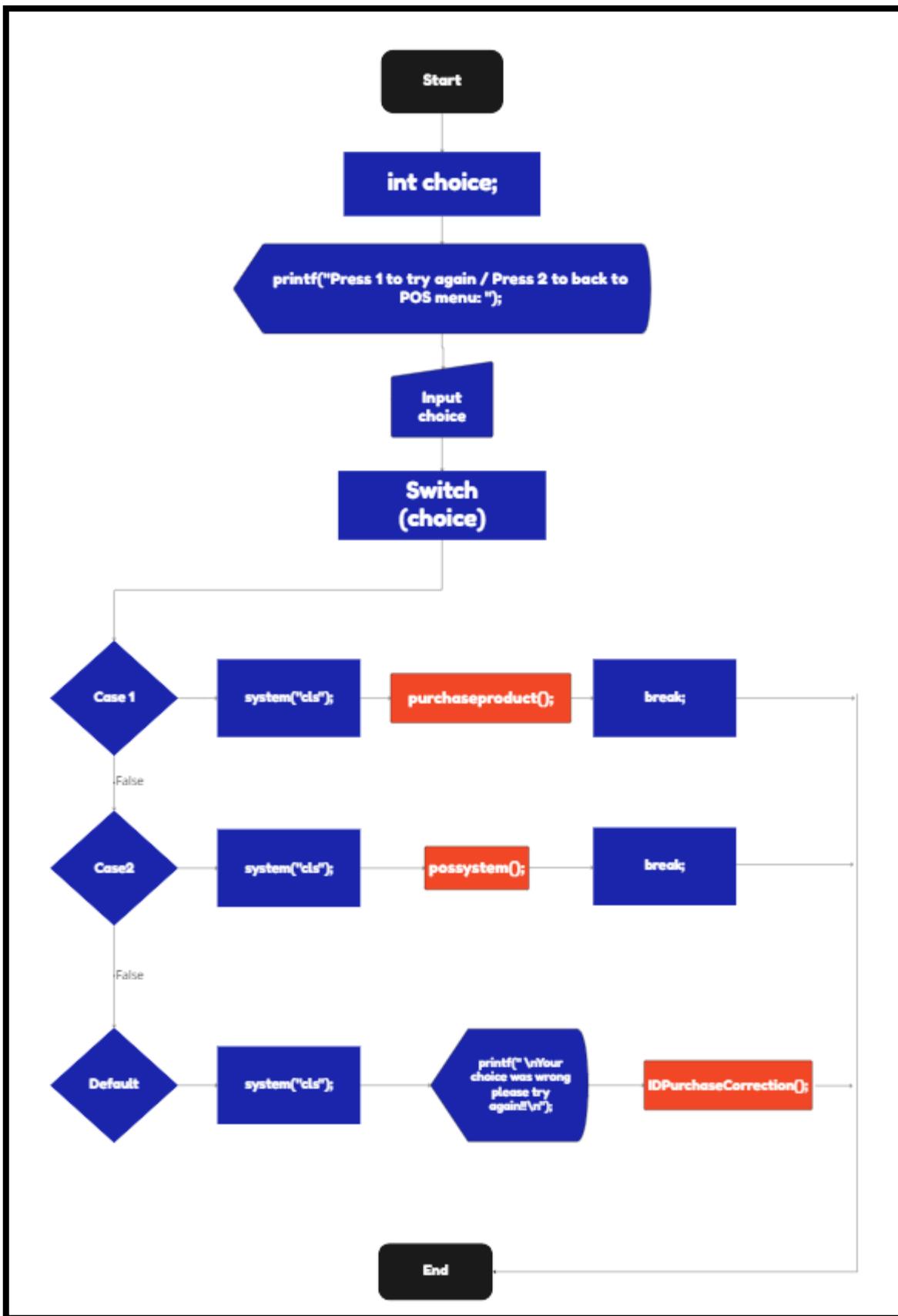
Void Previousweeklyadmin



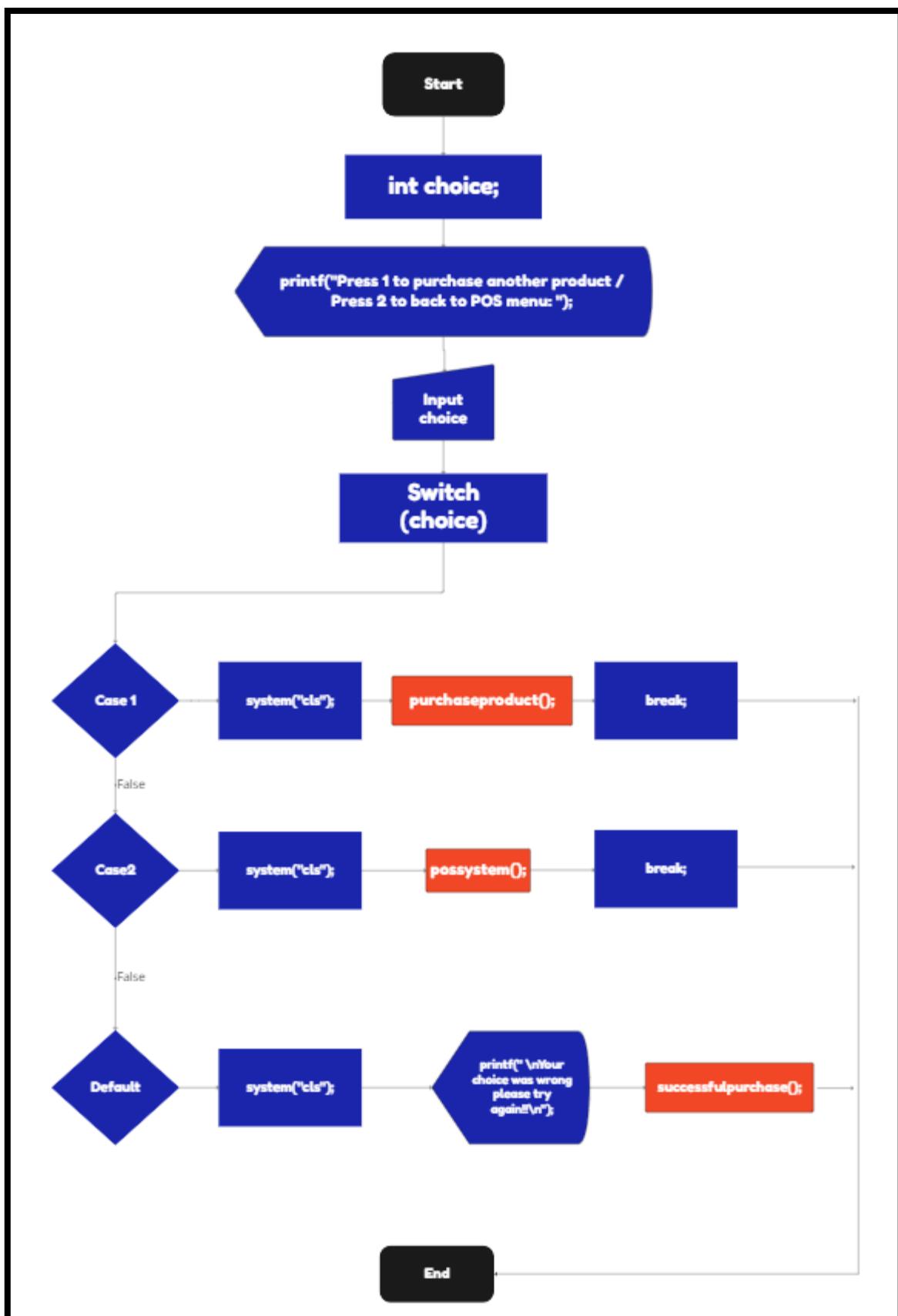
Void Previousdaily



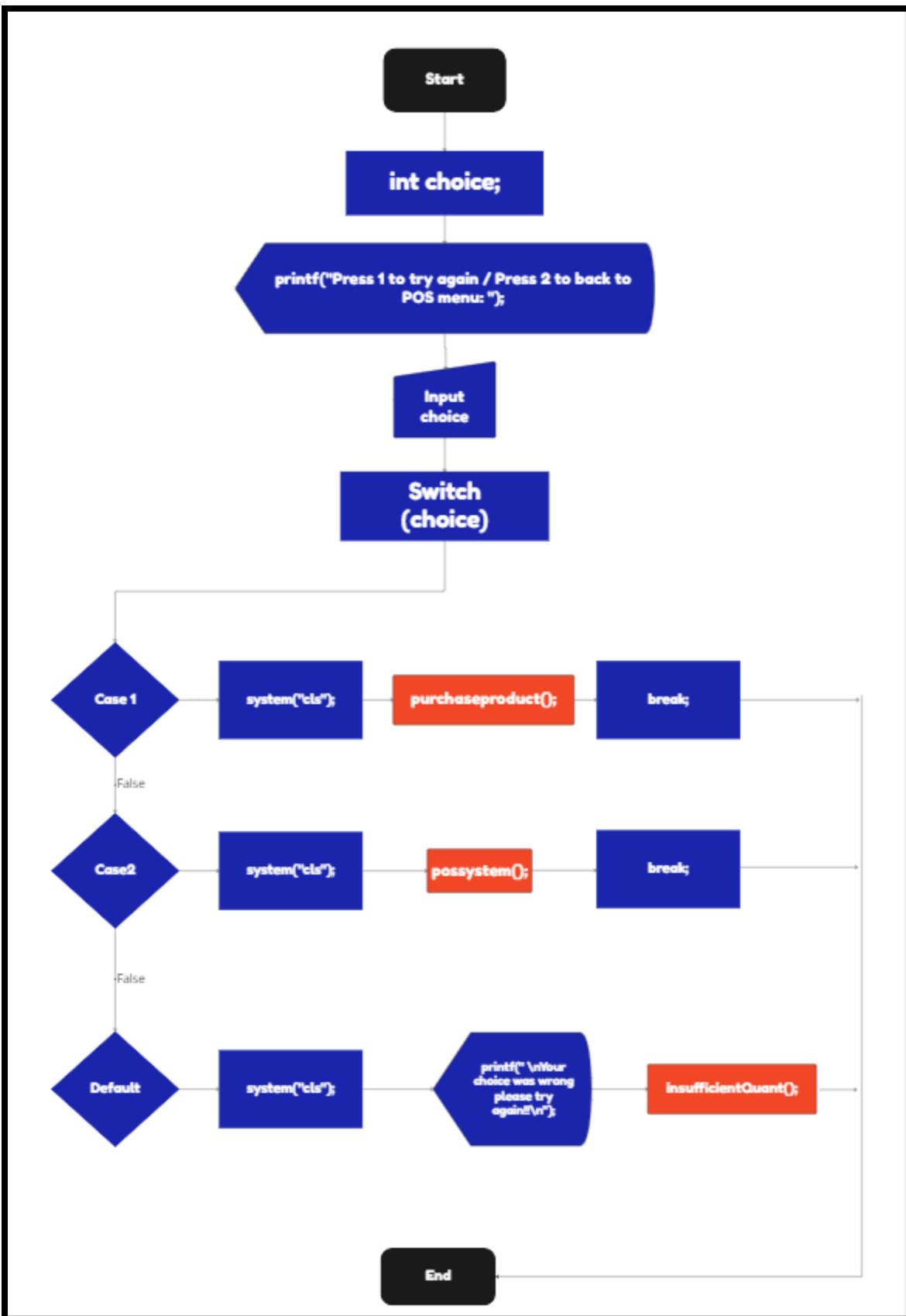
Void Previousweekly



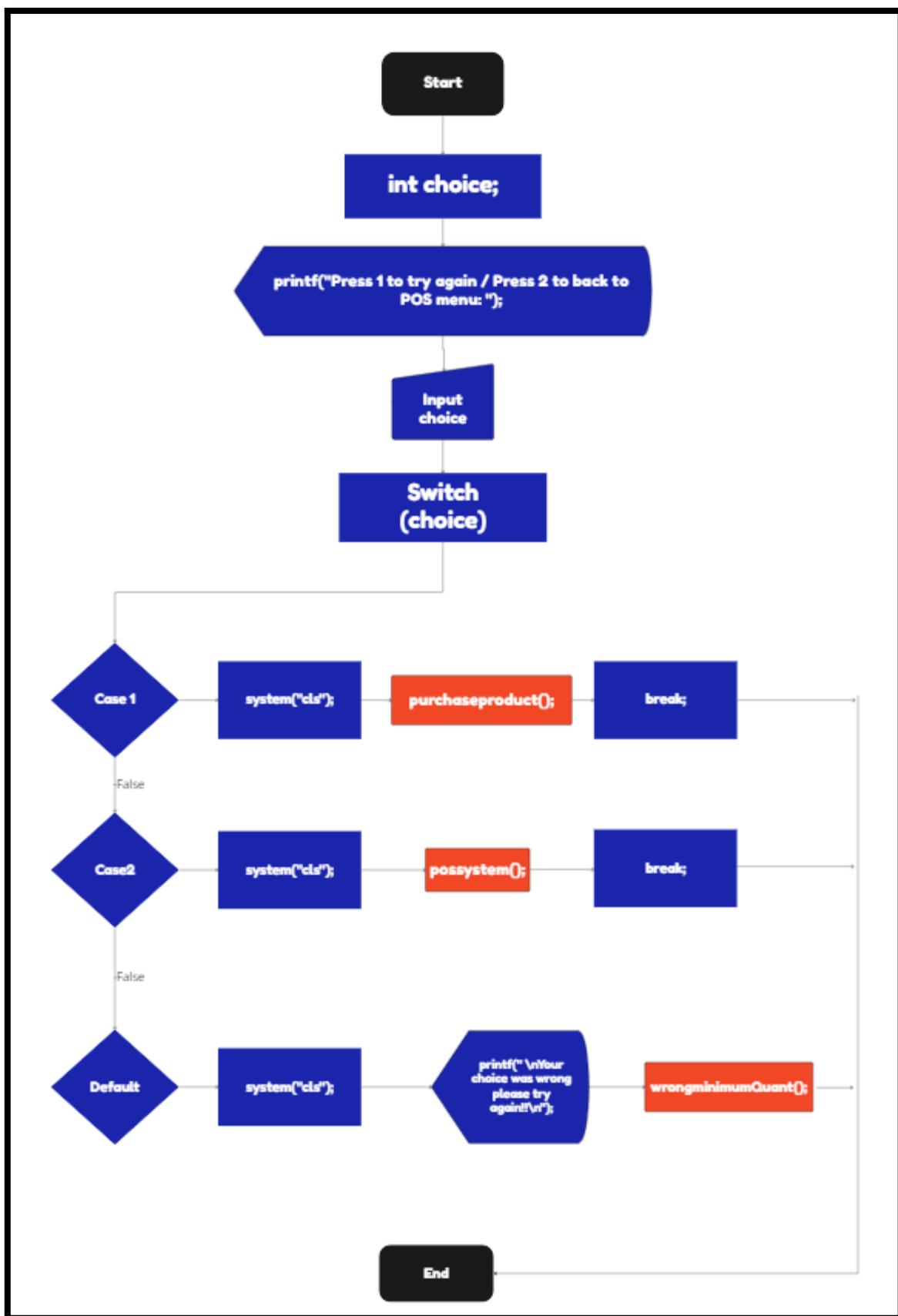
Void IDPurchaseCorrection



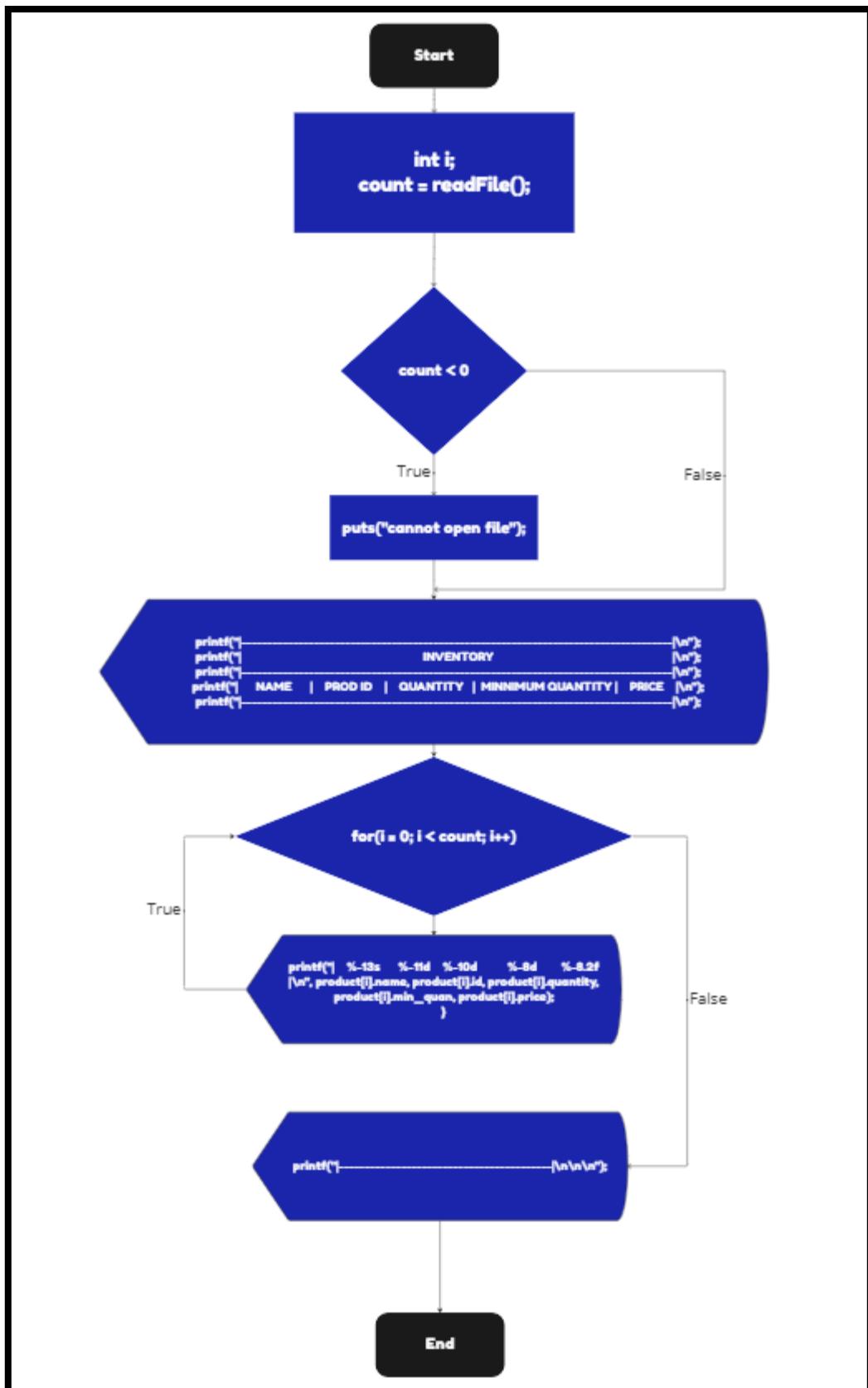
Void Successfulpurchase



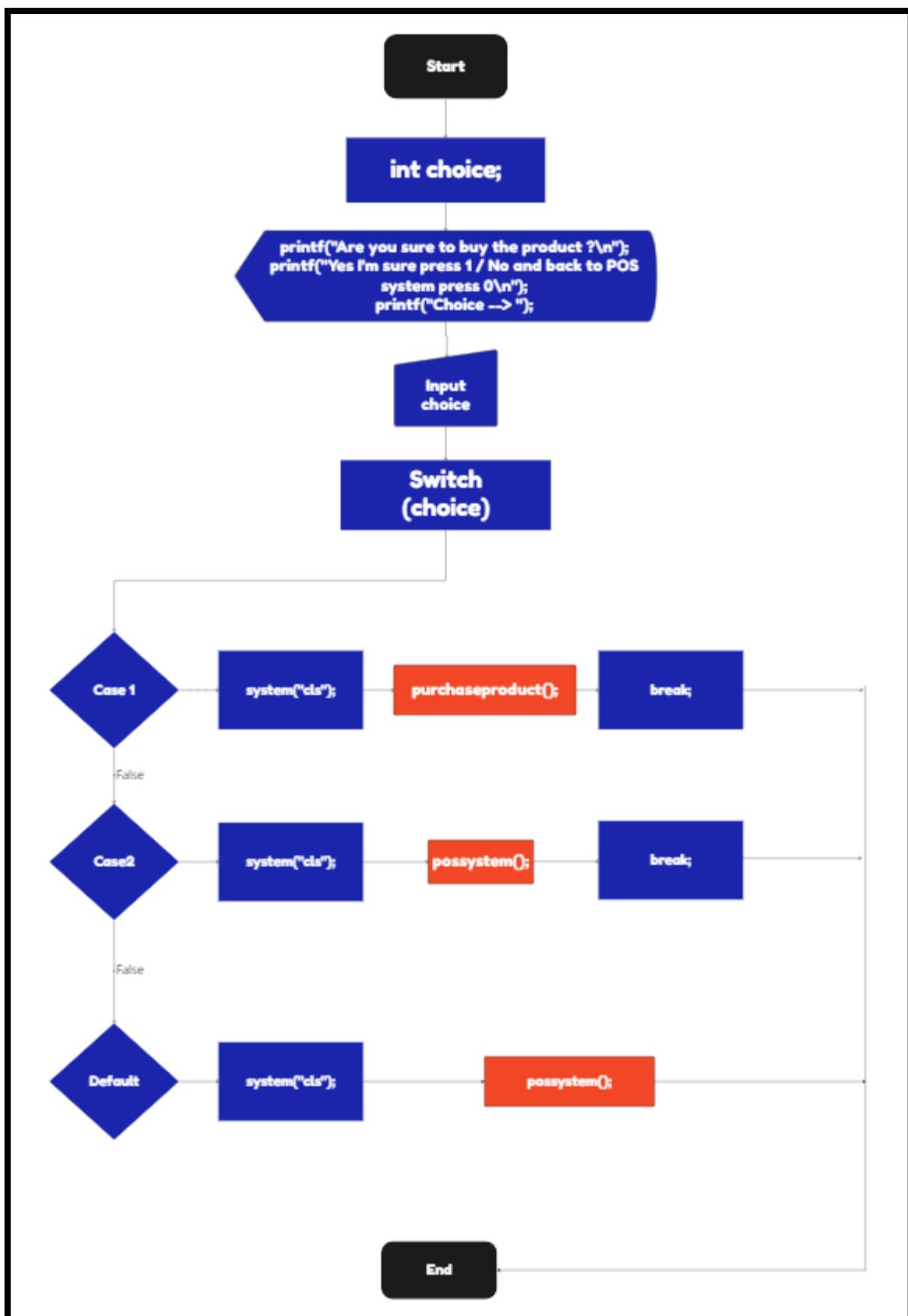
Void InsufficientQuant



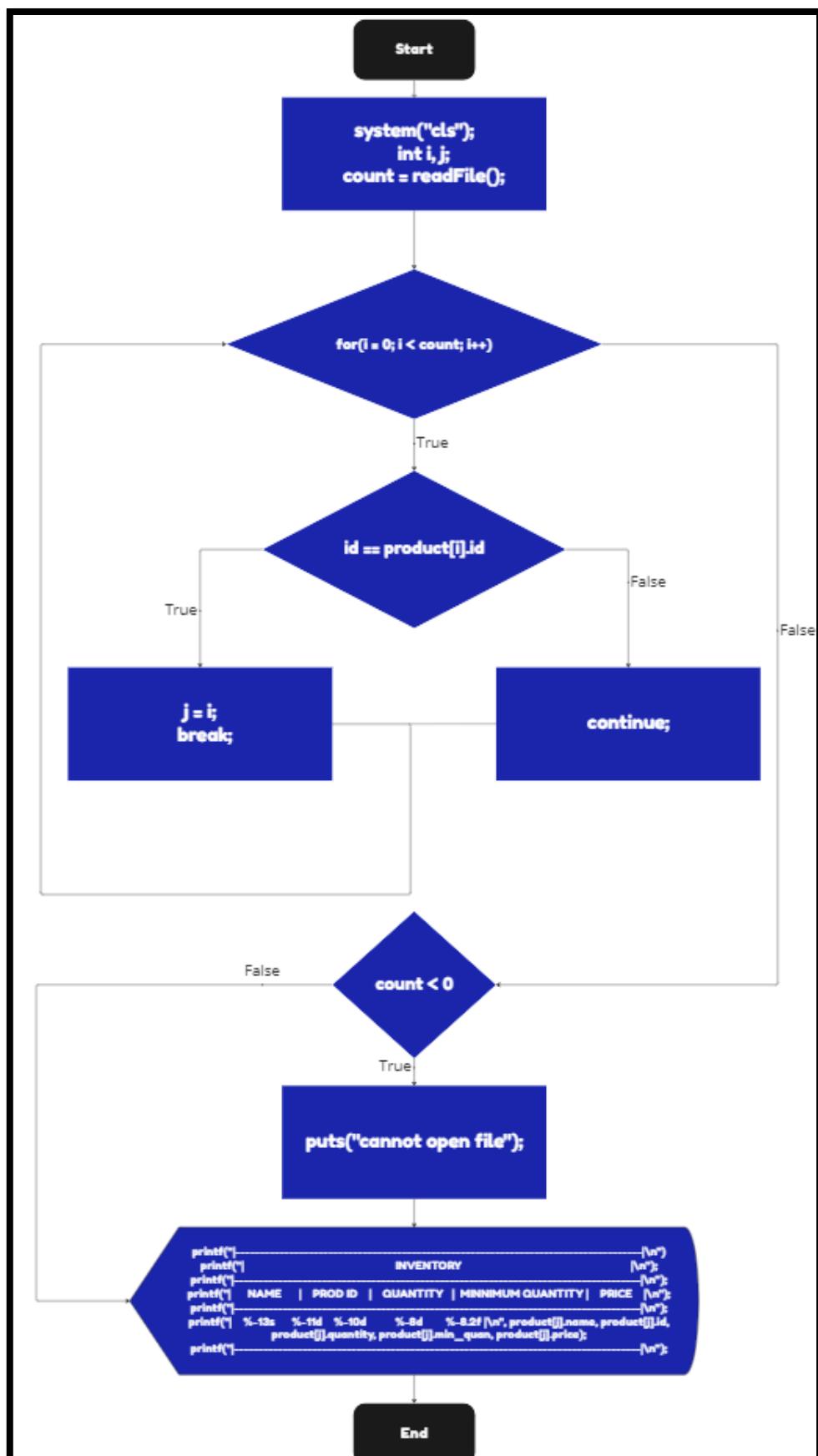
Void WrongminimumQuant



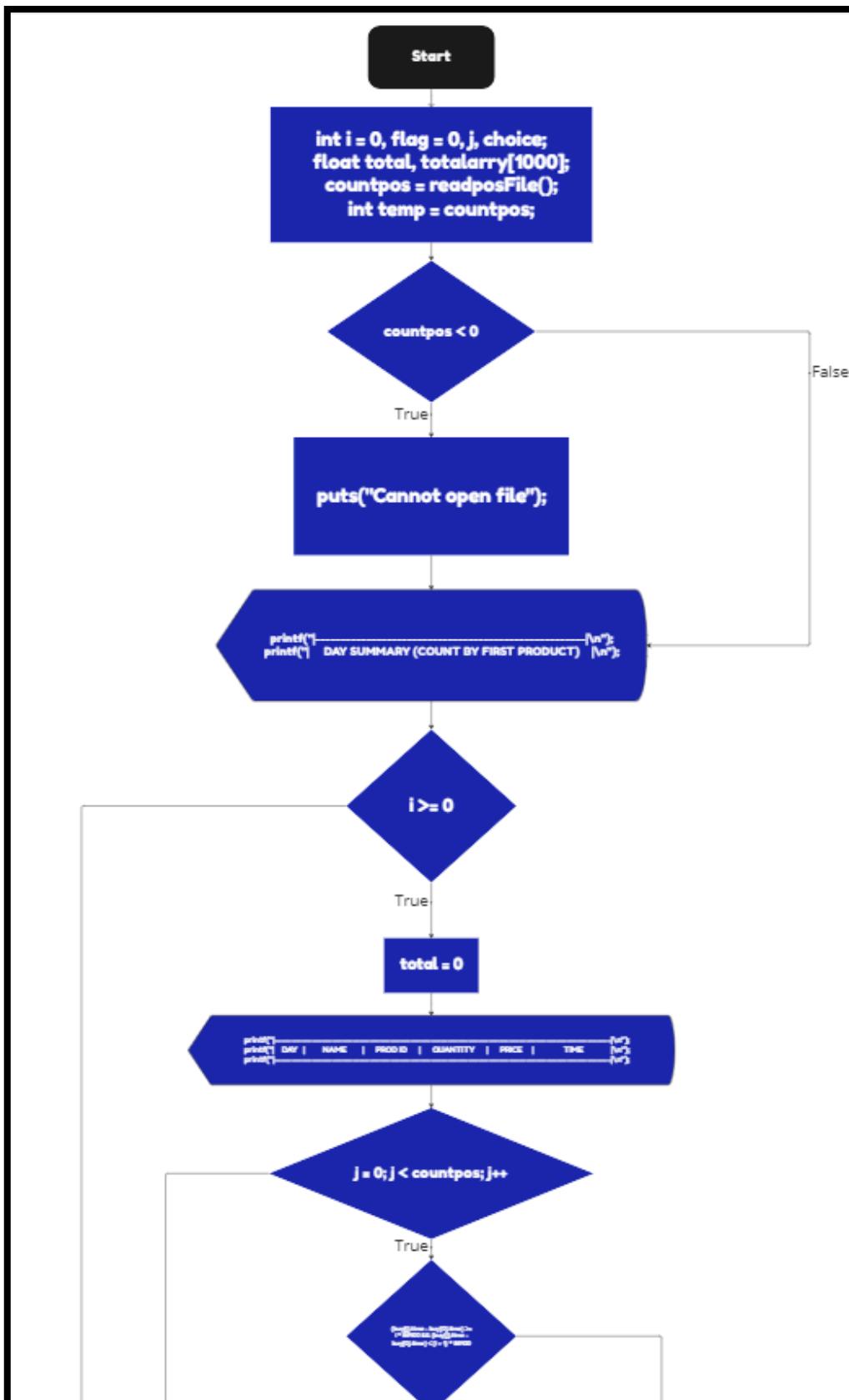
Void Viewproduct



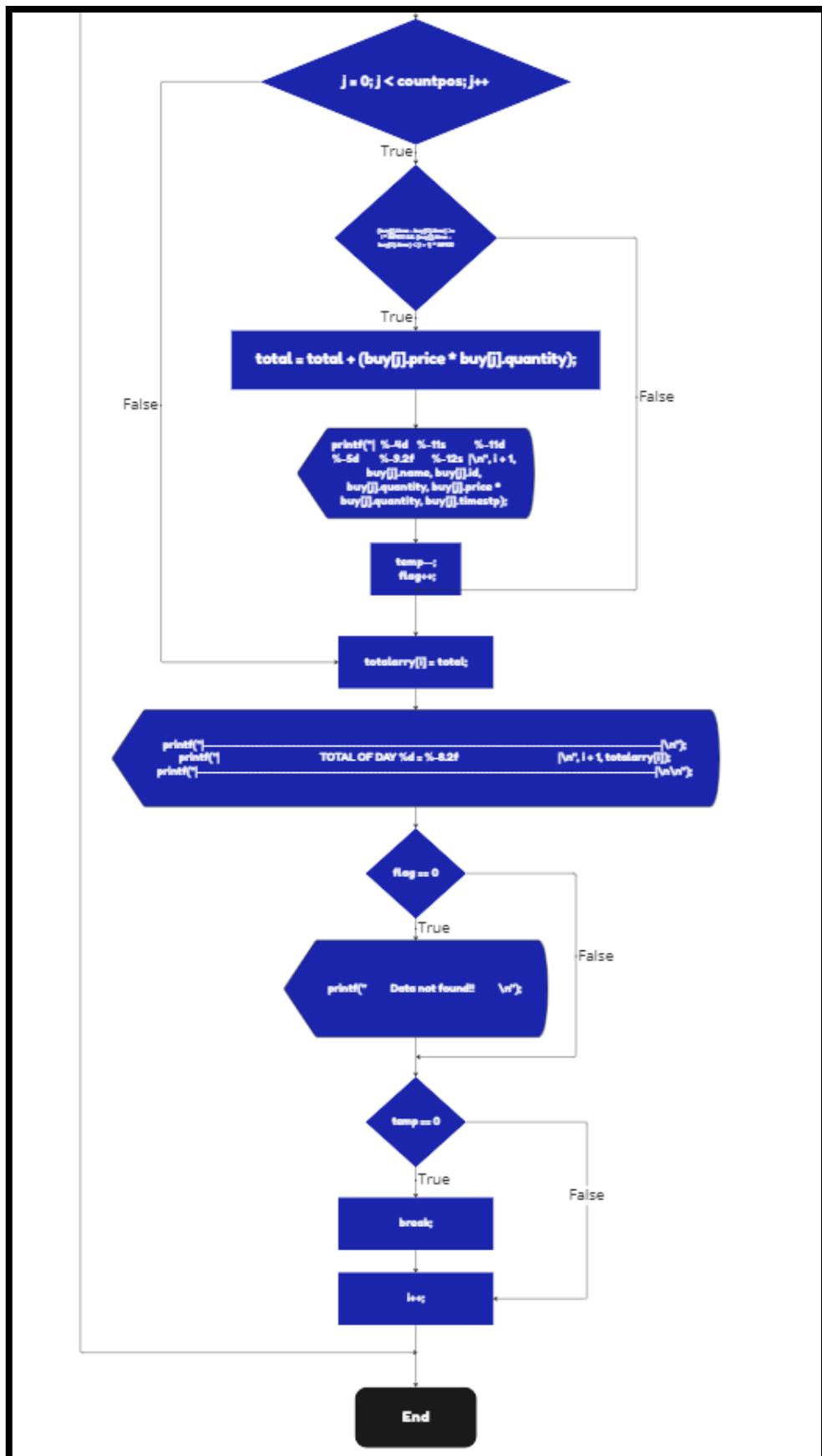
Void Viewproductforpurchase



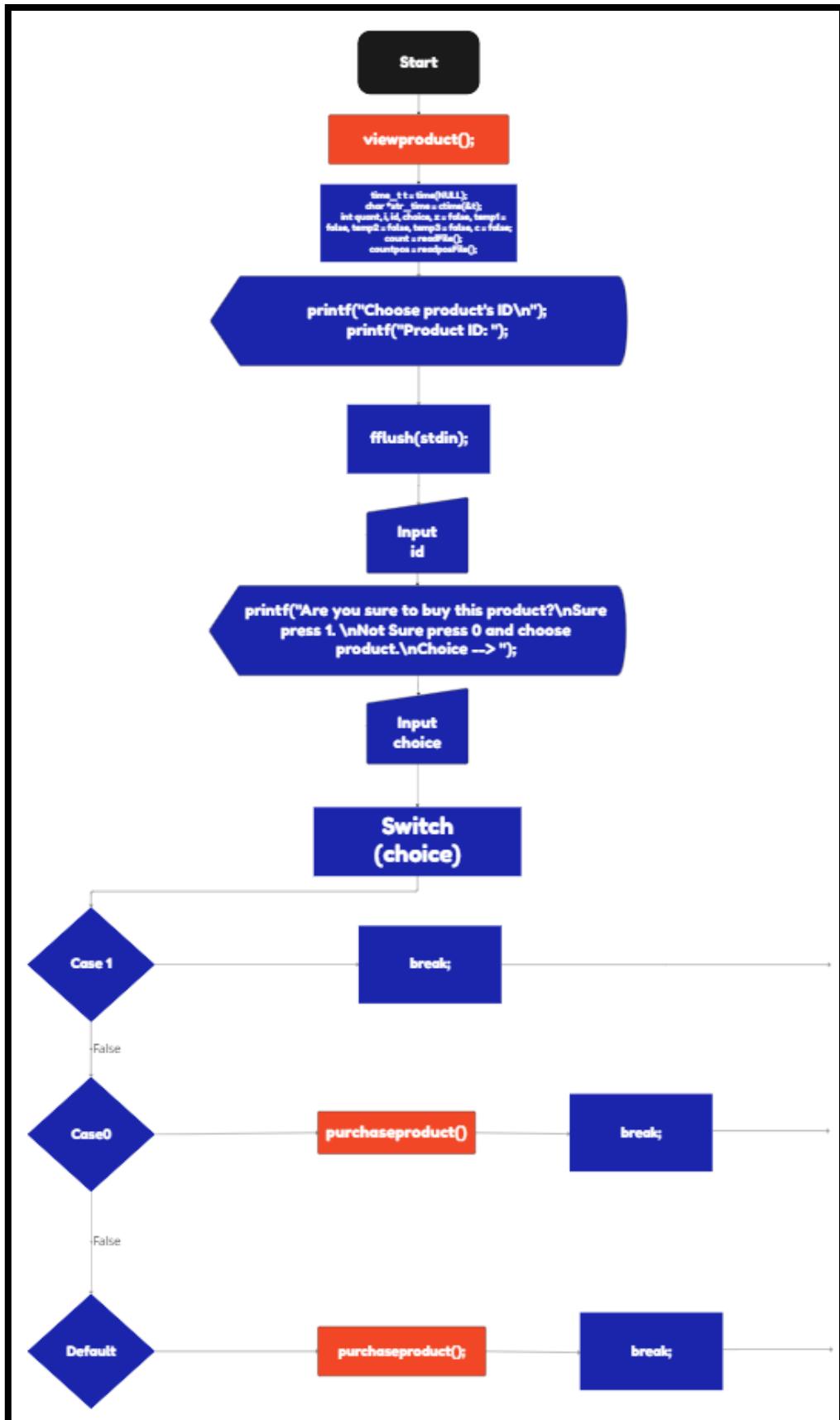
Void Cartviewproductforpurchase



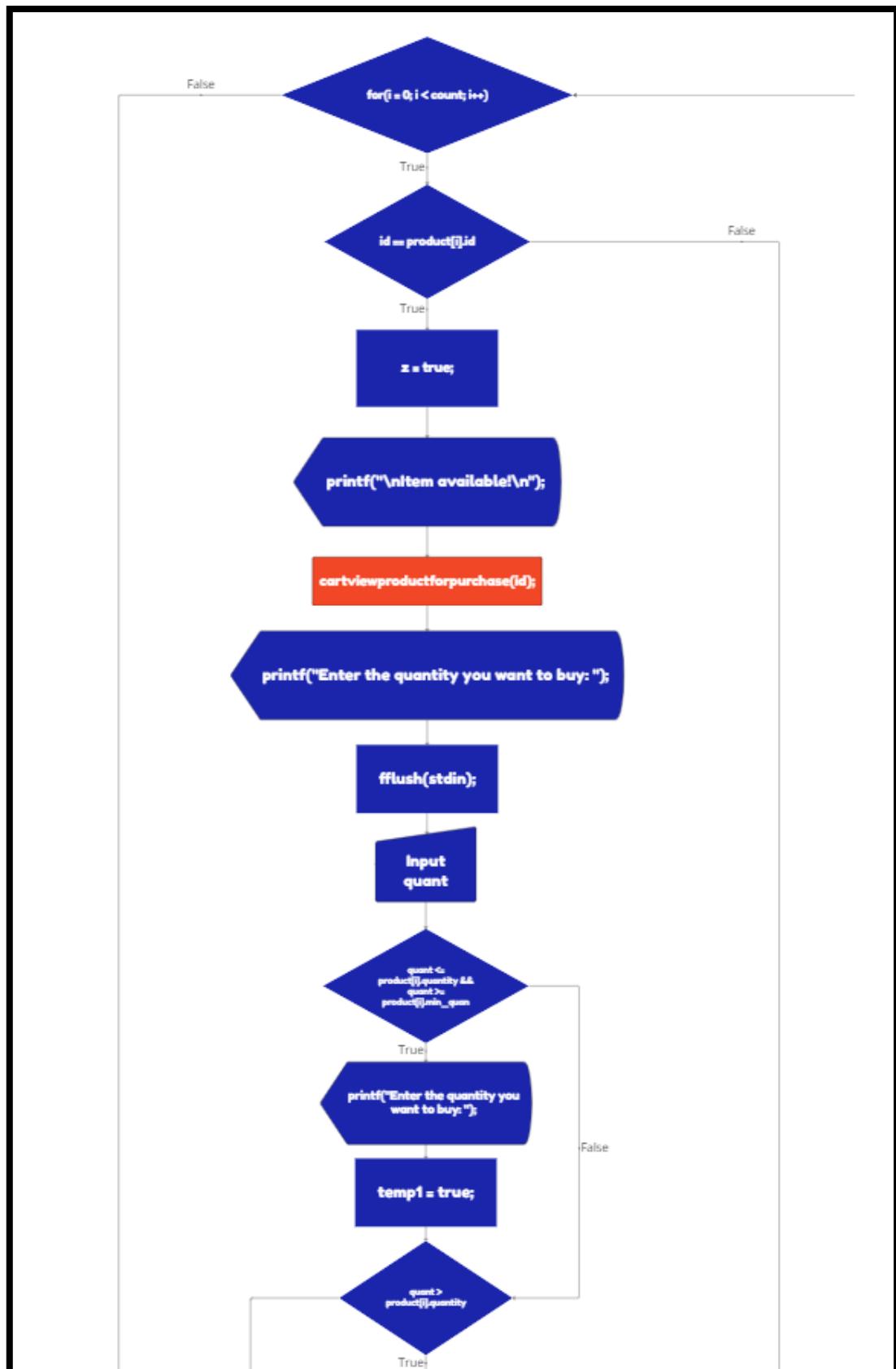
Void daily (1)



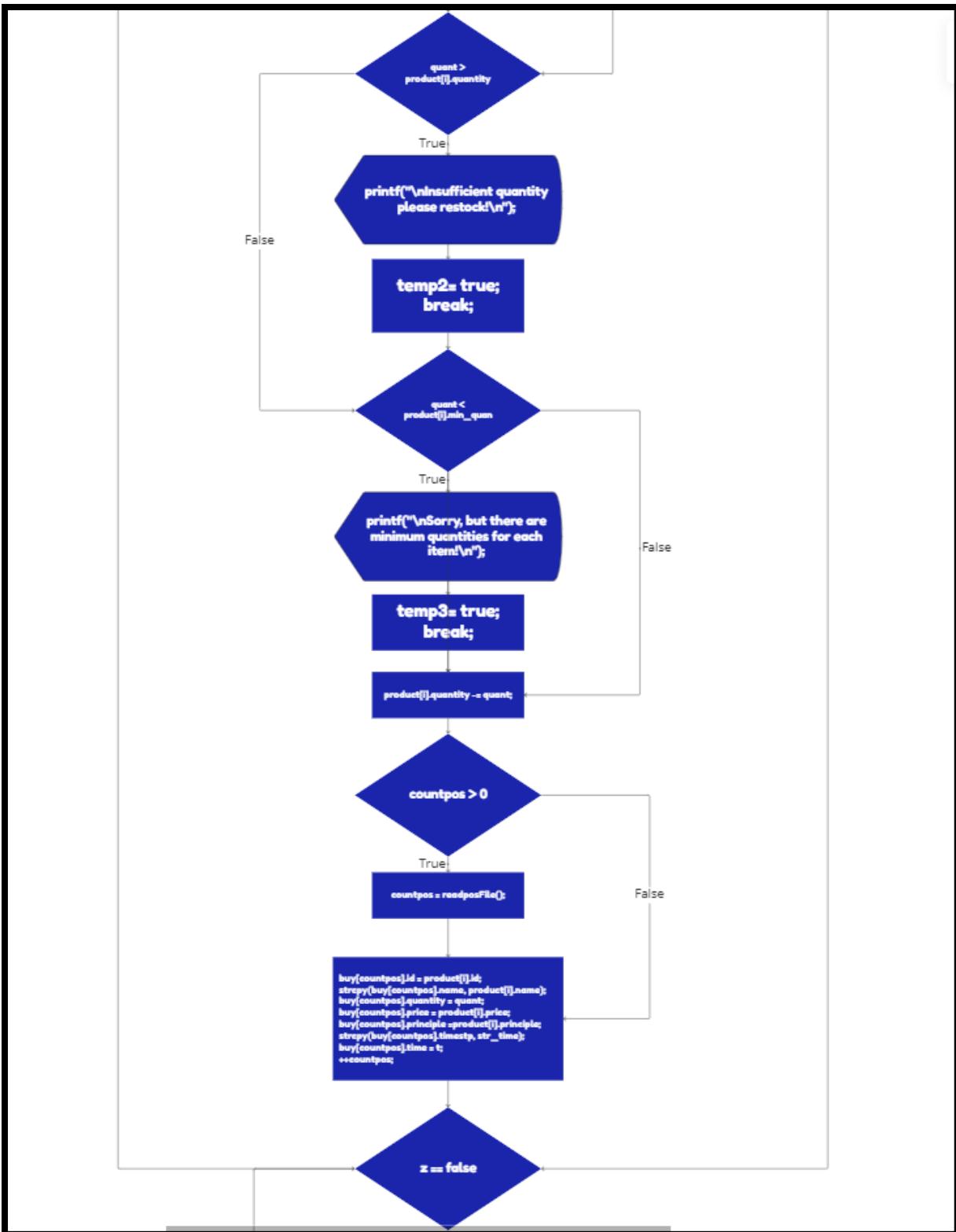
Void daily (2)



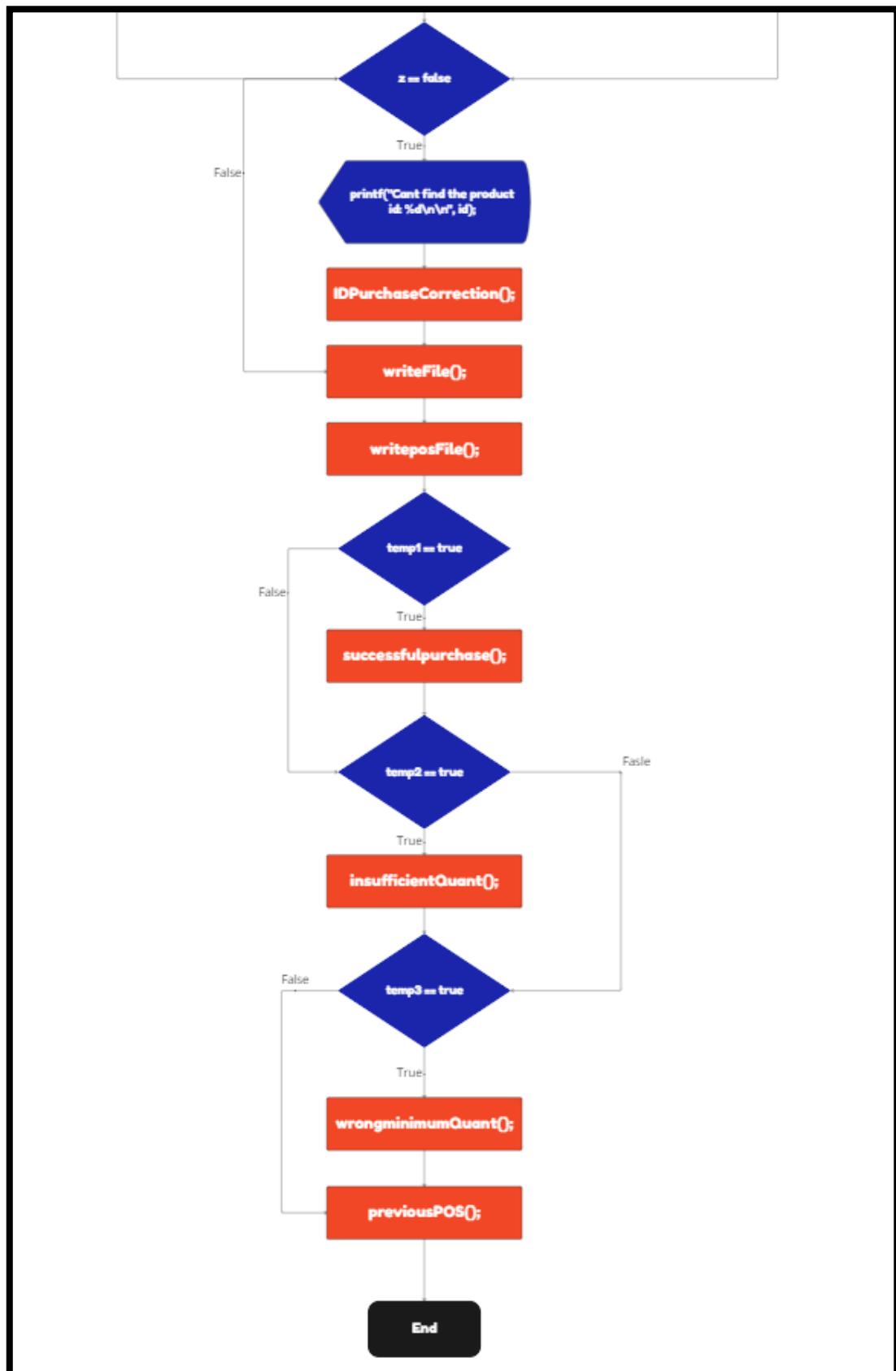
Void Purchaseproduct (1)



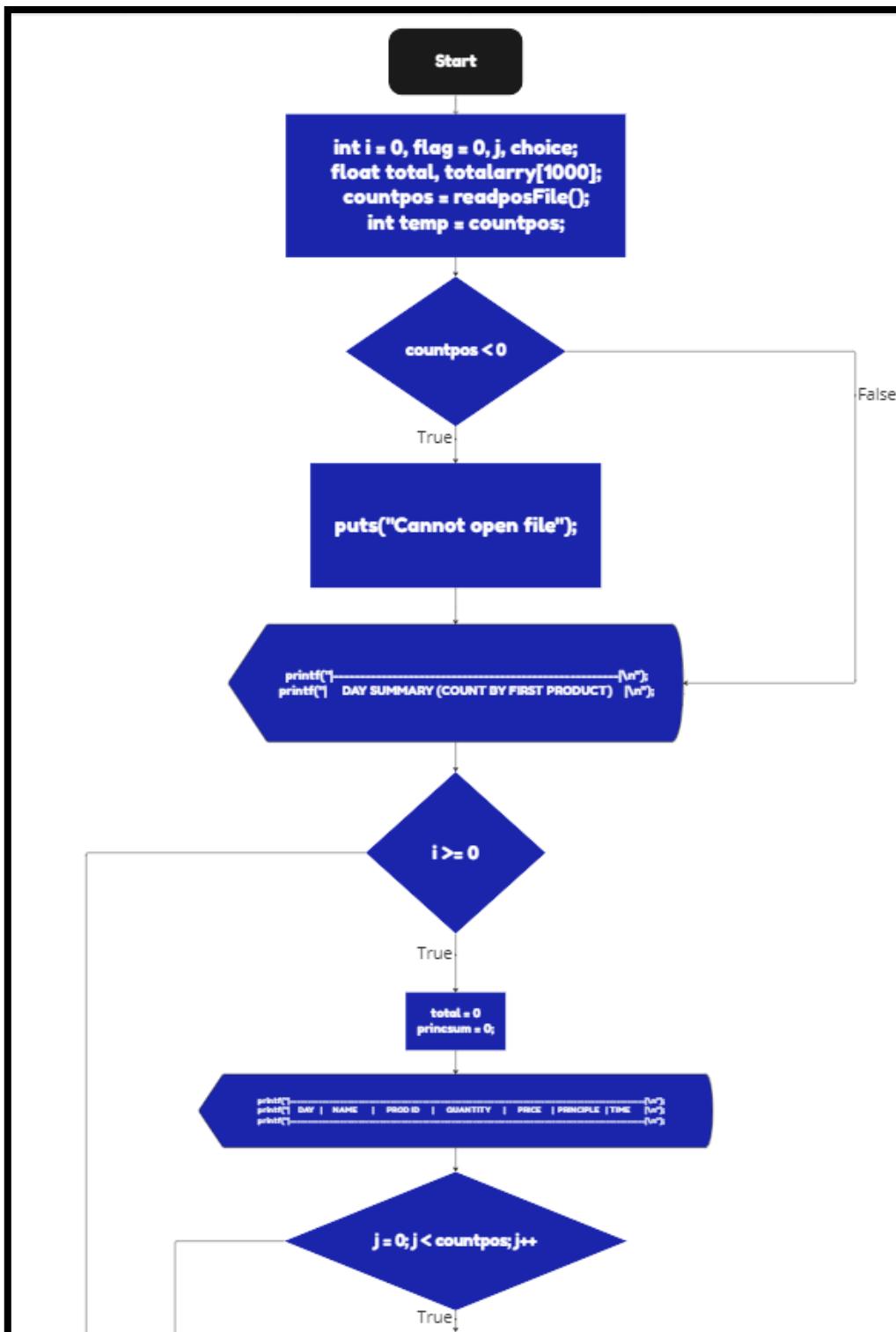
Void Purchaseproduct (2)



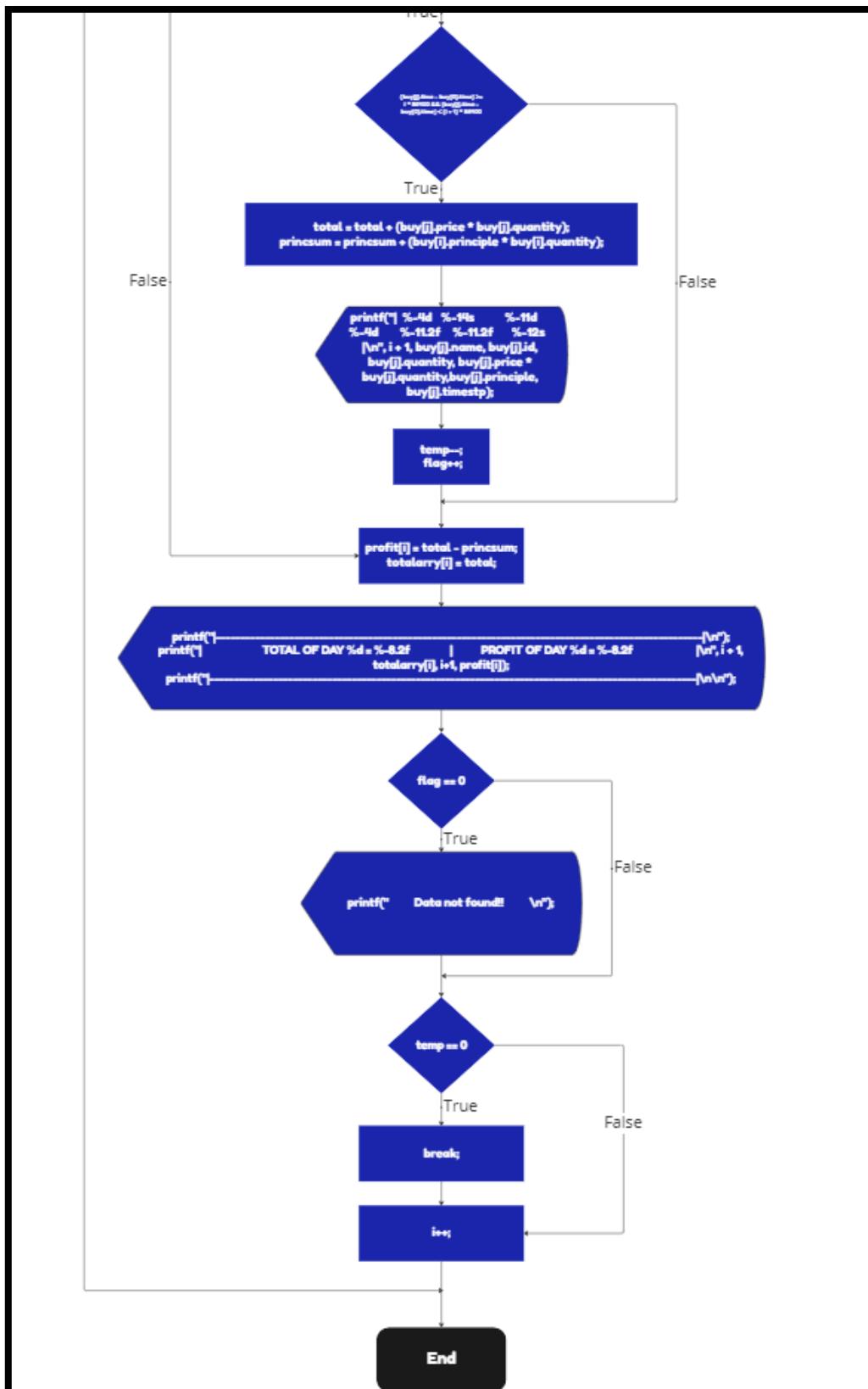
Void Purchaseproduct (3)



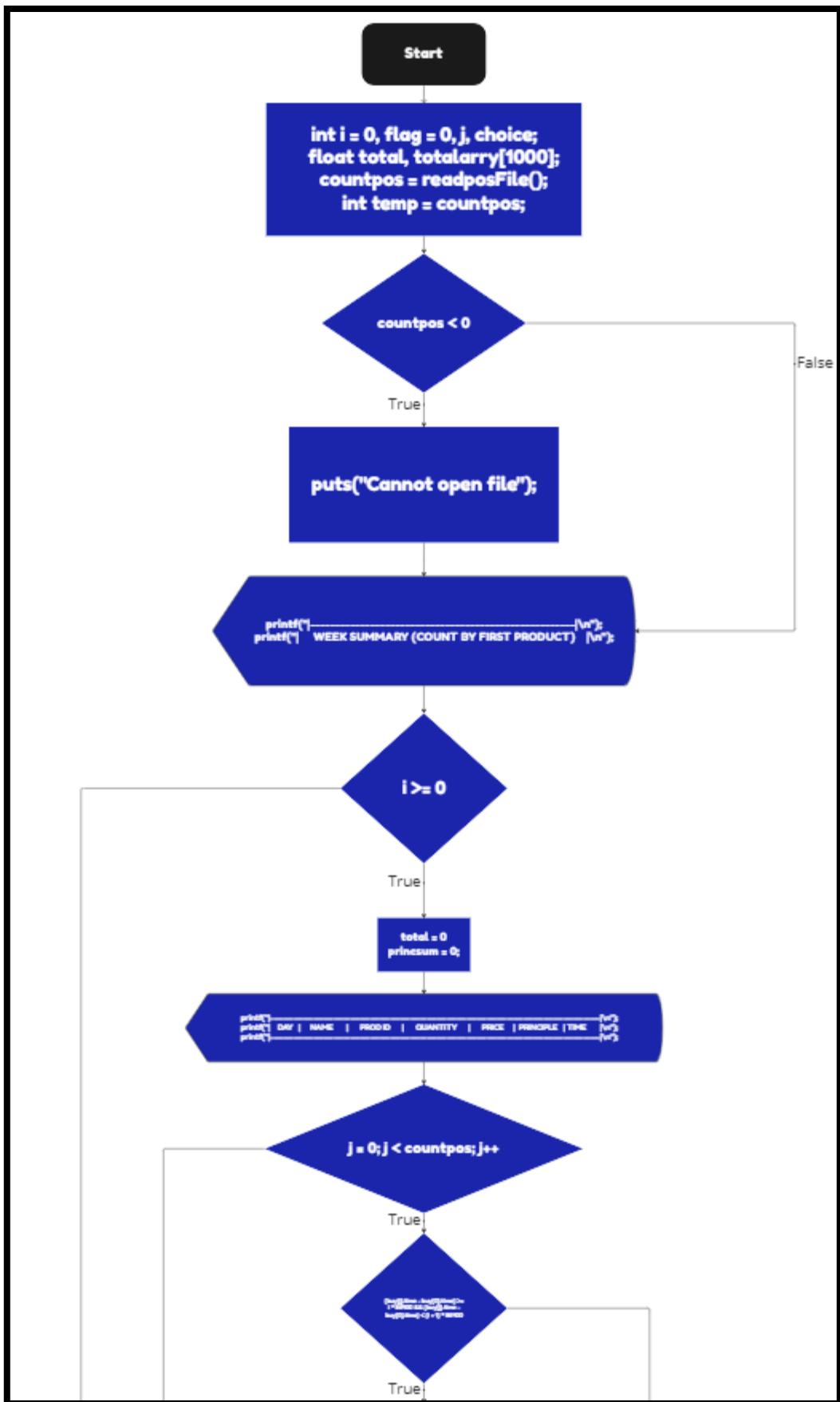
Void Purchaseproduct (4)



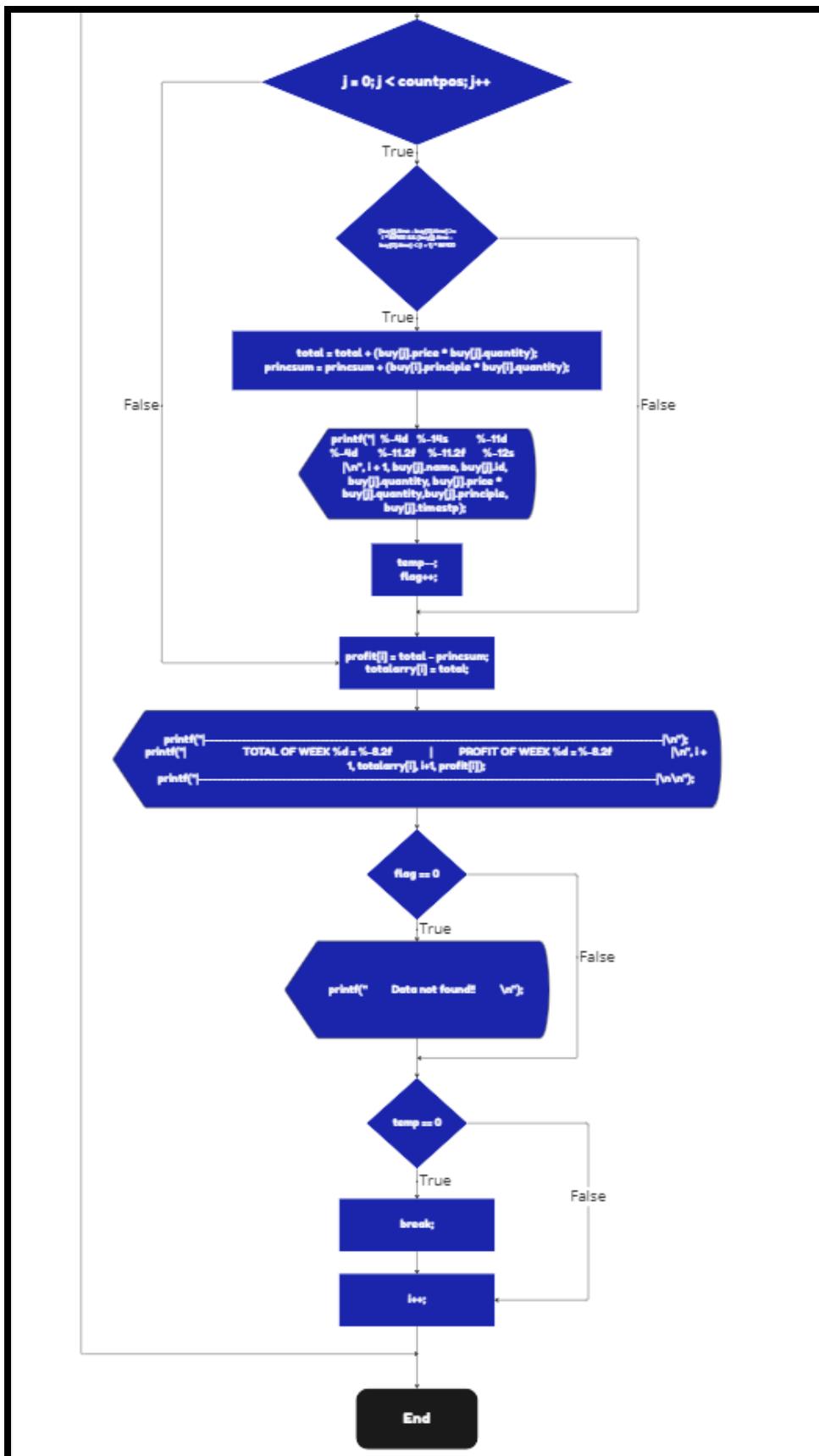
Void Admindaily (1)



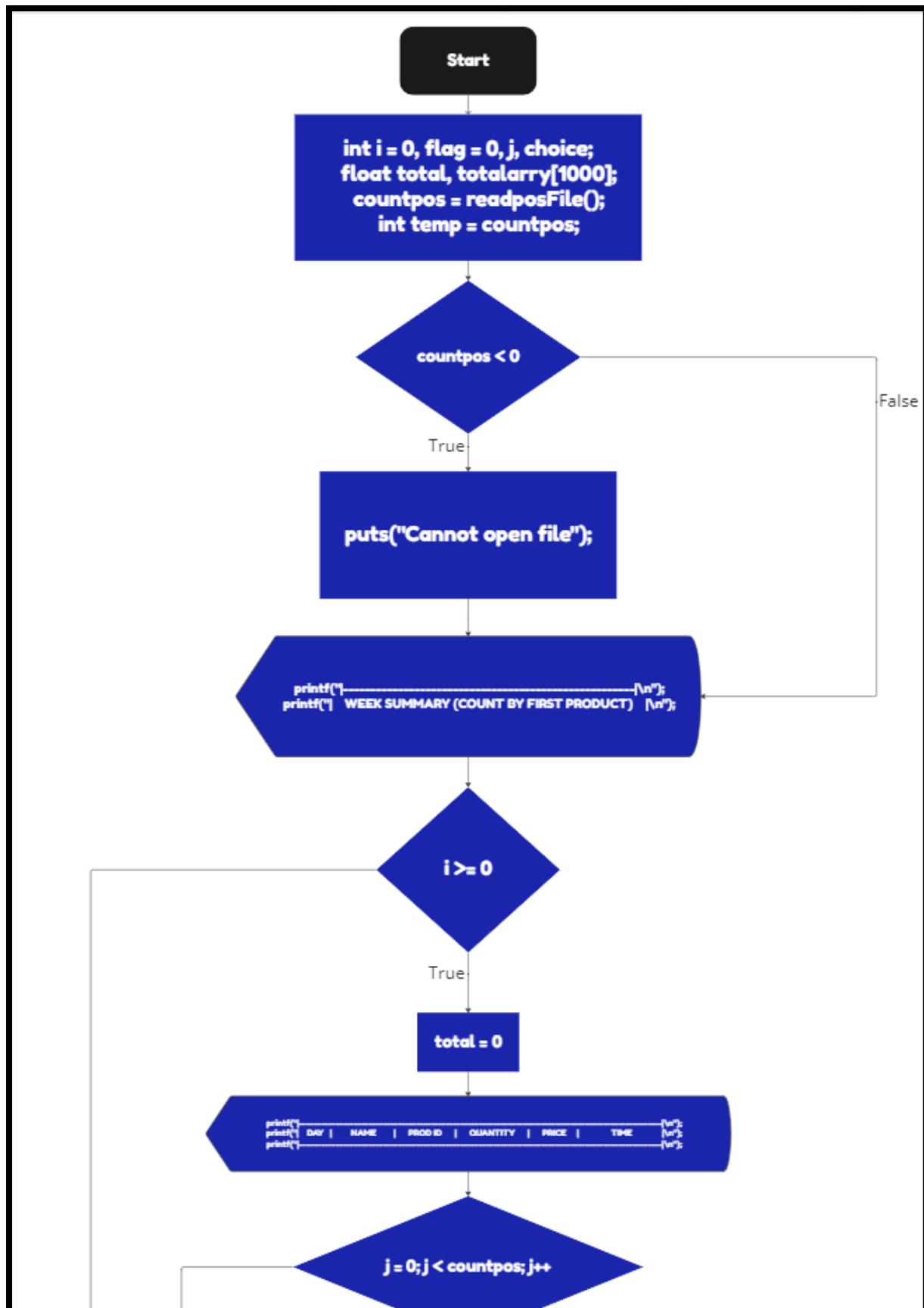
Void Admindaily (2)



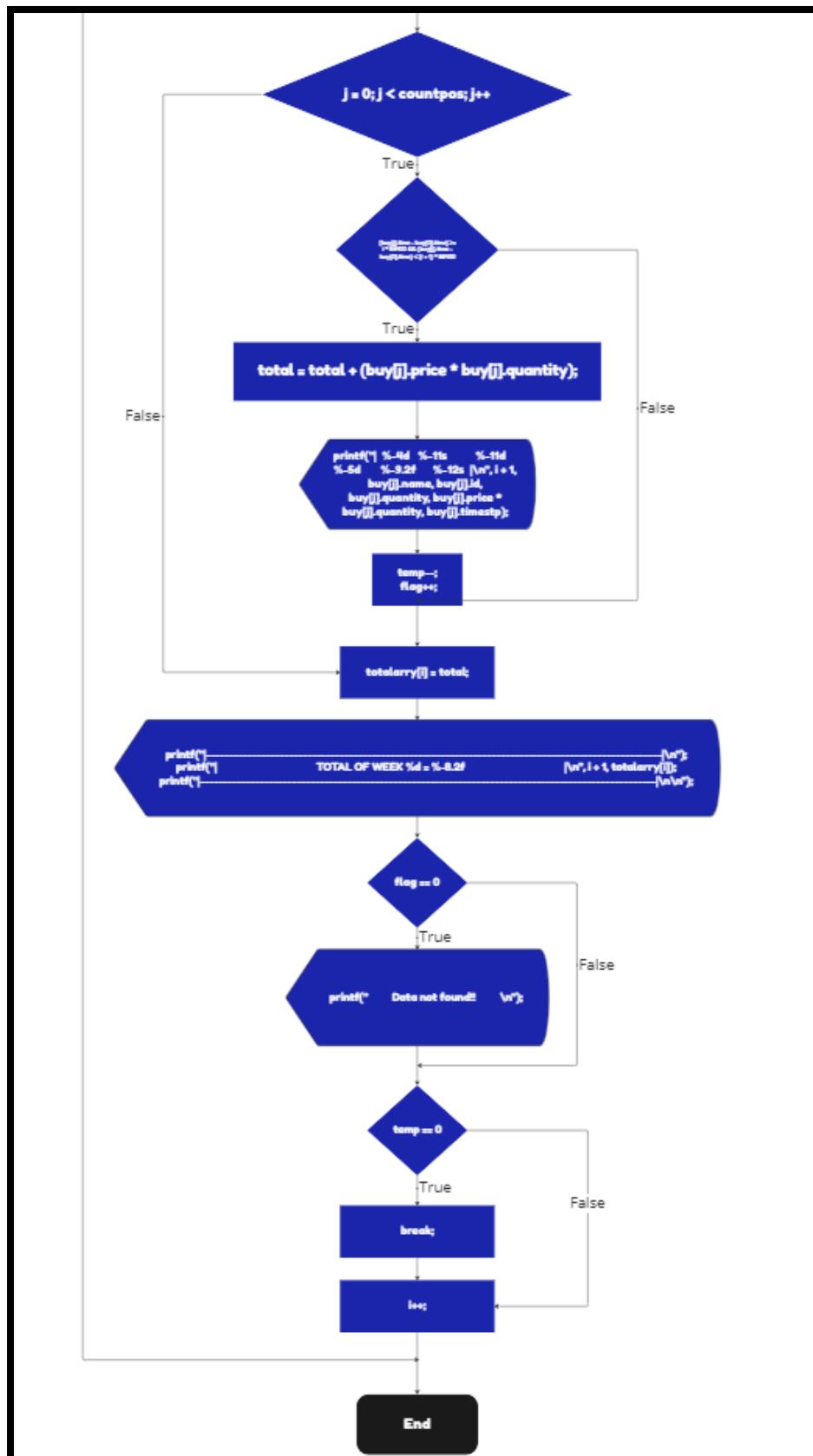
Void adminweekly (1)



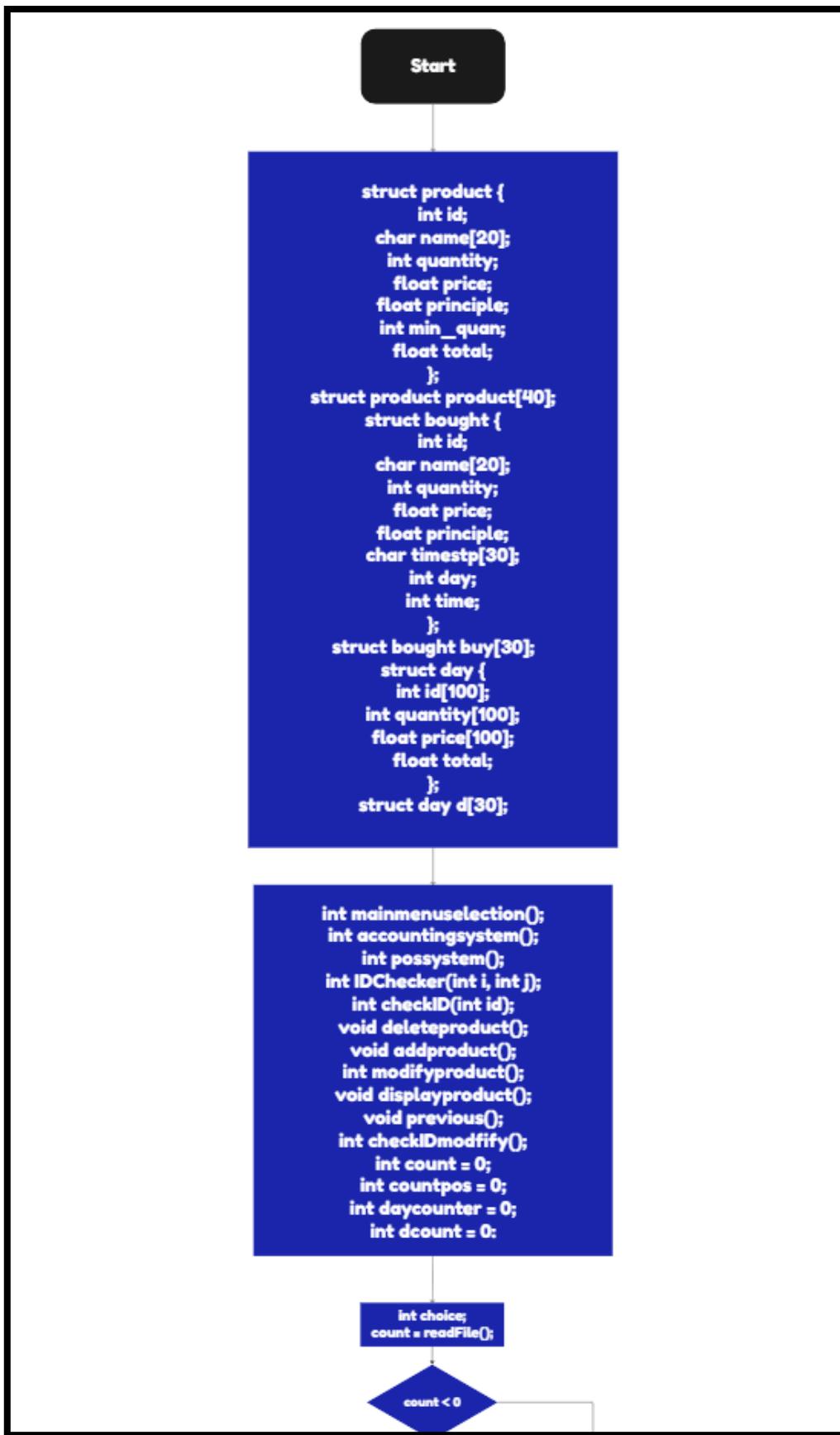
Void adminweekly(2)



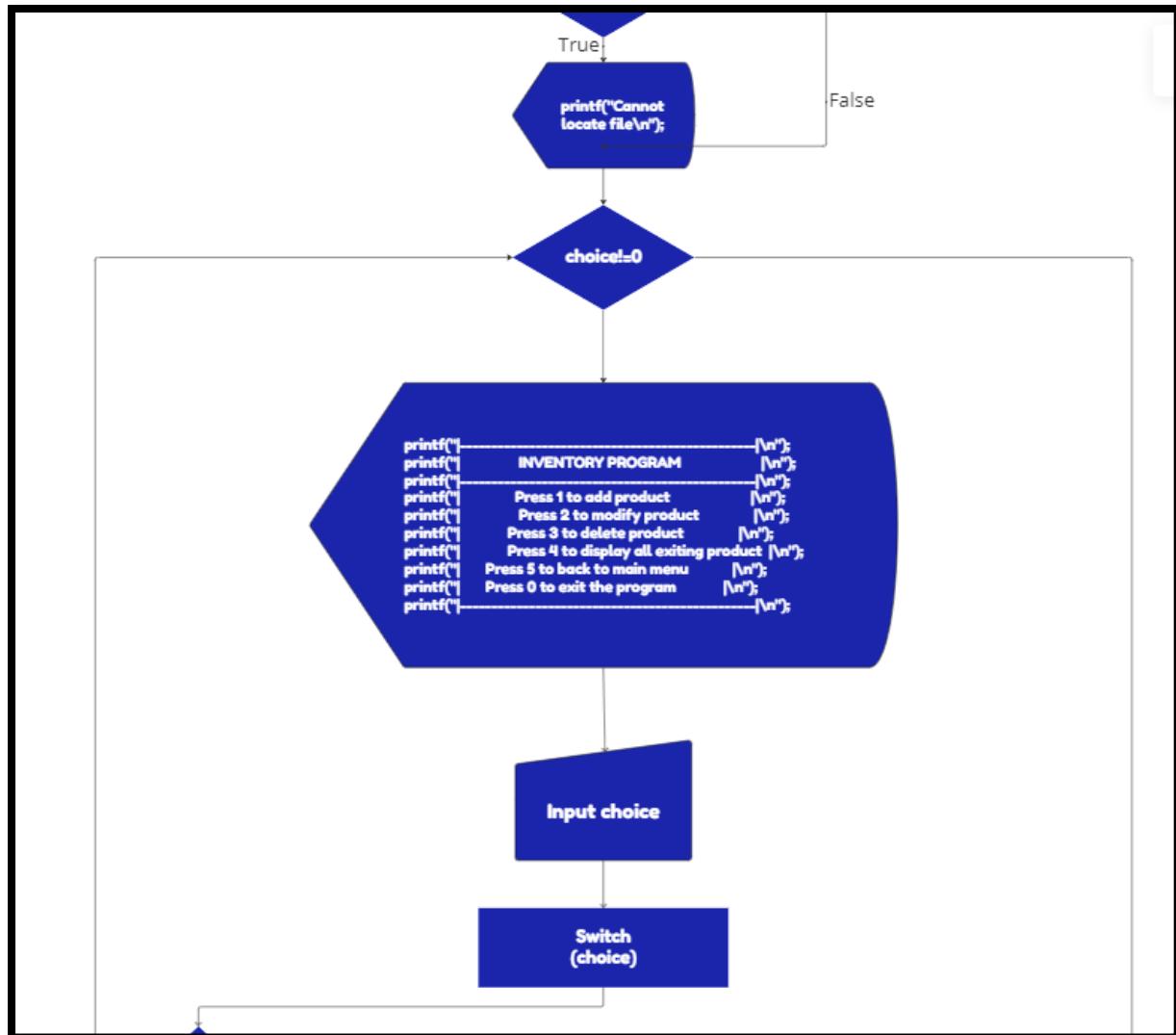
Void weekly (1)



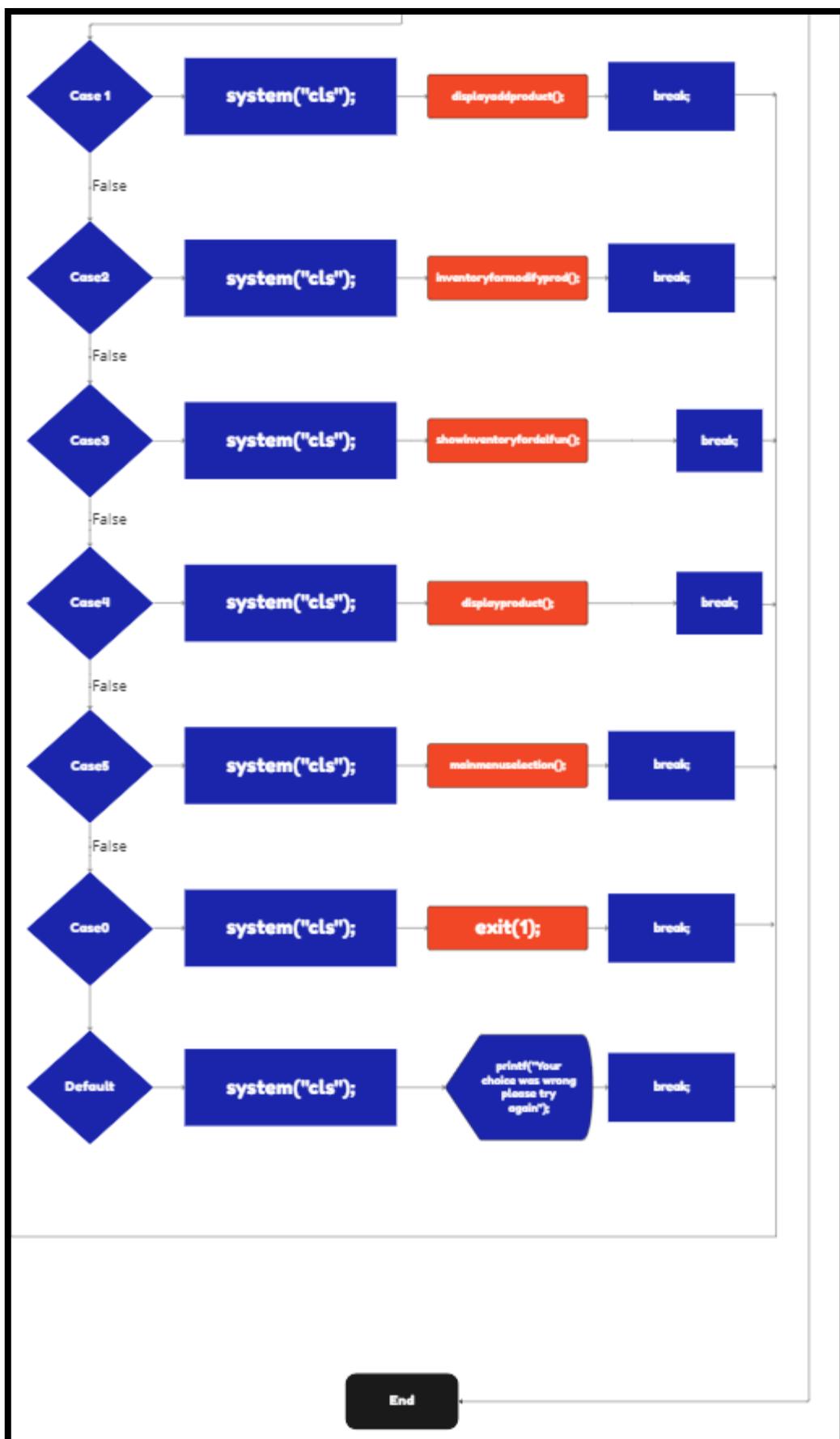
Void weekly (2)



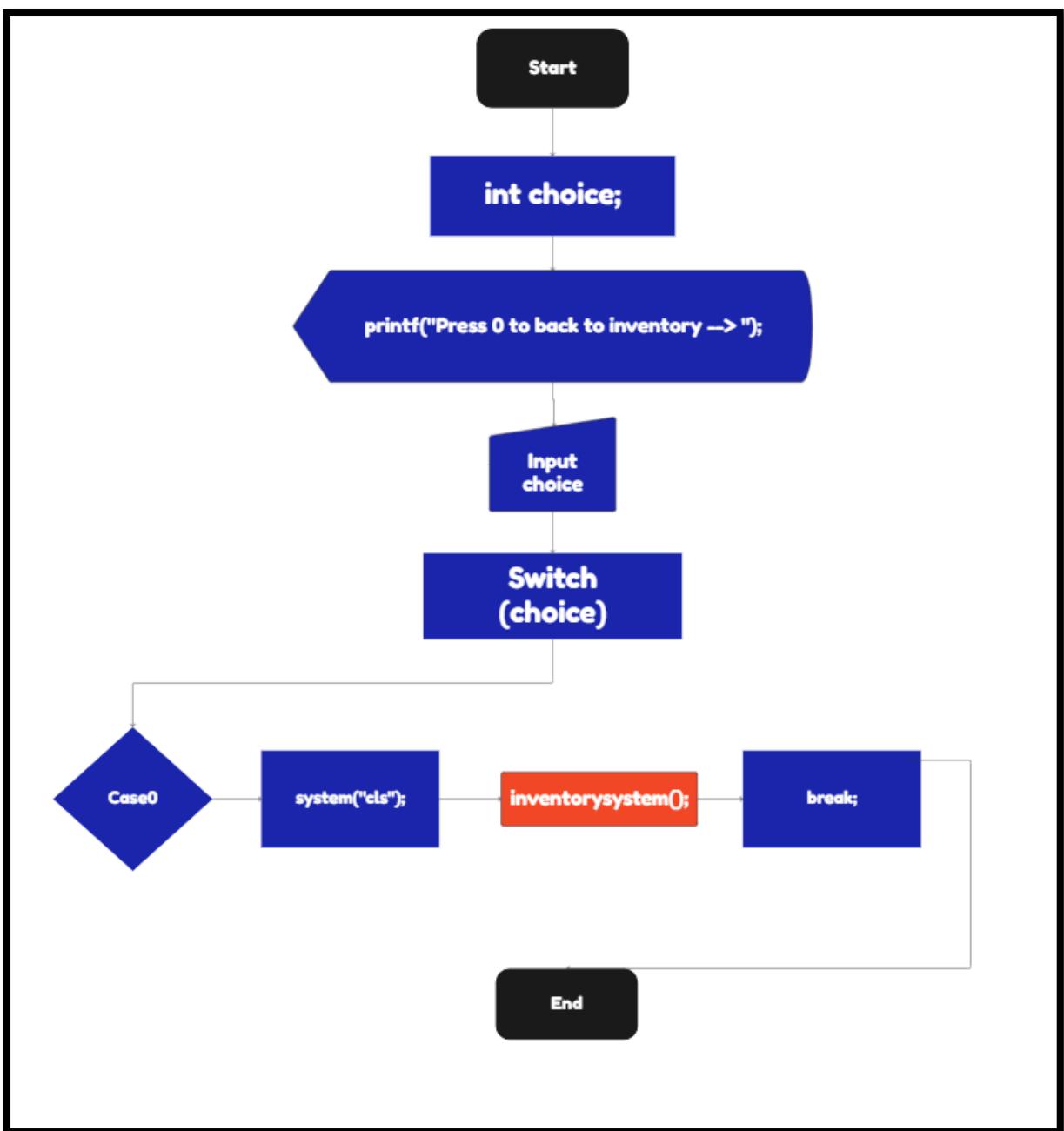
Int inventoriesystem (1)



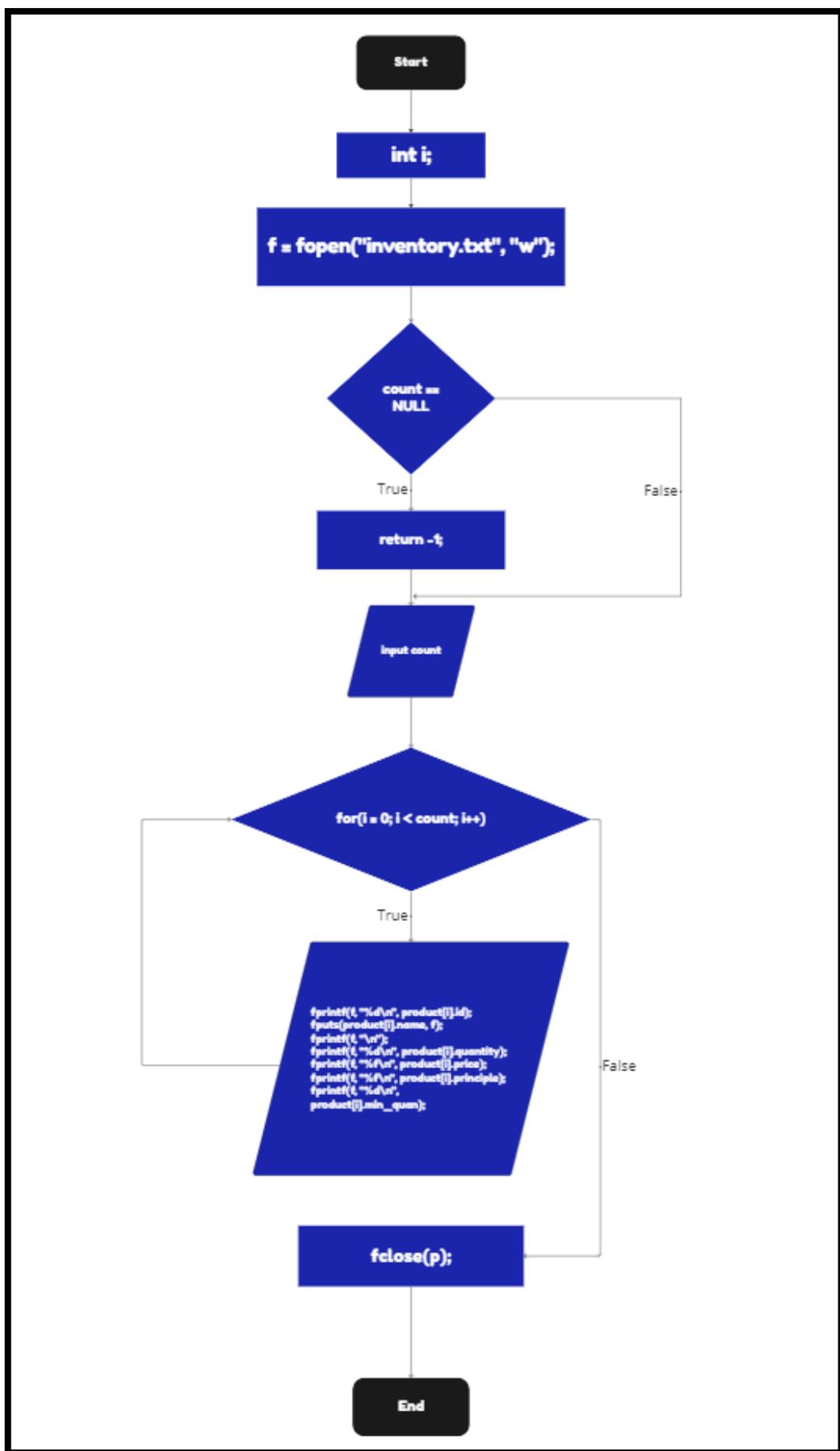
Int inventoriesystem (2)



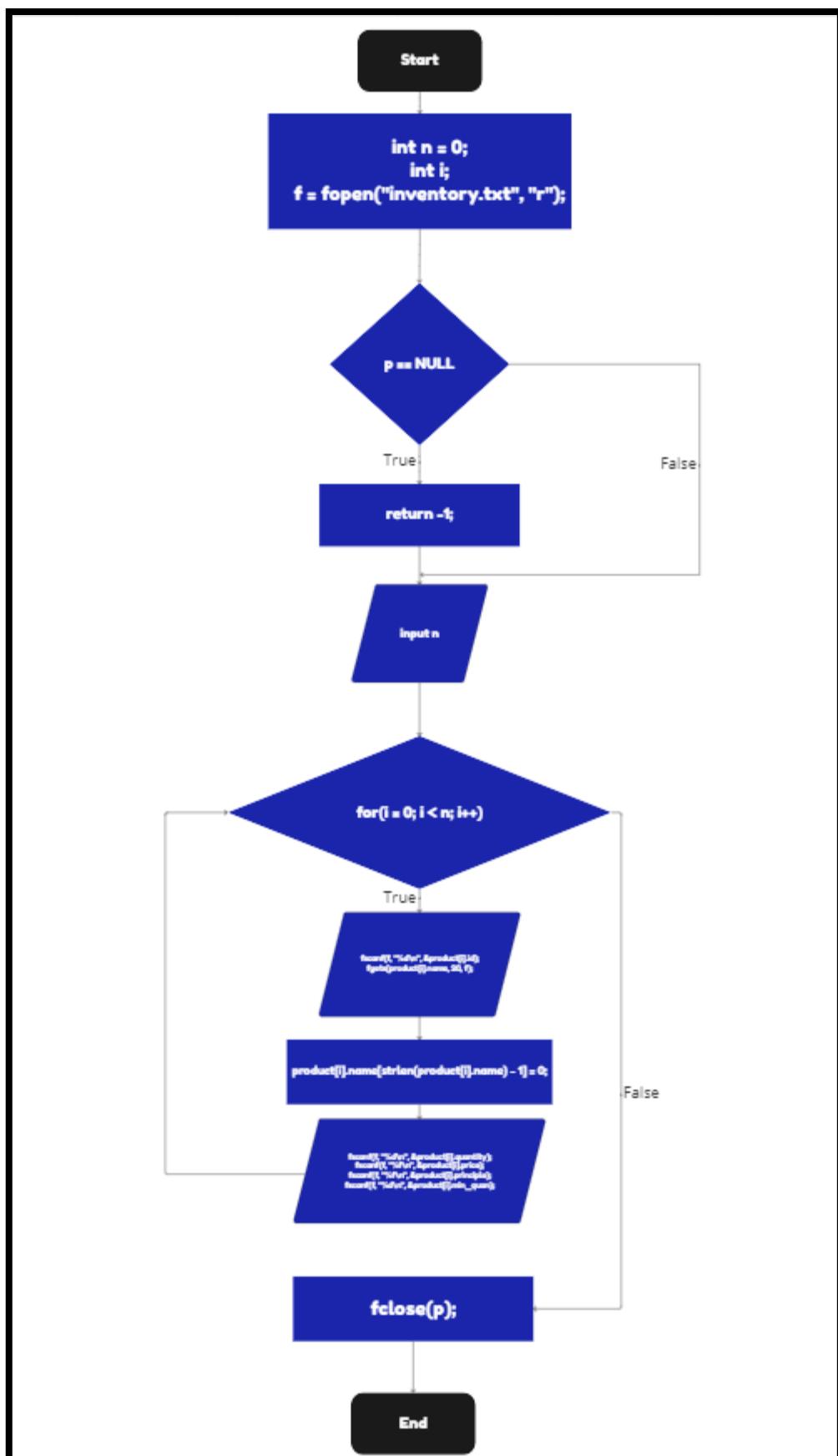
Int inventoriesystem (3)



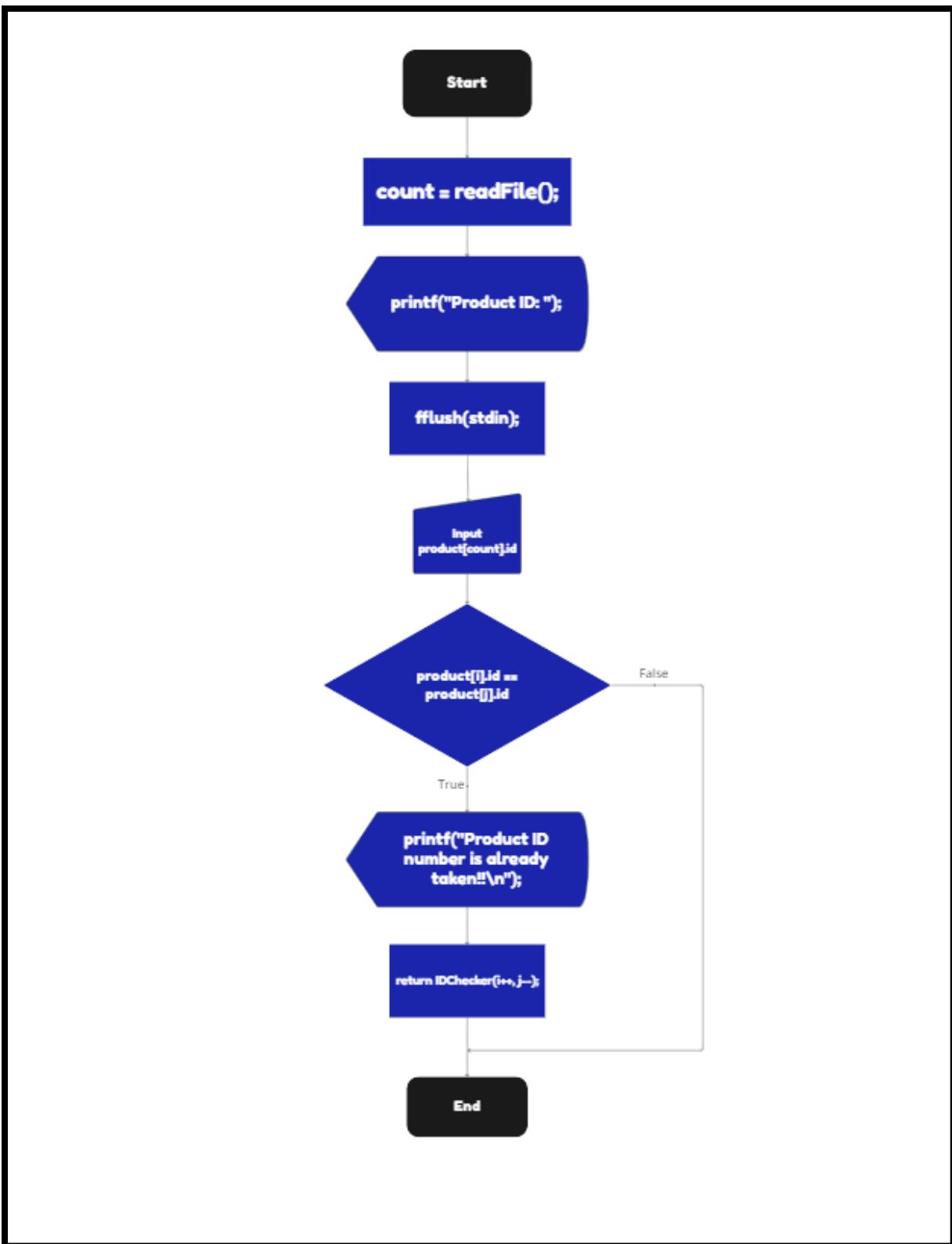
Void Previous



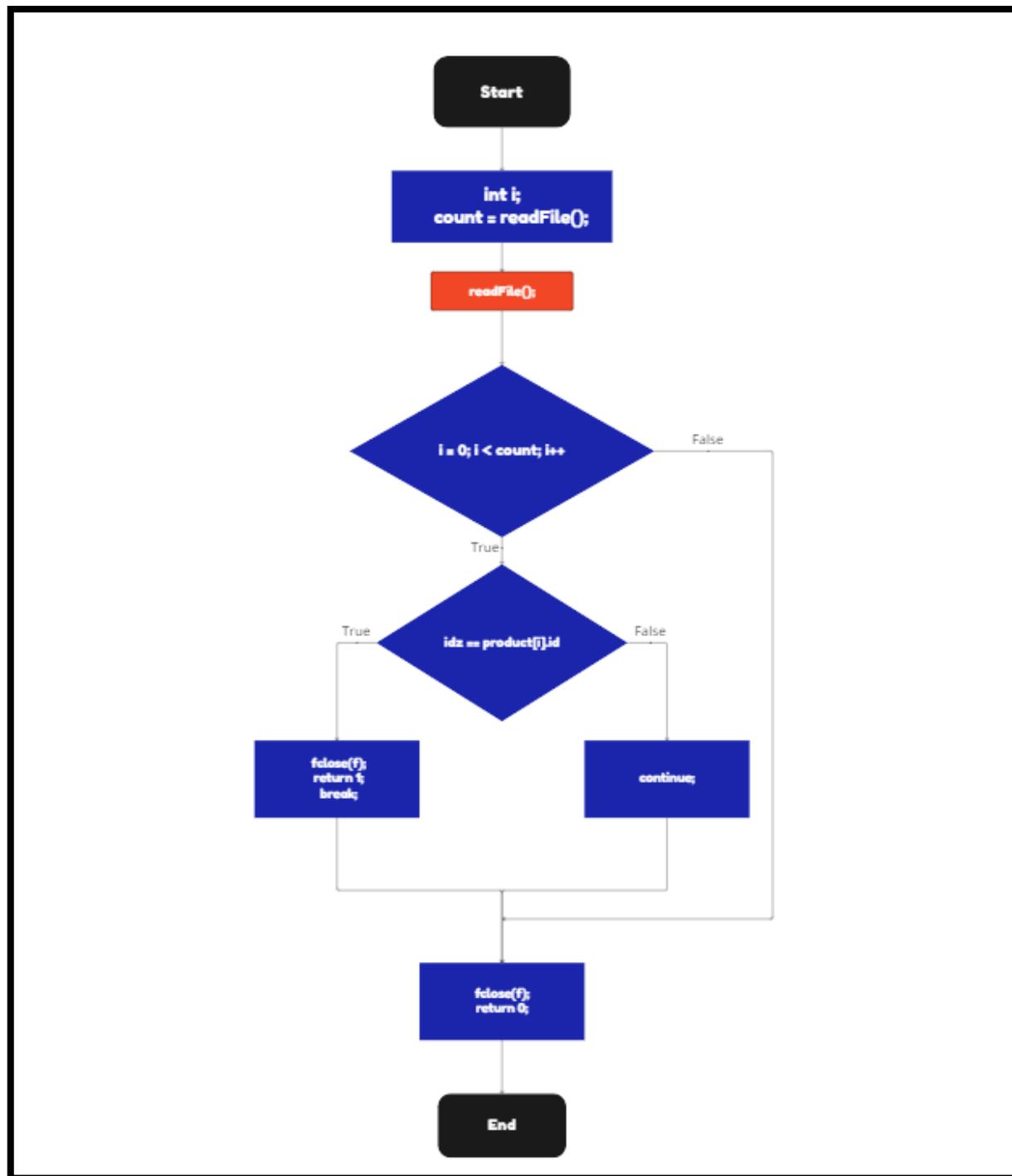
Int writefile



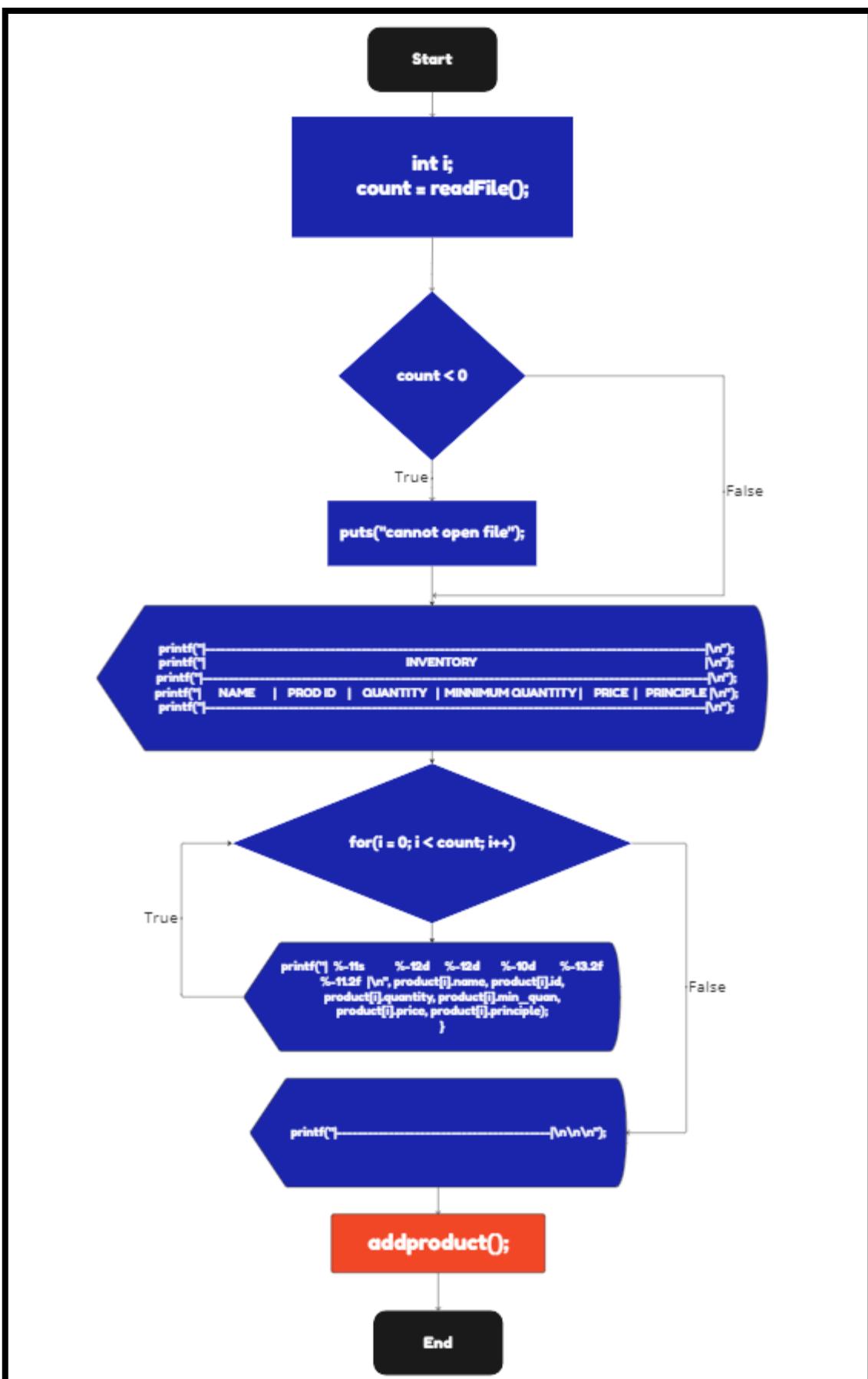
Int readfile



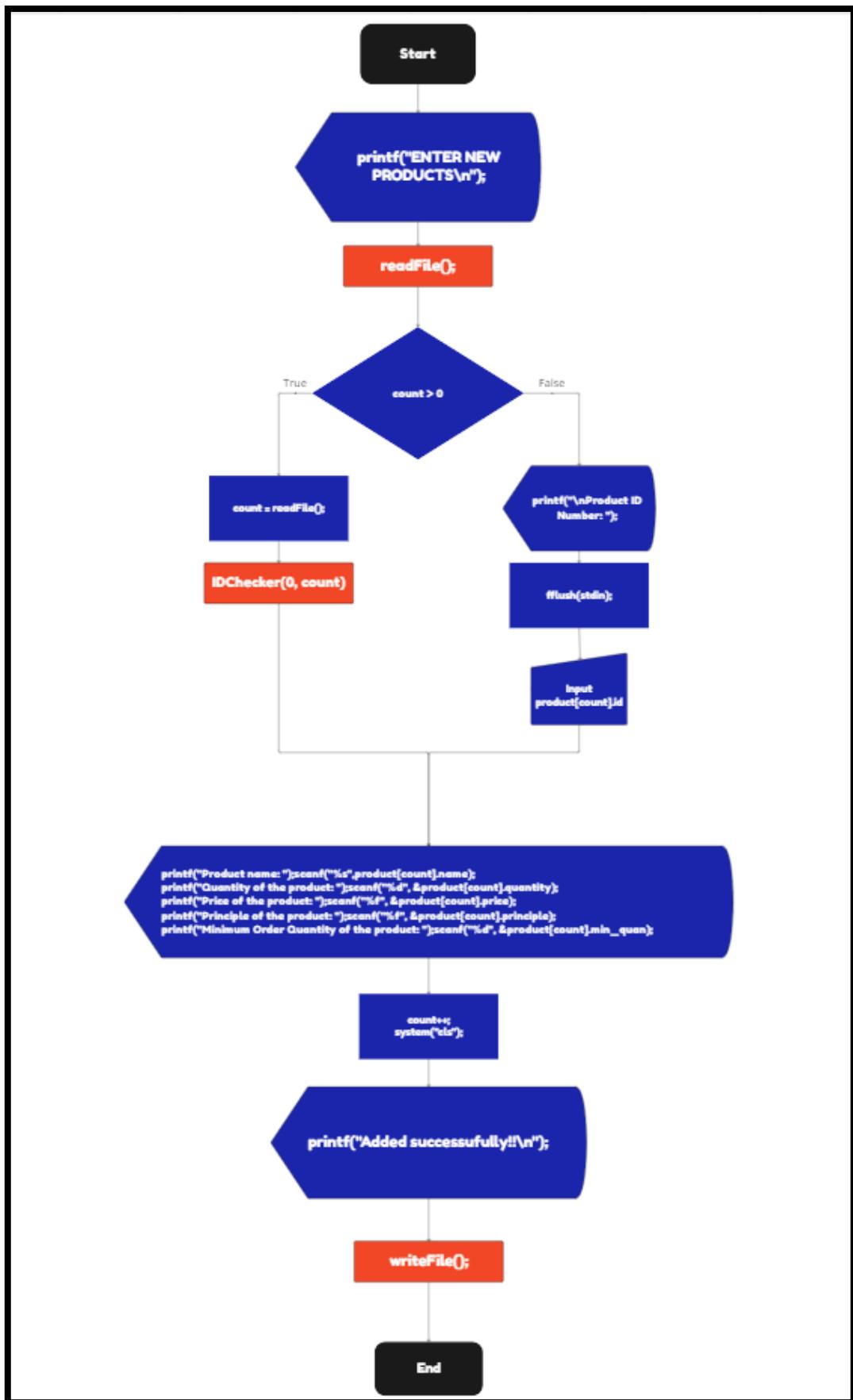
Int IDChecker(int i, int j)



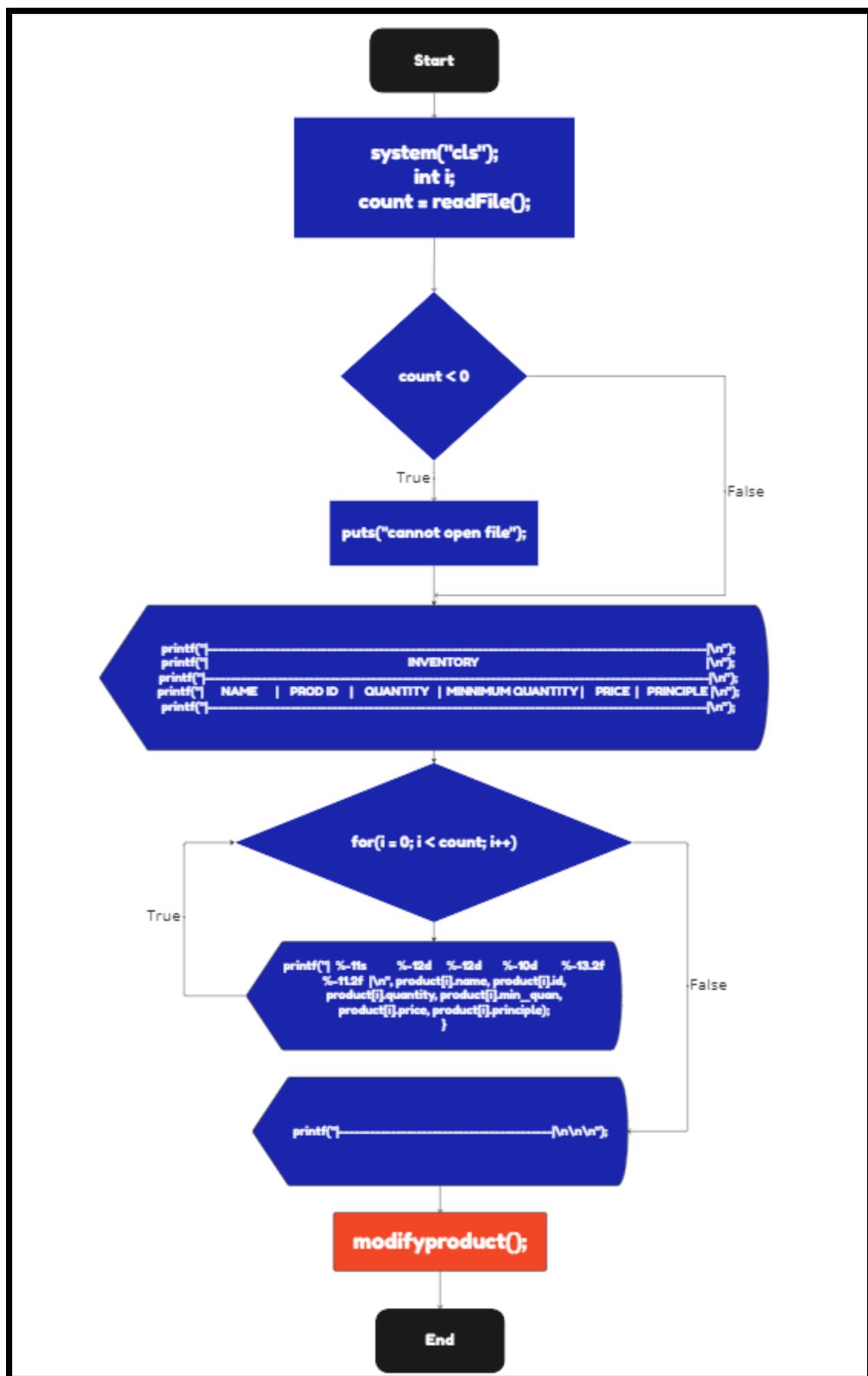
Int checkID(int idz)



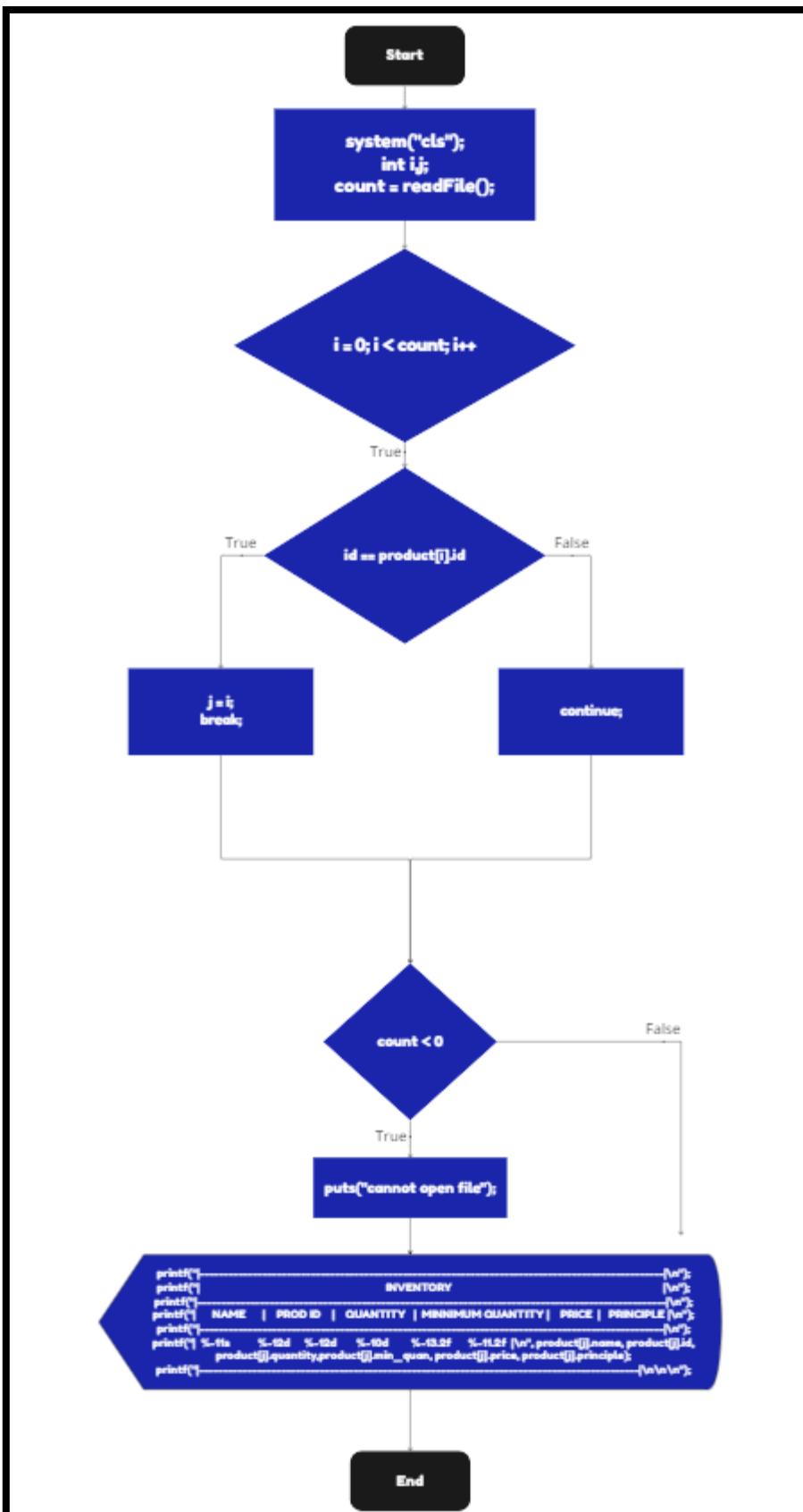
Void displayaddproduct



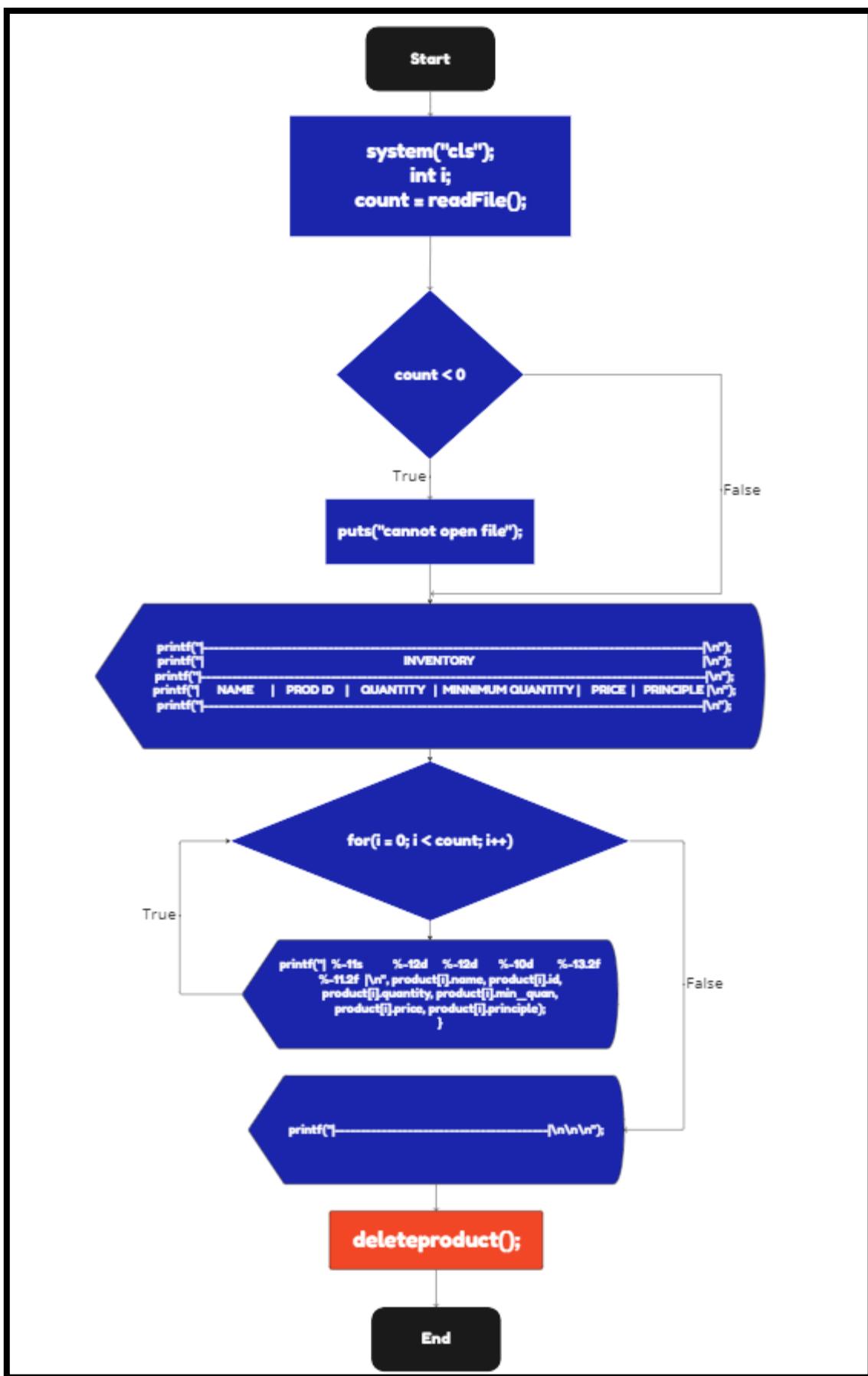
Void addproduct



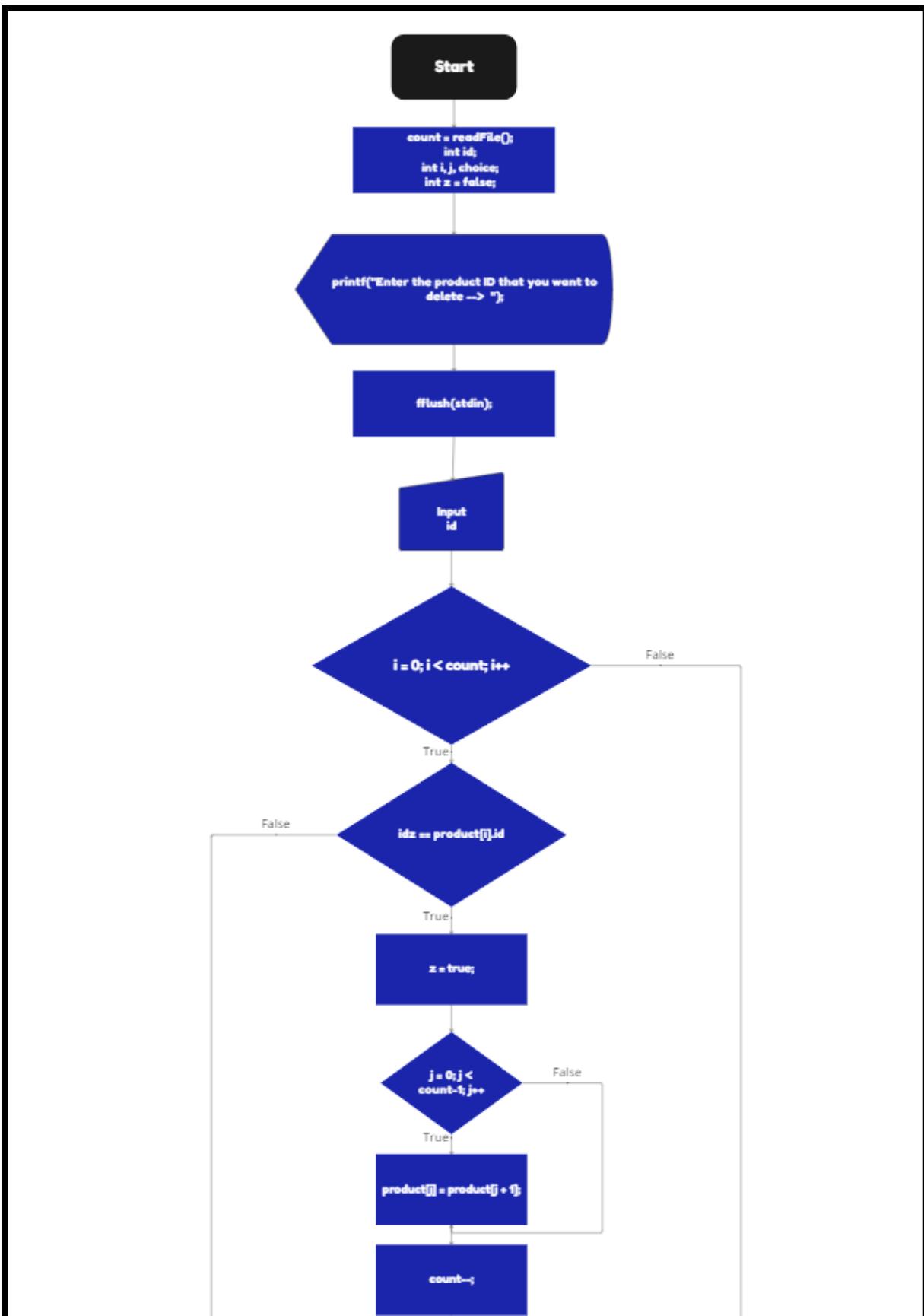
Void inventoryformodifyprod



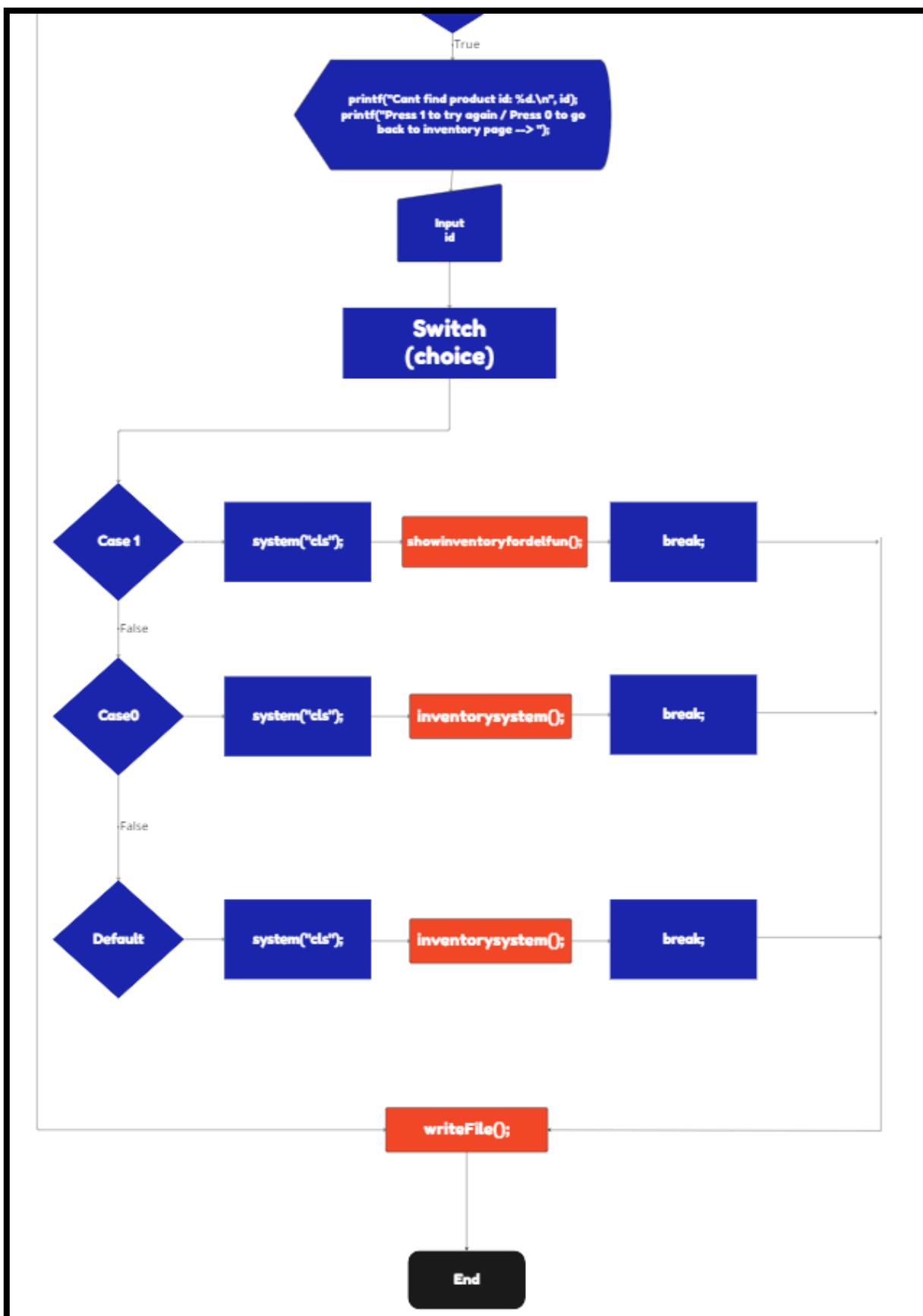
Int showinveninmodify(int id)



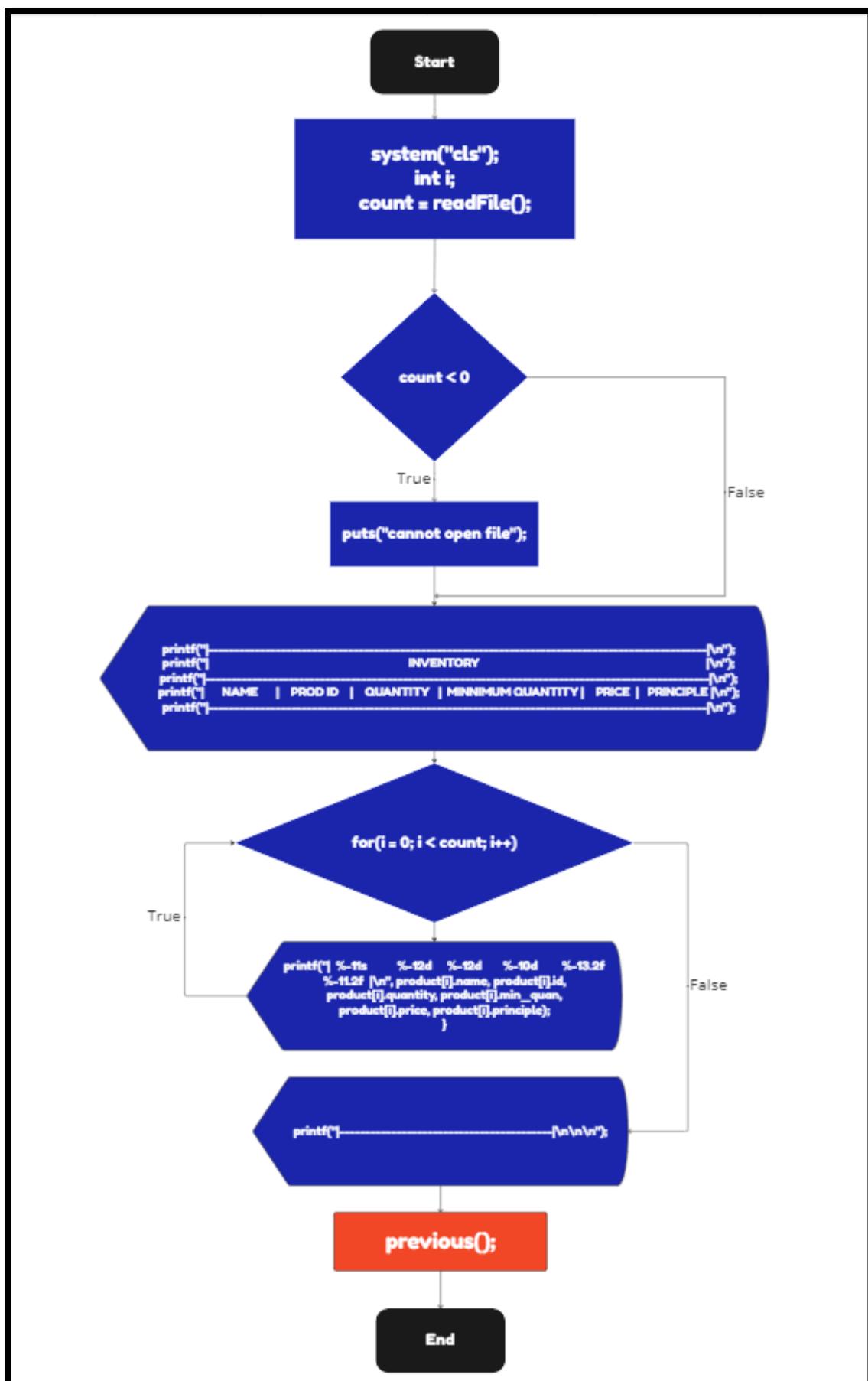
Void showinventoryfordelfun



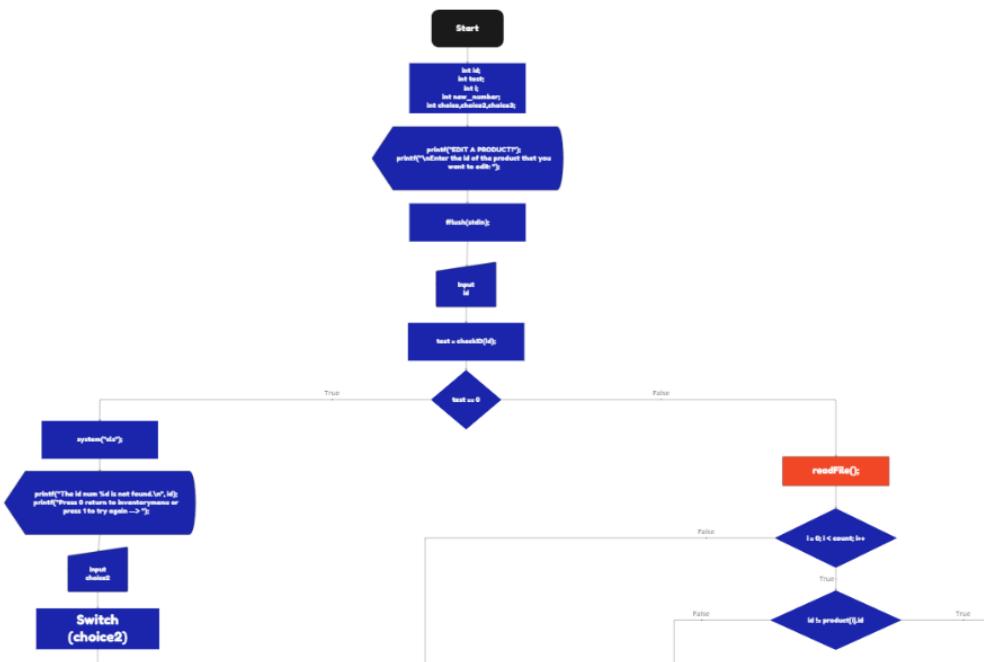
Void deleteproduct (1)



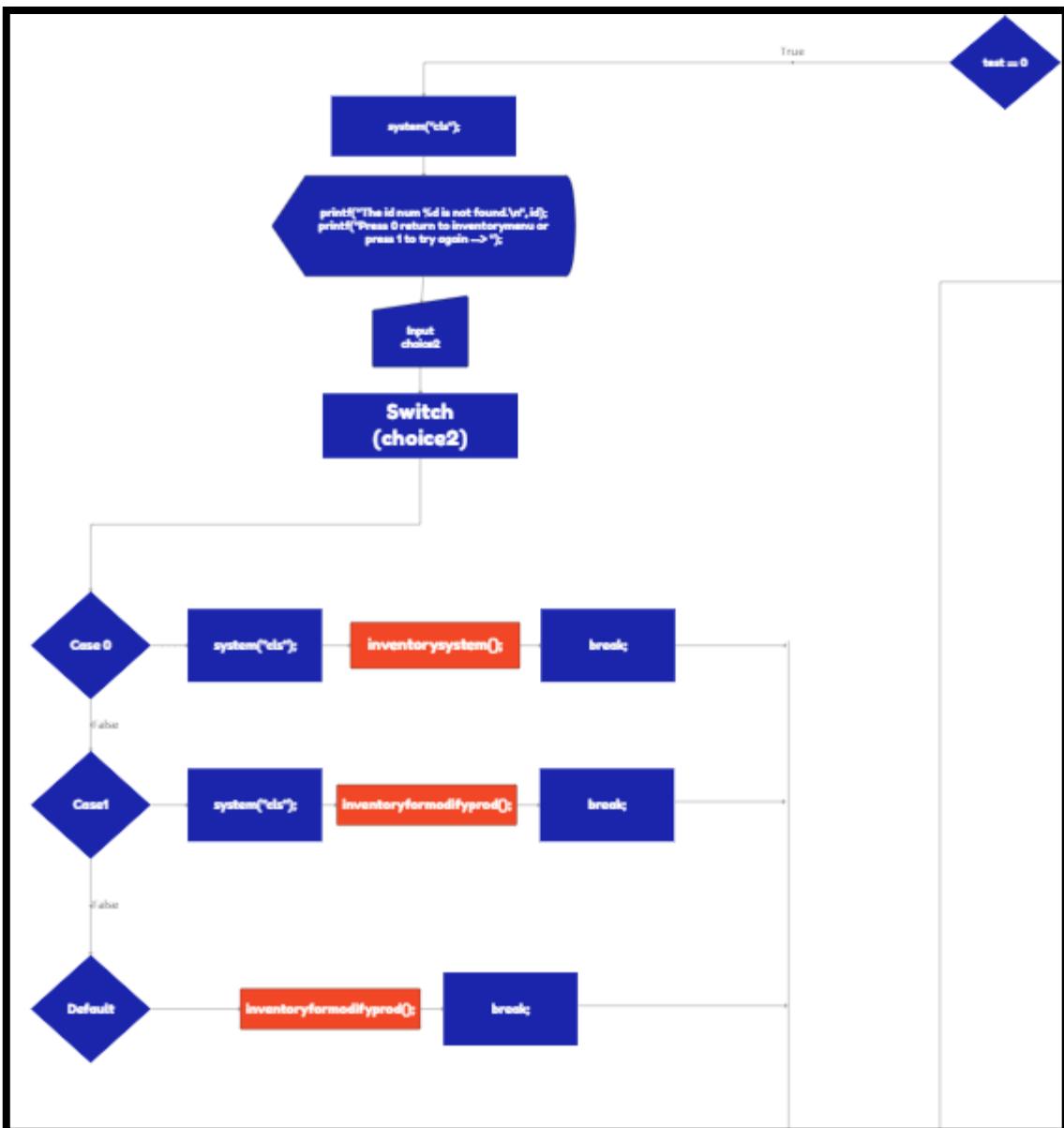
Void deleteproduct (2)



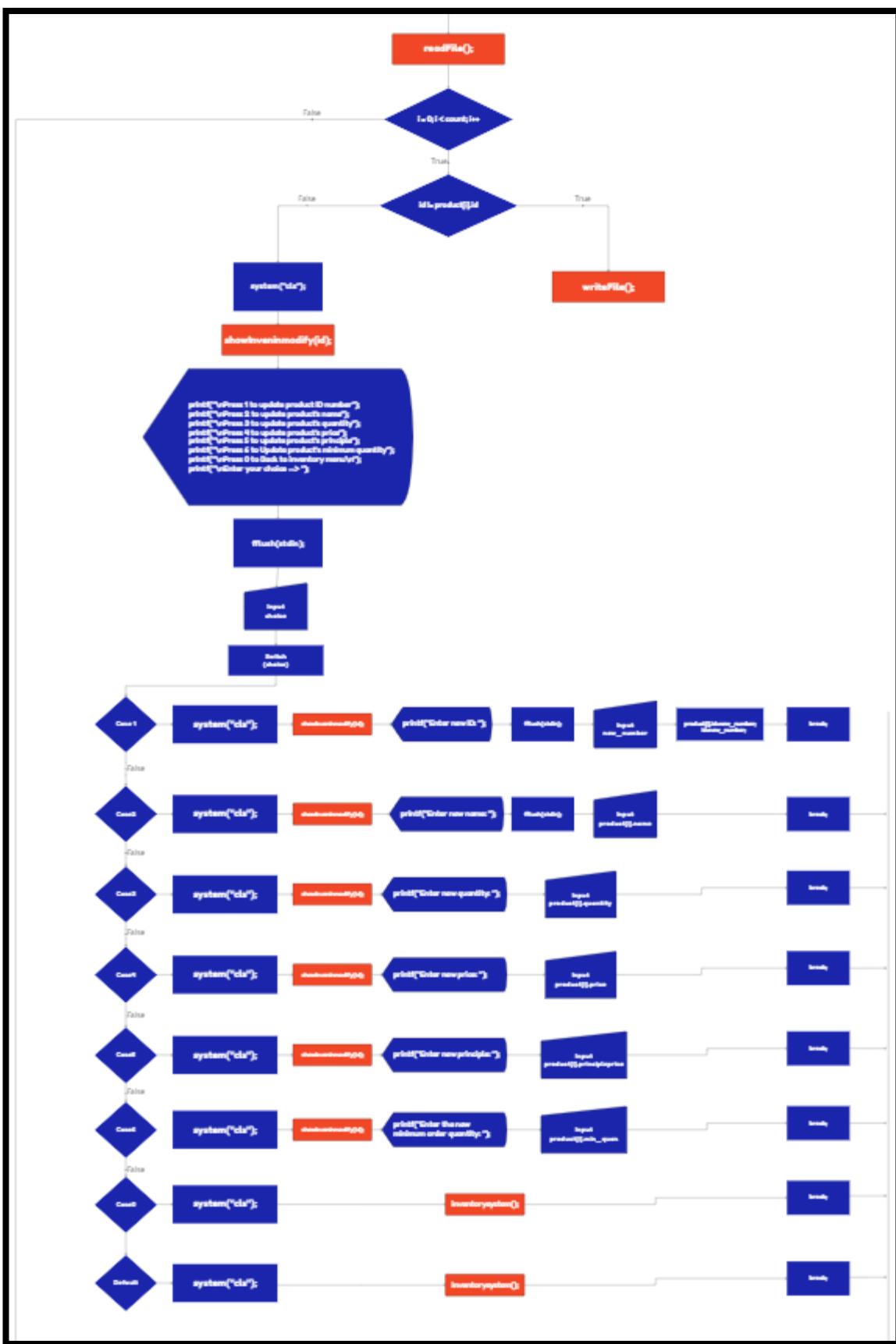
Void displayproduct



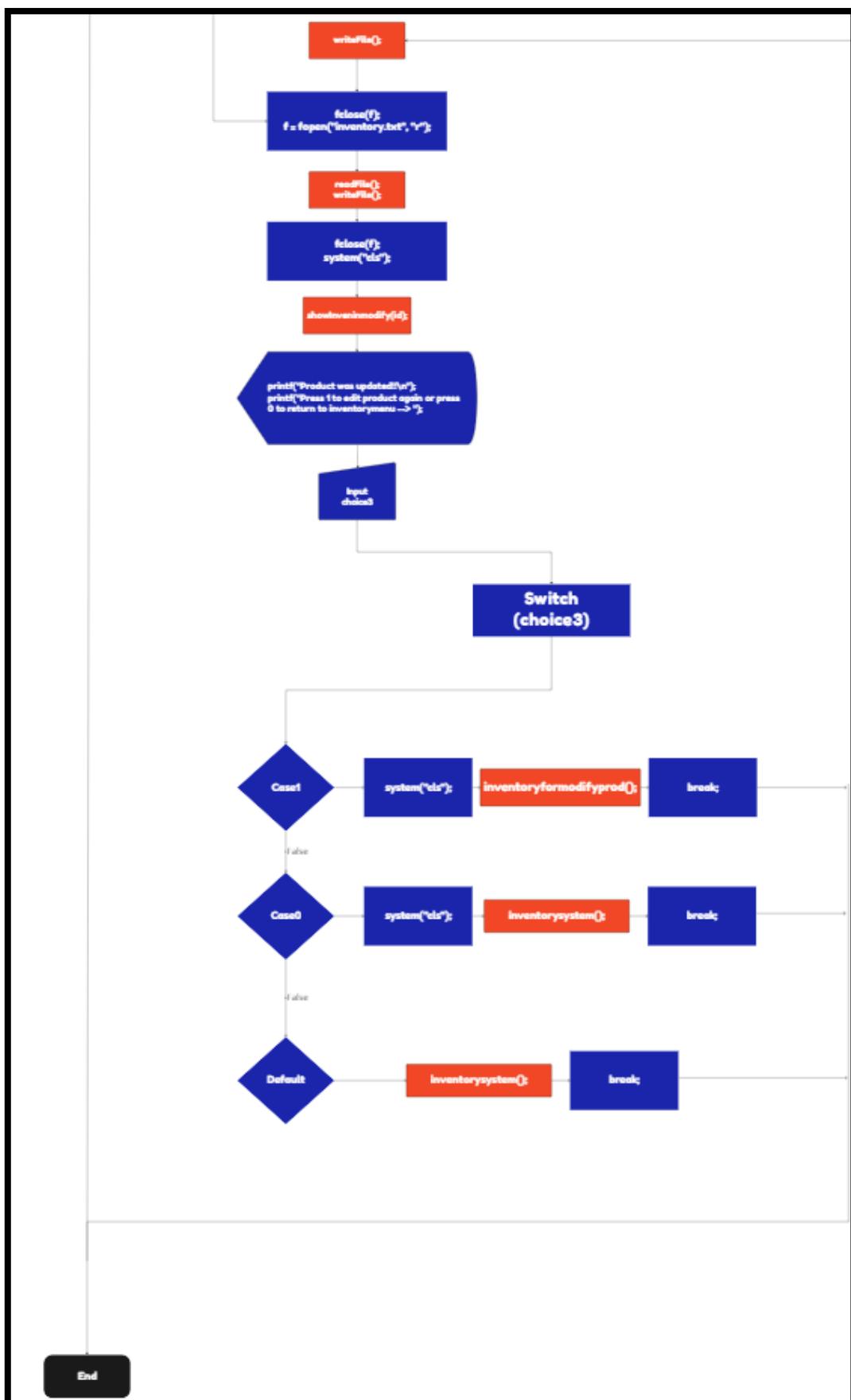
Void Modifyproduct (1)



Void Modifyproduct (2)



Void Modifyproduct (3)



Void Modifyproduct (4)

How program work?

On our landing page, there is a straightforward switch that can be used to start a program, navigate to the login page, or end it. This page, which we refer to as the login page, is reached if the user presses 1 on the landing page. Additionally, it has a basic switch casing. This function has three possible outcomes: client menu, admin login, and program exit. The main menu page, which has a simple switch case, is the page that the user will reach if they press 1 on the login page. In this function, there are three possible outcomes. Press 1 to access the POS system, 2 to return to the login page, and 0 to end the process. After you have pressed 1 in the client menu, you will be brought to the point of sales system which contains a simple switch case. There are 6 cases, in the first case when you press it, it will play 2 functions which are viewproduct and previous pos function. In the second case, It will bring you to purchaseproduct function. In the third case, user will be brought to daily function which contain daily summary and after daily function end, it will bring you to previousdaily function. In the fourth case, user will be brought to weekly function which contain weekly summary and after the function end, it will bring you to previous weekly. In the fifth case, user will be brought back to clientmainmenuselection function. And lastly is the exit program case. After the user has pressed 1 will come to the Viewproduct function. This product will show name, product id, quantity, minimum quantity and price of all products that are in the inventory.txt file. inventory.txt is a file where we store a struct list of products. After this function display all products, it will bring you to the previous pos function which will ask you to press 0 to go back to possystem. From main POS page, if you press 2 it will take you to a purchase product function to allow you to start buying product. Process of buying item including, first the program will ask you for the product ID that you want to buy. Next the program will ask you for the confirmation that if you are sure to buy this product lets press 1 or if you not sure let's press 0. If you press 0 the program will restart the purchase product function and it will let you select the product ID again. But in case of you press 1 it will ask you for the

quantity of the product that you want. After you typing the quantity into it and if the quantity that you have typing have more than or equal to the minimum quantity of the product and the quantity of the product it will show you message successful purchase and there are 2 option showing up are press 1 to purchase another product and press 2 to back to POS menu if you press 1 it will allow you to buy another product but if you press 2 it will take you to POS menu page. If the user press 3 in the point of sales menu it will bring you to the daily summary function. This function will show a daily summary by using a while loop in which int i is declared as day number. We will let countpos = readfile() in order to store the number of data set in countpos and use it in the for loop below. In the for loop there will be a condition which sets buy[0].time as a starter of the time counter. We will use the range of

$i*86400 \leq \text{buy}[j].\text{time} - \text{buy}[0].\text{time} \leq 86400*(i+1)$ to get the duration between days. And in the if condition it will print out the day, name, product id, quantity, price and time which are bought and also total price(outside for loop) but still inside while loop. For total, we use normal integer total to store total value each day then store in total array with days index and redeclared int total to be 0 to store next day total value and print out. If the user presses 4 in the point of sales menu it will bring you to the weekly summary function. This function will show weekly summary by using a while loop which int i is declared as week number. We will let countpos = readfile() in order to store the number of data sets in countpos and use it in the for loop below. In the for loop there will be a condition which sets buy[0].time as a starter of the time counter. We will use the range of

$i*604800 \leq \text{buy}[j].\text{time} - \text{buy}[0].\text{time} \leq 604800*(i+1)$ to get the duration between weeks. And in the if condition it will print out the week, name, product id, quantity, price and time which are bought and also total price(outside for loop) but still inside while loop. For total, we use normal integer total to store total value each week then store in total array with week's index and redeclared int total to be 0 to store next week total value and print out.

(This program will be the same as daily summary, just change the time integer to multiply by 640800). After user press 2 in login page it

bring user to login as admin page user must enter username and password correctly if user enter wrong password it will bring user to previouslogin function this function will ask user want to retry enter username and password again or want to back to login menu, and we use switch case to do what the user chooses. After login in, the admin login page will bring the user to the admin menu page. This page is contain menu or function that the user wants to manage their shop. If user press 1 will go to inventorysystem. if user press 2 will go to accountingsystem. if user press 3 will go to daily summary for admin. If user press 4 will bring the user to weekly summary for admin. if user press 5 will go back to the login page. if user press 0 it will exit the program. After login to admin menu and pressed 1, you will be brought to inventory system menu. In this function count will be assign to equal to readFile() in order to get number of data set and if it is lower than 0 it will said cannot locate file but if it does satisfy the condition, it will print out the inventory menu which have 6 cases and it will clear screen every time we picked number and enter it. First, it will bring you to displayaddproduct function which show you existing product and let you add more if you want. Second, it will bring you to inventoryformmodifyprod function which will first show you the current inventory and let you modify. Third, will play the function showinventoryfordelfun which will show you the current items and let you delete it. Fourth, this case will bring you to displayproduct function which will show you current product. And the last one is for exiting the program. After pressed 1 at inventory system, it will first read file and bring you to displayaddproduct function which will start with IDchecking function checking if the id was taken or not after finish checking it will display all current product and play addproduct function to let you input sets of data which are product id, name, quantity, minimum quantity, price and principle. After that it will write a file by the writeFile function on inventory.txt. After pressed 2 at inventory system, it will bring you to inventoryformmodifyprod function which will first show you the current inventory and play the modifyproduct function which will get the id you want to modify and take to checkID function to check if it is equal to any product[i].id if it is, then continue then close file and return 0 otherwise it will close file , return 1 and break. Int Test will be assigned to equal to the return

number and if test is equal to 0 means there is no id that is the same to the input one and will let the user to input again if press 1 or exit program by press 0. After users input correct id, it will let users decide whether to modify id, name, quantity, price, principle or minimum quantity of product. And it will read a file and write then close and then show you the modified product in the table for the user to check if it is correct each time you finish editing it. After pressed 3 in inventory, It will bring you to showinventoryfordelfun function to show current items and after that you will be brought to deleteproduct function which before user input id to delete, there will be declaration for z to be false and when enter for loop and pass id check condition after enter id it will let z equal true and then delete the every sub data in in that product. If the program found that z is still false it will say it can find the product id and let the user choose to enter again or go back to inventory. In the inventory function, if you press 4 it will display all existing products. There will be Name, Product's ID, Product's quantity, Product's minimum quantity, Price, and Principle. You can press 0 that will take you back to the inventory function. After pressing 2 in the admin menu.it will bring you to the accounting system. and if user press 1 again will bring the user to display purchased lists. this function is only for admin this function will show bought list and the bought contain name productid quantity price principle and time that user buy the product and show total of income and profit since the first day until now after show all of this the program will ask user to press 1 to back to accounting system page or press 0 to exit the program. or user press 2 it brings the user back to the admin mainmenu. or if user press 0 it will exit the program. In this admin daily summary, It will all be the same as the client daily summary but there is principle and profit that are added. And in this admin weekly summary, It will all be the same as the client weekly summary but there is principle and profit that are added.

Section 2

Function

```
● ● ●

int main() { //main function of program
    printf("-----|\n");
    printf("CPE100 Module 3 : Online Shop Management System|\n");
    printf("-----|\n");
    printf("-----| Group Members |\n");
    printf("-----|\n");
    printf("-----| 65070503449 Thanaphat Ngoennet |\n");
    printf("-----| 65070503450 Thanakorn Soonjaw |\n");
    printf("-----| 65070503456 Parunchai Kochseni |\n");
    printf("-----| 65070503459 Phakalpol Maneesopa |\n");
    printf("-----| 65070503475 Kanchai Lerdsrisakulrat |\n");
    printf("-----|\n");
    printf("-----| Press 1 to enter the program |\n");
    printf("-----| Press 0 to exit the program |\n");
    printf("-----|\n");
    printf("\nChoice --> ");
    int main_menu;
    scanf("%d", &main_menu);
    switch(main_menu) { //switch case use int main_menu as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            login(); //run login function
            break;
        case 0:
            exit(1); //exit program
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("Your choice was wrong please try again!!\n\n\n");
            main(); //run main function
            break;
    }
    getch(); //input any key to exit program
    return 0;
}
```

```

void previouslogin() { //add more option for other function
    int choice;
    printf("Press 1 to try again / Press 2 to back to login menu: ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            admin_log(); //run admin_log function
            break;
        case 2:
            system("cls"); //clear screen after choice was selected
            login(); //run login function
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("Your choice was wrong please try again!!\n");
            previouslogin(); //run previouslogin function
            break;
    }
}

```

```

void admin_log() { //login function for admin
    char adminusername[100] = "admin";
    char adminpassword[100] = "cpe";
    char inputuser[100];
    char inputpassword[100];
    printf("-----|\n");
    printf("|          Login as administrator          |\n");
    printf("-----|\n");
    printf("\nEnter username: ");
    scanf("%s", &inputuser); //input admin's username
    printf("\nEnter password: ");
    scanf("%s", &inputpassword); //input admin's password
    if((strcmp(adminusername, inputuser)) == 0 &&
       (strcmp(adminpassword, inputpassword)) == 0) { //if both username and
       password are same with input get to mainmenuselection function
        system("cls"); //clear screen after choice was selected
        mainmenuselection(); //run mainmenuselection function
    }else {
        printf("\nInvalid username or password\n");
        previouslogin(); //run previouslogin function
    }
}

```

```
● ● ●

int login() { //login function for client and admin
    printf("-----|\n");
    printf("      LOGIN |\n");
    printf("-----|\n");
    printf("      Press 1 to login as client |\n");
    printf("      Press 2 to login as administrator |\n");
    printf("      Press 0 to exit the program |\n");
    printf("-----|\n");
    printf("\nChoice --> ");
    int choice;
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            clientmainmenuselection();
            break;
        case 2:
            system("cls"); //clear screen after choice was selected
            admin_log();
            break;
        case 0:
            exit(1); //exit exit program
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("      Your choice was wrong please try
again!!\n\n");
            login(); //run login function
            break;
    }
}
```

```
int mainmenuselection() { //mainmenuselection for admin
    printf("-----\n");
    printf("          MAIN MENU\n");
    printf("-----\n");
    printf("          Press 1 to enter inventory system\n");
    printf("          Press 2 to enter accounting system\n");
    printf("          Press 3 to enter daily summary\n");
    printf("          Press 4 to enter weekly summary\n");
    printf("          Press 5 to back to Login menu\n");
    printf("          Press 0 to exit the program\n");
    printf("-----\n");
    printf("\nChoice --> ");
    int button;
    scanf ("%d", &button);
    switch(button) { //switch case use int button as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            inventorysystem(); //run inventorysystem function
            break;
        case 2:
            system("cls"); //clear screen after choice was selected
            accountingsystem(); //run accountingsystem function
            break;
        case 3:
            system("cls"); //clear screen after choice was selected
            admindaily(); //run admindaily function
            previousdailyadmin();
            break;
        case 4:
            system("cls"); //clear screen after choice was selected
            adminweekly(); //run adminweekly function
            previousweeklyadmin();
            break;
        case 5:
            system("cls"); //clear screen after choice was selected
            login(); //run login function
            break;
        case 0:
            exit(1); //exit program
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("      Your choice was wrong please try\n");
            again!\\n\\n\\n");
            mainmenuselection(); //run mainmenuselection function
            break;
    }
}
```

```
● ● ●

int clientmainmenuselection() { //mainmenuselection for client
    printf("|-----|\n");
    printf("|          MAIN MENU          |\n");
    printf("|\n");
    printf("|      Press 1 to enter POS system |\n");
    printf("|      Press 2 to enter login menu |\n");
    printf("|      Press 0 to exit the program |\n");
    printf("|\n");
    printf("\nChoice --> ");
    int button;
    scanf("%d", &button);
    switch(button) { //switch case use int button as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            possystem(); //run possystem function
            break;
        case 2:
            system("cls"); //clear screen after choice was selected
            login(); //run login function
            break;
        case 0:
            system("cls"); //clear screen after choice was selected
            exit(1);
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("      Your choice was wrong please try\nagain!!\n\n");
            clientmainmenuselection(); //run clientmainmenuselection
    }
}
```

```
int writeposFile() {
    int i;
    p = fopen("pos.txt", "w"); //let p = openfile pos.txt for write
    if(p == NULL) {
        return -1;//p==NULL(file haven't created yet), return -1
    }
    fprintf(p, "%d\n", countpos);//write countpos value
    for(i = 0; i < countpos; ++i) {
        fprintf(p, "%d\n", buy[i].id); //write buy[i].id value
        fputs(buy[i].name, p); //write buy[i].name string name
        fprintf(p, "\n"); //write to enter new line
        fprintf(p, "%d\n", buy[i].quantity); //write buy[i].quantity
        value
        fprintf(p, "%f\n", buy[i].price); //write buy[i].price value
        fprintf(p, "%f\n", buy[i].principle); //write
        buy[i].principle value
        fputs(buy[i].timestp, p); //write buy[i].timestp string name
        fprintf(p, "\n"); //write to enter new line
        fprintf(p, "%d\n", buy[i].time); //write buy[i].time value
    }
    fclose(p);
    return 0;
}
```

```

int readposFile() {
    int n = 0;
    int i;
    p = fopen("pos.txt", "r");//let p = openfile pos.txt for read
    if(p == NULL) {
        return -1;//p==NULL(file haven't created yet), return -1
    }
    fscanf(p, "%d\n", &n);//read countpos value
    for(i = 0; i < n; ++i) {// read all the details from all the
function to the text file.
        fscanf(p, "%d\n", &buy[i].id);
        fgets(buy[i].name, 20,p);
        buy[i].name[strlen(buy[i].name) - 1] = 0; //remove new line
        fscanf(p, "%d\n", &buy[i].quantity);
        fscanf(p, "%f\n", &buy[i].price);
        fscanf(p, "%f\n", &buy[i].principle);
        fgets(buy[i].timestp, 30,p);
        buy[i].timestp[strlen(buy[i].timestp) - 1] = 0;//remove new
line
        fscanf(p, "%d\n", &buy[i].time);
    }
    fclose(p);
    return n;
}

```

```

void previousPOS() {
    int choice;
    printf("Press 0 to back to POS system: ");
    scanf("%d", &choice);
    switch(choice) {//switch case use int choice as case switcher
        case 0:
            system("cls");//clear screen after choice was selected
            possystem();
            break;
        default:
            system("cls");
            viewproduct();
            break;
    }
}

```

```
● ● ●

void previousdailyadmin() {
    int choice;
    printf("\nPress 1 to back to mainmenu / Press 0 to exit the
program : ");
    scanf("%d", &choice);
    switch(choice) {//switch case use int choice as case switcher
        case 1:
            system("cls");//clear screen after choice was selected
            mainmenuselection();
            break;
        case 0:
            exit(1);//exit program
            break;
        default:
            printf("\nYour choice was wrong please try again!!\n");
            previousdailyadmin();
            break;
    }
}
```

```
● ● ●

void previousweeklyadmin() {
    int choice;
    printf("\nPress 1 to back to mainmenu / Press 0 to exit the
program : ");
    scanf("%d", &choice);
    switch(choice) {//switch case use int choice as case switcher
        case 1:
            system("cls");//clear screen after choice was selected
            mainmenuselection();
            break;
        case 0:
            exit(1);//exit program
            break;
        default:
            printf("\nYour choice was wrong please try again!!\n");
            previousweeklyadmin();
            break;
    }
}
```

```
● ● ●

void previousdaily() {
    int choice;
    printf("\nPress 1 to back to POS menu / Press 0 to exit the
program : ");
    scanf("%d", &choice);
    switch(choice) {//switch case use int choice as case switcher
        case 1:
            system("cls");//clear screen after choice was selected
            possystem();
            break;
        case 0:
            exit(1);//exit program
            break;
        default:
            printf("\nYour choice was wrong please try again!!\n");
            previousdaily();
            break;
    }
}
```

```
● ● ●

void previousweekly() {
    int choice;
    printf("\nPress 1 to back to POS menu / Press 0 to exit the
program : ");
    scanf("%d", &choice);
    switch(choice) {//switch case use int choice as case switcher
        case 1:
            system("cls");//clear screen after choice was selected
            possystem();
            break;
        case 0:
            exit(1);//exit program
            break;
        default:
            printf("\nYour choice was wrong please try again!!\n");
            previousweekly();
            break;
    }
}
```

```

void IDPurchaseCorrection() {
    int choice;
    printf("Press 1 to try again / Press 2 to back to POS menu: ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            purchaseproduct();
            break;
        case 2:
            system("cls");
            possystem();
            break;
        default:
            printf("\nYour choice was wrong please try
again!!\n\n");
            IDPurchaseCorrection();
            break;
    }
}

```

```

void successfulpurchase() {
    int choice;
    printf("Press 1 to purchase another product / Press 2 to back to
POS menu: ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            purchaseproduct();
            break;
        case 2:
            system("cls");
            possystem();
            break;
        default:
            printf("\nYour choice was wrong please try
again!!\n\n");
            successfulpurchase();
            break;
    }
}

```

```
void insufficientQuant() {
    int choice;
    printf("Press 1 to try again / Press 2 to back to POS menu: ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            purchaseproduct();
            break;
        case 2:
            system("cls");
            possystem();
            break;
        default:
            printf("\nYour choice was wrong please try
again!!\n\n");
            insufficientQuant();
            break;
    }
}
```

```
void wrongminimumQuant() {
    int choice;
    printf("Press 1 to try again / Press 2 to back to POS menu: ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            purchaseproduct();
            break;
        case 2:
            system("cls");
            possystem();
            break;
        default:
            printf("\nYour choice was wrong please try
again!!\n\n");
            wrongminimumQuant();
            break;
    }
}
```

```

● ● ●

void viewproduct() {
    int i;
    count = readFile(); //let count = number of dataset in readFile()
    if(count < 0) {
        puts("cannot open file"); //if dataset is below 0 show cannot open file
    }
    printf("-----|\\n");
    printf("----- INVENTORY |\\n");
    printf("-----|\\n");
    printf("----- NAME | PROD ID | QUANTITY | MINNIMUM QUANTITY | PRICE |\\n");
    printf("-----|\\n");
    for(i = 0; i < count; i++) { //print all product data but no principle
        printf("-----| %-13s %-11d %-10d %-8d %-8.2f |\\n", product[i].name,
product[i].id, product[i].quantity, product[i].min_quan, product[i].price);
    }
    printf("-----|\\n\\n\\n")
}

```

```

● ● ●

void viewproductforpurchase() {
    int choice;
    printf("Are you sure to buy the product ?\\n");
    printf("Yes I'm sure press 1 / No and back to POS system press 0\\n");
    printf("Choice --> ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
    case 1:
        system("cls"); //clear screen after choice was selected
        purchaseproduct();
        break;
    case 0:
        system("cls");
        posssystem();
        break;
    default:
        system("cls");
        posssystem();
        break;
    }
}

```

```

● ● ●

int cartviewproductforpurchase(int id) {
    system("cls");//clear screen after choice was selected
    int i, j;
    count = readfile();//let count = number of dataset in readfile()
    for(i = 0; i < count; i++) {//check the id that user input
        if(id == product[i].id) {//if id input is equal to any product id let j =i and break
            j = i;
            break;
        }else {
            continue;//if id input is not equal to any product id let j =i and continue
        }
    }
    if(count < 0) {
        puts("cannot open file");//if dataset is below 0 show cannot open file
    }
    printf("-----|\\n");
    printf(" | INVENTORY |\\n");
    printf(" |-----|\\n");
    printf(" | NAME | PROD ID | QUANTITY | MINNIMUM QUANTITY | PRICE |\\n");
    printf(" |-----|\\n");
    printf(" | %-13s %-11d %-10d %-8d %-8.2f |\\n", product[j].name,
    product[j].id, product[j].quantity, product[j].min_quan, product[i].price);
    printf(" |-----|\\n")
}

```

```

● ● ●

void daily() {
    int i = 0, flag = 0, j, choice;
    float total, totalarry[1000];
    countpos = readposfile();//let countpos = number of dataset in readposfile()
    int temp = countpos;//let temp = number of dataset
    if(countpos < 0) {
        puts("Cannot open file");//if dataset is below 0 show cannot open file
    }
    printf("-----|\\n");
    printf(" | DAY SUMMARY (COUNT BY FIRST PRODUCT) |\\n");
    while(i >= 0) {//while i(day) >= 0 do condition below
        total = 0;//let total = 0 everytime enter loop
        printf(" |-----|\\n");
        printf(" | DAY | NAME | PROD ID | QUANTITY | PRICE | TIME |\\n");
        printf(" |-----|\\n");
        for(j = 0; j < countpos; j++) {
            if((buy[j].time - buy[0].time) >= i * 86400 && (buy[j].time - buy[0].time) < (i + 1) * 86400) {//use buy[0].time
                as starter of timer, and if it satisfy the equation printout each data of days
                total = total + (buy[j].price * buy[j].quantity);//store each bought item price* item quantity in total
                itself
                printf(" | %-4d %-11s %-11d %-5d %-9.2f %-12s |\\n", i + 1, buy[j].name, buy[j].id,
                buy[j].quantity, buy[j].price * buy[j].quantity, buy[j].timestamp);//print out each data in each days
                temp--;//decrease number of data set that store in temp everytime enter if
                flag++;//increase number of data set that store in temp everytime enter if
            }
        }
        totalarry[i] = total;//store total price in each day in total array
        printf(" |-----|\\n");
        printf(" | TOTAL OF DAY %d = %-8.2f |\\n", i + 1, total);
    }
    totalarry[i];//print day index and total
    printf(" |-----|\\n");
    if(flag == 0) {//if it didnt enter if which temp didnt plus 1 and print data not found
        printf(" | Data not found!! |\\n\\n");
    }
    if(temp == 0) {//if temp decrease until it = 0 break out of while
        break;
    }
    i++;//add day index every time when it read here
}

```



```

void purchaseproduct() {
    viewproduct();
    time_t t = time(NULL); //get time in sec to time_t t
    char *str_time = ctime(&t); //get string time to str_time
    int quant, i, id, choice, z = false, temp1 = false, temp2 = false, temp3 = false, c = false;
    count = readfile(); //count = number of dataset in readfile()
    countpos = readposFile(); //countpos = number of dataset in readposFile()
    printf("Choose product's ID\n");
    printf("Product ID: ");
    fflush(stdin);
    scanf("%d", &id);
    printf("Are you sure to buy this product?\nSure press 1. \nNot Sure press 0 and choose product.\nChoice --> ");
    scanf("%d", &choice);
    switch (choice){ //switch case use int choice as case switcher
        case 1:
            break;
        case 0:
            purchaseproduct();
            break;
        default:
            purchaseproduct();
            break;
    }
    for(i = 0; i < count; i++) {
        if(id == product[i].id) {
            z = true;
            printf("\nItem available!\n");
            cartviewproductforpurchase(id);
            printf("Enter the quantity you want to buy: ");
            fflush(stdin);
            scanf("%d", &quant);
            if(quant <= product[i].quantity && quant >= product[i].min_quan) { //if quantity that is entered is right for min
                quantity do below
                printf("\nSuccessful purchase!\n");
                temp1 = true;
            }
            else if(quant > product[i].quantity) { //if quantity user enter greater than actual quantity do below
                printf("\nInsufficient quantity please restock!\n");
                temp2 = true;
                break;
            }
            else if(quant < product[i].min_quan) { //if quantity user enter less than than actual quantity do below
                printf("\nSorry, but there are minimum quantities for each item!\n");
                temp3 = true;
                break;
            }
            product[i].quantity -= quant; //decrease actual quantity by quantity user input
            if(countpos > 0) {
                countpos = readposFile(); //if there is data, store number of set of data in countpos
            }
            buy[countpos].id = product[i].id; //store each data user choose from product to buy
            strcpy(buy[countpos].name, product[i].name); //get string name
            buy[countpos].quantity = quant; //store buy[countpos].quantity = int quant
            buy[countpos].price = product[i].price;
            buy[countpos].principle = product[i].principle;
            strcpy(buy[countpos].timestep, str_time); //get string time stamp
            buy[countpos].time = t; //get current time in int
            ++countpos; //add 1 everytime
        }
    }
    if(z == false) { //if it didnt get into any if
        printf('Cant find the product id: %d\n\n', id);
        IDPurchaseCorrection();
    }
    writeFile();
    writeposFile();
    if(temp1 == true) {
        successfulpurchase();
    }
    else if(temp2 == true) {
        insufficientQuant();
    }
    else if(temp3 == true) {
        wrongminimumQuant();
    }
    previousPOS();
}

```

```

int possystem() { //possystem function
    printf("-----\n");
    printf("          Point of sale\n");
    printf("-----\n");
    printf("          Press 1 to view product\n");
    printf("          Press 2 to purchase product\n");
    printf("          Press 3 to view daily summary\n");
    printf("          Press 4 to view weekly summary\n");
    printf("          Press 5 to back to main menu\n");
    printf("          Press 0 to exit the program\n");
    printf("-----\n");
    printf("\nChoice --> ");
    int choice;
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            viewproduct(); //run viewproduct function
            previousPOS(); //run previousPOS function
            break;
        case 2:
            system("cls"); //clear screen after choice was selected
            purchaseproduct(); //run purchase function
            break;
        case 3:
            system("cls"); //clear screen after choice was selected
            daily(); //run daily function
            previousdaily(); //run previousdaily function
            break;
        case 4:
            system("cls"); //clear screen after choice was selected
            weekly(); //run weekly function
            previousweekly(); //run previousweekly function
            break;
        case 5:
            system("cls"); //clear screen after choice was selected
            clientmainmenuselection(); //run clientmainmenuselection
function
            break;
        case 0:
            exit(1); //exit program
            break;
        default:
            system("cls"); //clear screen after choice was selected
            printf("Your choice was wrong please try again!!\n");
            possystem(); //run possystem function
            break;
    }
}

```

```
void previousacc() {
    int choice;
    printf("Press 1 to back to accounting menu / Press 0 to exit the
program : ");
    scanf("%d", &choice);
    switch(choice) { //switch case use int choice as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            accountingsystem(); //run accountingsystem function
            break;
        case 0:
            system("cls");
            exit(1); //exit program
            break;
        default:
            printf("Your choice was wrong please try again!!\n\n");
            previousacc(); //running their own function
            break;
    }
}
```

```

int accountingsystem() {
    printf("-----|\n");
    printf("|          Accounting          |\n");
    printf("-----|\n");
    printf("|      Press 1 to display purchased lists |\n");
    printf("|      Press 2 to back to main menu       |\n");
    printf("|      Press 0 to exit the program        |\n");
    printf("-----|\n");
    printf("\nChoice --> ");
    int button;
    scanf("%d", &button);
    switch(button) { //switch case use int button as case switcher
        case 1:
            system("cls"); //clear screen after choice was selected
            purchasedlists();
            break;
        case 2:
            system("cls");
            mainmenuselection();
            break;
        case 0:
            system("cls");
            exit(1); //exit program
            break;
        default:
            system("cls");
            printf("Your choice was wrong please try again!!\n");
            accountingsystem();
            break;
    }
}

```

```

void previous() { //For press back to main menu
    int choice;
    printf("Press 0 to back to inventory --> ");
    scanf("%d", &choice);
    switch(choice){
        case(0):
            system("cls");
            inventorysystem();
            break;
    }
}

```

```
int writeFile() { // open file .txt and write on it
    int i;
    f = fopen("inventory.txt", "w");
    if(f == NULL) {      //if express Null = file dosen't create
        return - 1;
    }
    fprintf(f, "%d\n", count);
    for(i = 0; i < count; ++i) {
        fprintf(f, "%d\n", product[i].id);
        fputs(product[i].name, f); // print but string
        fprintf(f, "\n");
        fprintf(f, "%d\n", product[i].quantity);
        fprintf(f, "%f\n", product[i].price);
        fprintf(f, "%f\n",
product[i].principle);
        fprintf(f, "%d\n", product[i].min_quan);
    }
    fclose(f);
    return 0;
}
```

```
int readFile() { //open file and received value
    int n = 0;
    int i;
    f = fopen("inventory.txt", "r");
    if(f == NULL) {
        return -1; //if doesn't have retrun -1
    }
    fscanf(f, "%d\n", &n); // received index of struct in file
    for(i = 0; i < n; ++i) {
        fscanf(f, "%d\n", &product[i].id);
        fgets(product[i].name, 20, f);
        product[i].name[strlen(product[i].name) - 1] = 0;
        fscanf(f, "%d\n", &product[i].quantity);
        fscanf(f, "%f\n", &product[i].price);
        fscanf(f, "%f\n", &product[i].principle);
        fscanf(f, "%d\n", &product[i].min_quan);
    }
    fclose(f);
    return n; //return index of struct to count
}
```

```
int IDchecker(int i,int j){
    int temp, temp2;
    count = readFile(); //let count = index of struct in file inventory.txt
    printf("Product ID: ");
    scanf("%d", &temp);
    for(i = 0; i < count; i++){
        if(product[i].id == temp) { //if input id = product id print number is taken
            printf("Product ID number is already taken!!\n");
            printf("Press 0 to try again or Press 1 to back to inventorymenu --> ");
            scanf("%d", &temp2);
            if(temp2 == 0) {
                system("cls"); //clear screen
                displayaddproduct() ;
            }else {
                system("cls");
                inventoriesystem();
            }
        }
    }
}
```

```
void addproduct() { // this function is add new product in to inventory.txt list
    printf("ENTER NEW PRODUCTS\n");
    readFile();
    if(count > 0) {
        count = readFile(); //let count = index of struct in file inventory.txt
        IDchecker(0,count); //sent index of struct in file to check that id already have yet
    }else {
        printf("\nProduct ID Number: ");
        fflush(stdin);
        scanf("%d",&product[count].id);//input value to last index in product id
    }
    printf("Product name: ");scanf("%s",product[count].name);
    printf("Quantity of the product: ");scanf("%d", &product[count].quantity);
    printf("Price of the product: ");scanf("%f", &product[count].price);
    printf("Principle of the product: ");scanf("%f", &product[count].principle);
    printf("Minimum Order Quantity of the product: ");scanf("%d", &product[count].min_quan);
    ++count;
    system("cls");
    printf("Added successfully!!\n");
    writeFile();
}
```

```
● ● ●  
int modifyproduct() {  
    int id;  
    int test;  
    int i;  
    int new_number;  
    int choice,choice2,choice3;  
    printf("EDIT A PRODUCT!");  
    printf("\nEnter the id of the product that you want to edit: ");  
    fflush(stdin);  
    scanf("%d",&id);  
    test = checkID(id);//let test store value return from checkID(id)  
// printf("\ntest = %d\n",test);  
    if(test == 0) {//if test = 0 system clear then say id not found  
        system("cls");//clear screen  
        printf("The id num %d is not found.\n", id);  
        printf("Press 0 return to inventorymenu or press 1 to try again --> ");  
        scanf("%d",&choice2);  
        switch (choice2)  
        {//switch case use int choice2 as case switcher  
            case 0:  
                system("cls");  
                inventorysystem();  
                break;  
            case 1:  
                system("cls");  
                inventoryformodifyprod();  
            default:  
                inventorysystem();  
                break;  
        }  
    }else {  
        readFile();//read file  
        {  
            for(i = 0; i < count; i++){  
                if(id != product[i].id){//if id not equal to product id writefile  
                    writeFile();  
                }else {  
                    system("cls");//clear screen
```

```
showInveninmodify(id);
printf("\nPress 1 to update product's ID number");
printf("\nPress 2 to update product's name");
printf("\nPress 3 to update product's quantity");
printf("\nPress 4 to update product's price");
printf("\nPress 5 to update product's principle");
printf("\nPress 6 to update product's minimum quantity");
printf("\nEnter 0 to back to inventory menu\n");
printf("\nEnter your choice --> ");
fflush(stdin);
scanf("%d", &choice);
switch(choice) //switch case use int choice as case switcher
case 1:
    system("cls");
    showInveninmodify(id);
    printf("Enter new ID: ");
    fflush(stdin);
    scanf("%d", &new_number);
    product[i].id = new_number;
    id = new_number;
break;
case 2:
    system("cls");//clear screen
    showInveninmodify(id);
    printf("Enter new name: ");
    fflush(stdin);
    gets(product[i].name);
break;
case 3:
    system("cls");
    showInveninmodify(id);
    printf("Enter new quantity: ");
    scanf("%d", &product[i].quantity);
break;
case 4:
    system("cls");
    showInveninmodify(id);
    printf("Enter new price: ");
    scanf("%f", &product[i].price);
break;
case 5:
    system("cls");
    showInveninmodify(id);
    printf("Enter new principle: ");
    scanf("%f", &product[i].principle);
break;
case 6:
    system("cls");
    showInveninmodify(id);
    printf("Enter the new minimum order quantity: ");
    scanf("%d", &product[i].min_quan);
```



```

● ● ●

void deleteproduct() { //For delete product in inventory.txt. file
    count = readFile(); // let count = index of struct in inventory.txt file
    int id;
    int i, j, choice;
    int z = false;
    printf("Enter the product ID that you want to delete --> ");
    fflush(stdin);
    scanf("%d", &id);
    for(i = 0; i < count; i++) { //loop for find same id that already have in inventory.txt file
        if(product[i].id == id){
            z = true;
            for(j = i; j < (count - 1); j++) { //set new index of struct in inventory.txt file that old index value more than index
that we deleted
                product[j] = product[j + 1];
            }
            count--; //Decrease count by 1 value because we delete 1 product
            printf("Delete successfully\n");
        }
    }
    if(z == false) {
        printf("Cant find product id: %d.\n", id);
        printf("Press 1 to try again / Press 0 to go back to inventory page --> ");
        scanf("%d", &choice);
        switch(choice){//switch case use int choice as case switcher
            case 1:
                system("cls");//clear screen
                showInventoryForDel();
                break;
            case 0:
                system("cls");
                inventorySystem();
                break;
            default:
                system("cls");
                inventorySystem();
                break;
        }
    }
    writeFile(); //write file
}

```

```

● ● ●

void displayproduct() { //Show product details
    int i;
    count = readFile(); //let count = number of dataset in readFile()
    if(count < 0) {
        puts("cannot open file"); //if dataset is below 0 show cannot open file
    }
    printf("|-----|-----|-----|-----|-----|-----|-----|\n");
    printf("|-----|-----|-----|-----|-----|-----|-----|\n");
    printf("|-----|-----|-----|-----|-----|-----|-----|\n");
    printf "|-----|-----|-----|-----|-----|-----|-----|\n";
    for(i = 0; i < count; i++) {
        printf "| %11s      %12d      %12d      %10d      %13.2f      %11.2f | \n", product[i].name, product[i].id,
product[i].quantity, product[i].min_quan, product[i].price, product[i].principle); //print out each data
    }
    printf "|-----|-----|-----|-----|-----|-----|-----|\n";
    previous();
}

```

```

int inventoriesystem() {
    int choice;
    count = readfile(); //let count = number of dataset in readfile()
    if(count < 0)
        printf("Cannot locate file\n");
    do {
        printf("-----\n");
        printf("          INVENTORY PROGRAM\n");
        printf("-----\n");
        printf("          Press 1 to add product\n");
        printf("          Press 2 to modify product\n");
        printf("          Press 3 to delete product\n");
        printf("          Press 4 to display all exiting product\n");
        printf("          Press 5 to back to main menu\n");
        printf("          Press 0 to exit the program\n");
        printf("-----\n");
        printf("\nChoice --> ");
        scanf("%d", &choice);
        switch(choice){ //switch case use int choice as case switcher
            case 1:
                system("cls"); //clear screen after choice was selected
                displayaddproduct(); //run displayaddproduct function
                break;
            case 2:
                system("cls"); //clear screen after choice was selected
                inventoryformodifyprod(); //run inventoryformodifyprod function
                break;
            case 3:
                system("cls"); //clear screen after choice was selected
                showinventoryfordelfun(); //run showinventoryfordelfun function
                break;
            case 4:
                system("cls"); //clear screen after choice was selected
                displayproduct(); //run displayproduct function
                break;
            case 5:
                system("cls"); //clear screen after choice was selected
                mainmenuselection(); //run mainmenuselection
                break;
            case 0:
                system("cls"); //clear screen after choice was selected
                exit(1); //exit program
                break;
            default:
                printf("Your choice was wrong please try again");
                break;
        }
    }while(choice != 0);
}

```

Section 3

Cepture Program

Main

```
CPE100 Module 3 : Online Shop Management System
-----
Group Members
-----
65070503449 Thanaphat Ngoennet
65070503450 Thanakorn Soonjaw
65070503456 Parunchai Kochseni
65070503459 Phakalpol Maneesopa
65070503475 Kanchai Lerdsrisakulrat
-----
Press 1 to enter the program
Press 0 to exit the program
-----
Choice -->
```

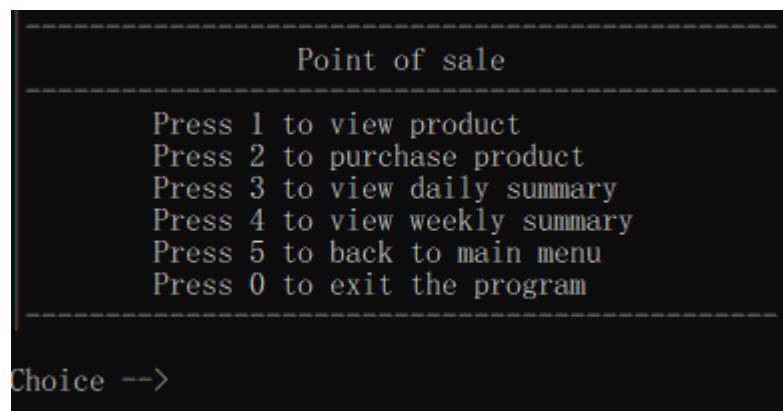
Press 1

```
LOGIN
-----
Press 1 to login as client
Press 2 to login as administrator
Press 0 to exit the program
-----
Choice --> -
```

Login as Client (Main Menu)

```
MAIN MENU
-----
Press 1 to enter POS system
Press 2 to enter login menu
Press 0 to exit the program
-----
Choice -->
```

POS system



view product (Client Functions)

INVENTORY					
NAME	PROD ID	QUANTITY	MINNIMUM QUANTITY	PRICE	
aa	1	1	1	20.00	
dd	4	51	1	3.00	
bb	2	24	2	15.00	
cc	3	17	1	56.00	

Press 0 to back to POS system:

purchase product (Client Functions)

INVENTORY					
NAME	PROD ID	QUANTITY	MINNIMUM QUANTITY	PRICE	
aa	1	1	1	20.00	
dd	4	51	1	3.00	
bb	2	24	2	15.00	
cc	3	17	1	56.00	

Choose product's ID
 Product ID:

INVENTORY					
NAME	PROD ID	QUANTITY	MINNIMUM QUANTITY	PRICE	
cc	3	17	1	56.00	

Enter the quantity you want to buy: 2

Successful purchase!
 Press 1 to purchase another product / Press 2 to back to POS menu: -

view daily summary (Client Functions)

TOTAL OF DAY 7 = 0.00								
DAY	NAME	PROD ID	QUANTITY	PRICE	TIME			
8	dd	4	1	3.00	Thu Dec 22 15:24:34 2022			
8	dd	4	2	6.00	Thu Dec 22 15:27:31 2022			
TOTAL OF DAY 8 = 9.00								
TOTAL OF DAY 9 = 23.00								
DAY	NAME	PROD ID	QUANTITY	PRICE	TIME			
9	aa	1	1	20.00	Fri Dec 23 15:29:55 2022			
9	dd	4	1	3.00	Fri Dec 23 15:30:07 2022			
TOTAL OF DAY 10 = 23.00								

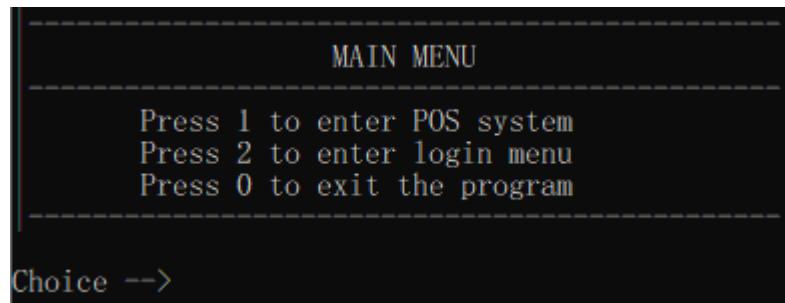
Press 1 to back to POS menu / Press 0 to exit the program : -

view weekly summary (Client Functions)

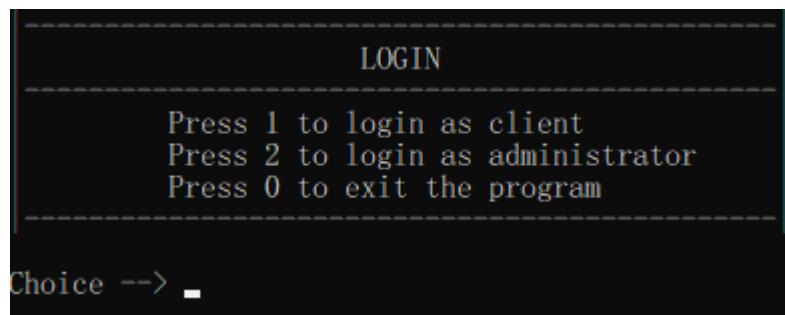
WEEK SUMMARY (COUNT BY FIRST PRODUCT)						
Week	NAME	PROD ID	QUANTITY	PRICE	TIME	
1	aa	1	1	20.00	Thu Dec 15 11:50:20 2022	
1	dd	4	1	3.00	Thu Dec 15 11:50:28 2022	
1	aa	1	1	20.00	Thu Dec 15 12:53:08 2022	
1	aa	1	1	20.00	Thu Dec 15 12:53:46 2022	
1	bb	2	3	45.00	Fri Dec 16 15:33:32 2022	
1	cc	3	1	56.00	Fri Dec 16 15:33:43 2022	
1	cc	3	1	56.00	Sat Dec 17 15:34:54 2022	
1	dd	4	1	3.00	Sun Dec 18 15:35:53 2022	
1	aa	1	1	20.00	Sun Dec 18 15:36:11 2022	
1	bb	2	3	45.00	Sun Dec 18 15:36:21 2022	
1	cc	3	2	112.00	Thu Dec 15 23:05:18 2022	
TOTAL OF WEEK 1 = 400.00						
Week	NAME	PROD ID	QUANTITY	PRICE	TIME	
2	dd	4	1	3.00	Thu Dec 22 15:24:34 2022	
2	dd	4	2	6.00	Thu Dec 22 15:27:31 2022	
2	aa	1	1	20.00	Fri Dec 23 15:29:55 2022	
2	dd	4	1	3.00	Fri Dec 23 15:30:07 2022	
2	dd	4	1	3.00	Sat Dec 24 15:30:36 2022	
2	aa	1	1	20.00	Sat Dec 24 15:30:48 2022	
TOTAL OF WEEK 2 = 55.00						

Press 1 to back to POS menu / Press 0 to exit the program : -

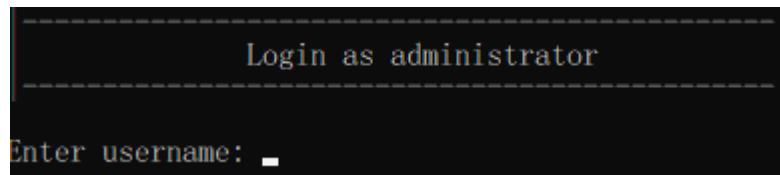
Press 5 (Back to main menu)



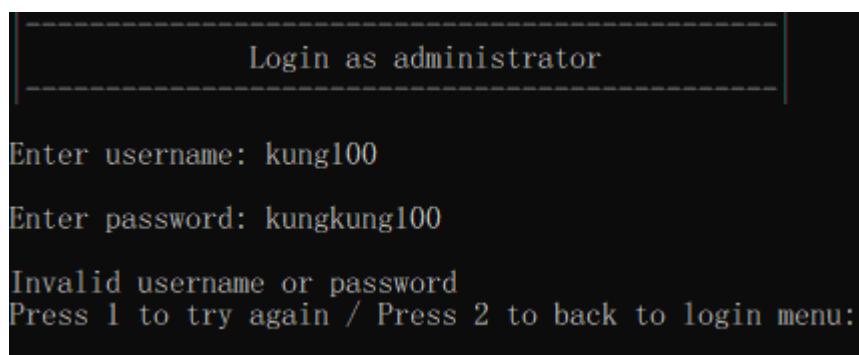
Press 2 (enter login menu)



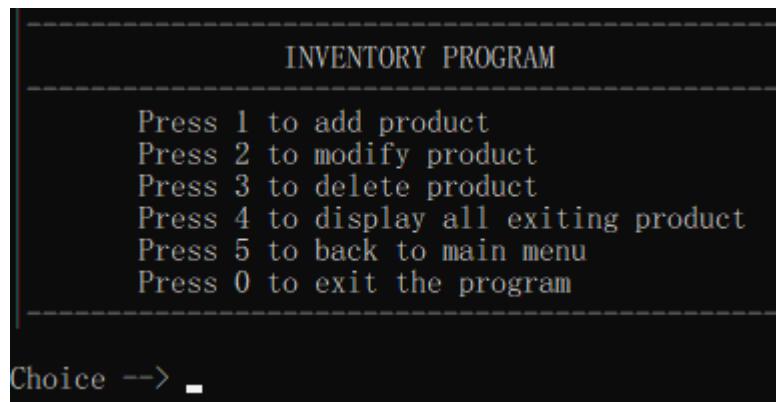
Login as administrator



If the username & password is wrong



If the username & password is true



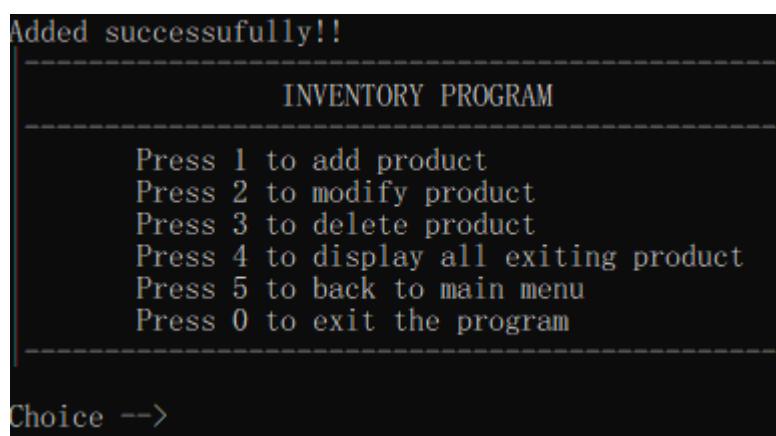
Add product (Administrator Functions)

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	15	1	56.00	25.00	

ENTER NEW PRODUCTS
Product ID:

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	

ENTER NEW PRODUCTS
Product ID: ee
Product name: Quantity of the product: 5
Price of the product: 65
Principle of the product: 40
Minimum Order Quantity of the product: 3.



Modify product (Administrator Functions)

INVENTORY						
NAME	PROD_ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	
ee	0	5	3	65.00	40.00	

EDIT A PRODUCT!

Enter the id of the product that you want to edit:

INVENTORY						
NAME	PROD_ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	
ee	0	5	3	65.00	40.00	

EDIT A PRODUCT!

Enter the id of the product that you want to edit: 0

INVENTORY						
NAME	PROD_ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
ee	0	5	3	65.00	40.00	

Press 1 to update product ID number

Press 2 to update product's name

Press 3 to update product's quantity

Press 4 to update product's price

Press 5 to update product's principle

Press 6 to Update product's minimum quantity

Press 0 to Back to inventory menu

Enter your choice -->

INVENTORY						
NAME	PROD_ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
ee	0	5	3	65.00	40.00	

Enter new ID: -

INVENTORY						
NAME	PROD_ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
ee	0	5	3	65.00	40.00	

Enter new ID: 5-

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
ee	5	5	3	65.00	40.00	

Product was updated!!
Press 1 to edit product again or press 0 to return to inventorymenu --> -

Delete Product (Administrator)

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	
ee	5	5	3	65.00	40.00	

Enter the product ID that you want to delete --> -

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	
ee	5	5	3	65.00	40.00	

Enter the product ID that you want to delete --> 5-

INVENTORY						
NAME	PROD ID	QUANTITY	MINIMUM QUANTITIES	PRICE	PRINCIPLE	
aa	1	1	1	20.00	5.00	
dd	4	51	1	3.00	1.00	
bb	2	24	2	15.00	3.00	
cc	3	17	1	56.00	25.00	

Press 0 to back to inventory --> -

Section 4

Work Load

Thanapat Ngoennet ID:65070503449

1. Inventory (In basically)
2. Accounting
3. POS
4. Debug & Improve

Thanakorn Soonjaw ID:65070503450

1. Inventory (In basically)
2. Inventory (In advance)
3. Debug & Improve

Parunchai Kochseni ID:65070503456

1. Inventory (In basically)
2. Accounting
3. POS
4. Debug & Improve

Phakalpol Maneesopa ID:65070503459

1. Inventory (In basically)
2. Accounting
3. POS
4. Debug & Improve

Kanchai Lerdsrisakulrat ID:65070503475

1. Inventory (In basically)
2. Inventory (In advance)
3. Debug & Impove

First, we start designing basic inventory functions together by using a collaboration tool in visual studio code called Live Share. After that we divide up the work by letting 2 people continue doing advance inventory function and 3 people holding an accounting system and POS. Lastly, we allow that someone who has any ideas or suggestions can bring it into files that other members are responsible for.