CUSTOMER BILLING SYSTEM

**Group Members :** Saichaitanya Kathiri (U101115FCS137)

Mandala Sai Kushal (U101115FCS114)

A Manovishnu (U101115FEC012)

**Description :**

Customer Billing System is an application designed to generate the bill to the customer which can be used in any departmental store, shops, cafes etc. You can use this application to keep the records such as name, address, mobile number, paid amount, due amount, payment data etc. of your regular customer. Moreover, if you have a new customer, you can add and edit the account at any time.

**Tools to be used :**

1. GCC Compiler

2. Gedit

3.GNU/GDB debugger

4.C/C++ interpreter Ch 5.0

**Timeline :**

Scheduled to be completed by the end of November.

**Role of individuals in the Group:**

Saichaitanya Kathiri : Code for Structure and pricing data.

Mandala Sai Sushal : Code for Data to be added to the bill.

A Mano Vishnu : Code for Accounting part and debugging.

**About Tools we use :**

**1. GCC Compiler**

The GCC Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages.GCC is a key component of the GNU toolchain and the standard compiler for most Unix-like Operating Systems.The standard way to compile this program is with the command g++ hello.C -o hello The exact set of GCC optimizations varies from release to release as it develops, but includes the standard algorithms, such as loop optimization, jump threading, common subexpression elimination,instruction scheduling, and so forth.The standard compiler releases since 4.6 include front ends for C (gcc), C++ (g++), Objective-C,Objective-C++, Fortran (gfortran), Java (gcj), Ada (GNAT), and Go (gccgo).

**2. Gedit**

gedit is a Linux text editor commonly deployed as part of the GNOME desktop environment. Most Linux guides and tutorials will get you to use the nano editor or vi to edit text files and configuration files and the reason for this is that nano and vi are almost guaranteed to be installed as part of the Linux operating system.While aiming at simplicity and ease of use, gedit is a powerful general purpose text editor. It can be used to create and edit all kinds of text files. gedit features a flexible plugin system which can be used to dynami‐cally add new advanced features to gedit itself.The gEdit user interface contains a single toolbar at the top with a panel for entering text below it.

**3.GNU/GDB debugger**

A debugger is a program thatruns other programs, allowing the user to exercise control over these programs, and to examine variables when problems arise.GNU Debugger, which is also called gdb, is the most popular debugger for UNIX systems to debug C and C++ programs.GDB allows you to run the program up to a certain point, then stop and print out the values of certain variables at that point, or step through the program one line at a time and print out the values of each variable after executing each line.GDB cannot be used for programs that compile with errors and it does not help in fixing those errors.

**4. C/C++ interpreter Ch 5.0**

The release of Ch and its toolkits for Linux PPC continues SoftIntegration&#39;s leadership in cross-platform scripting, numerical computing and embedded scripting. Ch Control System Toolkit, Ch Mechanism Toolkit, Ch CGI Toolkit and C++ Graphical Library are available in Linux PPC as well.Ch is easy and simple for embedded scripting. Embedded Ch allows users to embed Ch into C/C++ application programs and hardware as a scripting engine. It is ideal for open-architecture application integration. With Embedded Ch, C/C++ applications can be extended with all features of Ch.It allows execution of C/C++ scripts generated dynamically. Embedded Ch supports multi-threading and has a small footprint. The pointer and time deterministic nature of the C language provides a perfect interface with hardware in real-time systems.Ch supports an increasing number of third party software applications. It includes Ch Control toolkit,Ch Mechanism Toolkit, Ch NAG Statistics Package, Ch CGI toolkit, Ch ODBC toolkit, Ch XML packages,Intel OpenCV for computer vision and image processing, National Instruments&#39; data acquisition toolkit NI-DAQ and motion control toolkit NI-Motion, Barret Technology&#39;s Barret Hand robotic manipulator control, and IDE.

**Deliverables :**

1.You can hold any number of accounts and account can be added to the program at any time.

2.The programming of simple calculations such as calculation of due amount, balance etc. have been embed in the code of project.

3.The due amount to be paid is shown as negative balance.

4.If you have nothing to do with the program, you can directly exit from the main menu.

**Code :**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

struct itm

{

int prno;

char prname[10];

int qty;

int price;

}it;

struct customer

{

int prno;

char prname[10];

int qty;

int price;

int amount;

}cst[100];

struct tempor

{

int prno;

char prname[10];

int qty;

int price;

int amount;

}cstr[100];

int t=0,n=0,k=0,ba=0,ti=0;

///function section\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void create()

{

int i=0;

FILE \*fp,\*fpq;

fp=fopen("shop.txt","w");

if(fp==NULL)

{

printf("ERROR\n");

exit(0);

}

printf("enter the number of records\n");

scanf("%d",&n);

fpq=fopen("asd.txt","w");

fprintf(fpq,"%d",n);

fclose(fpq);

while(i<n)

{

printf("enter the product code\n");

scanf("%d",&it.prno);

printf("enter the name\n");

scanf("%s",it.prname);

printf("enter the quantity\n");

scanf("%d",&it.qty);

printf("enter the price\n");

scanf("%d",&it.price);

i++;

fprintf(fp,"%d\t%s\t%d\t%d\t",it.prno,it.prname,it.qty,it.price);

}

//fprintf(fp,"%d",0);

fclose(fp);

printf("\n records are being created so please come back again\n");

// exit(0);

}

/////function 2\*\*\*\*\*\*\*\*\*\*\*

void bill()

{

int na,a,cont=0,k,r=0,w=0,ch,l=0,bi=0,ln=0;

//char k;

l3 : printf("enter the product number\n");

scanf("%d",&na);

FILE \*fp,\*fptr;

fp=fopen("shop.txt","r");

fptr=fopen("temp.txt","w");

while(!(feof(fp)))

{

l++;

fscanf(fp,"%d",&it.prno);

fscanf(fp,"%s",it.prname);

fscanf(fp,"%d",&it.qty);

fscanf(fp,"%d",&it.price);

if(it.prno==na)

{

ln++;

if(ln==2)

break;

printf("enter the quantity\n");

scanf("%d",&a);

if(a<=it.qty)

{

cst[ti].prno=it.prno;

strcpy(cst[ti].prname,it.prname);

cst[ti].qty=a;

cst[ti].price=it.price;

it.qty=it.qty-a;

ti++;

//goto l1;

}

else

printf("out of stock\n");

bi++;

//goto l1;

}

else

{

cont++;

fprintf(fptr,"%d\t",it.prno);

fprintf(fptr,"%s\t",it.prname);

fprintf(fptr,"%d\t",it.qty);

fprintf(fptr,"%d\t",it.price);

continue;

}

fprintf(fptr,"%d\t",it.prno);

fprintf(fptr,"%s\t",it.prname);

fprintf(fptr,"%d\t",it.qty);

fprintf(fptr,"%d\t",it.price);

}

if(l==cont)

printf("error\nchoose proprly\n");

fclose(fptr);

fclose(fp);

remove("shop.txt");

rename("temp.txt","shop.txt");

ln=0;

printf("do you want to shop more\npress 1 for yes\n");

scanf("%d",&k);

if(k==1)

goto l3;

}

void display()

{

int nk=0;

FILE \*fp;

fp=fopen("shop.txt","r");

if(feof(fp))

{

printf("error in opening file\n");

exit(0);

}

printf("================================================================================\n");

printf("\t|product number\t|product name\t|quantity\t|price\t|\n");

printf("=================================================================================\n");

while(nk!=n)

{

nk++;

fscanf(fp,"%d",&it.prno);

if(it.prno==0)

{

printf("product number cannot be zero\n");

break;

}

fscanf(fp,"%s",it.prname);

fscanf(fp,"%d",&it.qty);

fscanf(fp,"%d",&it.price);

printf("\n\t %14d|\t%8s|\t%8d|\t%d|\n",it.prno,it.prname,it.qty,it.price);

}

fclose(fp);

}

////functin section\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void edit()

{

FILE \*fp,\*fptr,\*fpk;

int h,found=0,ka=0,z,p;

fp=fopen("shop.txt","r");

fptr=fopen("temp1.txt","w");

fpk=fopen("temp2.txt","w");

printf("enter the product number to delete\n");

scanf("%d",&h);

p=n;

while(!(ka==p))

{

ka++;

fscanf(fp,"%d",&it.prno);

if(it.prno==h)

{

found=1;

fscanf(fp,"%s",it.prname);

fscanf(fp,"%d",&it.qty);

fscanf(fp,"%d",&it.price);

n--;

printf("product has been sucessfully deleted\n");

}

else

{

fscanf(fp,"%s",it.prname);

fscanf(fp,"%d",&it.qty);

fscanf(fp,"%d",&it.price);

fprintf(fptr,"%d\t%s\t%d\t%d\t",it.prno,it.prname,it.qty,it.price);

}

}

fprintf(fpk,"%d",n);

fclose(fpk);

fclose(fptr);

fclose(fp);

if(found==0)

printf("\n product not found\n");

remove("asd.txt");

rename("temp2.txt","asd.txt");

remove("shop.txt");

rename("temp1.txt","shop.txt");

}

void invoice()

{

int i,j,o;

float amt,gt=0;

if(ti==0)

{printf("no products in your cart\n");

exit(0);}

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INVOICE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("================================================================================\n");

printf("product number\t| product name\t| price\t| quantity\t| amount\t|\n");

printf("================================================================================\n");

for(o=0;o<ti;o++)

{

cst[o].amount=0;

cst[o].amount=(cst[o].qty)\*(cst[o].price);

gt=gt+cst[o].amount;

}

for(j=0;j<ti;j++)

{

printf("%6d\t |%9s \t | %3d\t|%3d\t | %3d\n",cst[j].prno,cst[j].prname,cst[j].price,cst[j].qty,cst[j].amount);

}

printf("\n amount payable is %f",gt);

printf("thank you for shopping\ncome again\n");

exit(0);

}

void mdelete()

{

int i,j,f,g=0,aj=0,ag=0;

char ab[100];

FILE \*fw,\*fq;

printf("enter the product number to be deleted\n");

scanf("%d",&f);

for(i=0;i<ti;i++)

{

if(cst[i].prno==f)

{aj=cst[i].qty;

printf("\nproduct deleted successfully\n");}

continue;

cstr[g].prno=cst[i].prno;

strcpy(cstr[g].prname,cst[i].prname);

cstr[g].qty=cst[i].qty;

cstr[g].price=cst[i].price;

g++;

}

for(j=0;j<g;j++)

{

cst[j].prno=cstr[j].prno;

strcpy(cst[j].prname,cstr[j].prname);

cst[j].qty=cstr[j].qty;

cst[j].price=cstr[j].price;

}

ti=g;

printf("\n%d",aj);

fq=fopen("shop.txt","r");

fw=fopen("temp4.txt","w");

for(j=0;j<n;j++)

{

fscanf(fq,"%d",&it.prno);

fscanf(fq,"%s",it.prname);

fscanf(fq,"%d",&it.qty);

fscanf(fq,"%d",&it.price);

if(it.prno==f)

{ag=aj+it.qty;

fprintf(fw,"%d\t%s\t%d\t%d\t",it.prno,it.prname,ag,it.price);}

else

fprintf(fw,"%d\t%s\t%d\t%d\t",it.prno,it.prname,it.qty,it.price);

}

fclose(fq);

fclose(fw);

remove("shop.txt");

rename("temp4.txt","shop.txt");

}

void mdisplay()

{

int i,j,o;

printf("================================================================================\n");

printf("product number\t|product name\t|quantity\t|price\t\n");

printf("================================================================================\n");

for(o=0;o<ti;o++)

{

printf("%14d\t|%12s\t|%6d\t|%6d\n",cst[o].prno,cst[o].prname,cst[o].qty,cst[o].price);

}

}

main()

{

char ch,password[10],q[10]="yatish";

int s=0,y=0,w=1,i;

FILE \*fpe;

fpe=fopen("asd.txt","r");

fscanf(fpe,"%d",&n);

fclose(fpe);

printf("\t\*\*\*\*\*\*\*\*WELCOME TO DENIMXPRESS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t YOU ARE LUCKY TO BE SHOPPING HERE\n");

printf("\tplease note that all the products' price are discounted and are genuine\n\tso no bargain\n");

printf("\treference number%d\n",n);

q: printf("\n\n\t01. ADMINISTRATOR\n\t02. CUSTOMER\n\t03.BREAK\n");

scanf("%d",&s);

switch(s)

{

case 1: printf("\tenter password\n\t");

for(i=0;i<6;i++)

{

ch = getch();

password[i] = ch;

ch = '\*' ;

printf("%c",ch);

}

password[i]='\0';

if(strcmp(password,q))

{

printf("wrong password try again");

goto q;

}

else

goto l3;

l3: printf("\n\tacess granted\n");

printf("\t01.CREATE\n\t02.DELETE/MODIFY\n\t03.DISPLAY\n\t04.MAIN MENUE\n");

scanf("%d",&y);

switch(y)

{

case 1:create();

goto q;

case 2:edit();

goto q;

case 3:display();

goto q;

case 4: goto q;

default : printf("error\n");

goto q;

}

break;

case 2: printf("\t01.BILL\n\t02.INVOICE\n\t03.DISPLAY\n\t04.MY CART\n\t05.MAIN MENUE\n\t06. DELETE ITEMS IN CART\n");

scanf("%d",&y);

switch(y)

{

case 1:bill();

goto q;

case 2:invoice();

goto q;

case 3:display();

goto q;

case 4: mdisplay();

goto q;

case 5: goto q;

case 6: mdelete();

goto q;

default : printf("error\n");

goto q;

}

break;

case 3: printf("breaking...\n");

break;

default : printf("error\n");

break;

}

}

**Submitted by –**

**Mano Vishnu A**

**U101115FEC012**

**Section: S2**