OVERVIEW

* C++ is an object oriented language based on the C programming language.
* It can be viewed as a super set of C. Almost all of the features and constructs available in C are available in C++. However C++ is more than just an extension of C. Its additional features support the programming style known as Object Oriented Programming (OOP).
* Several features that are already available in C, such as Input and output may be implemented differently in C++.
* In C++ you may use the conventional C Input and Output routines or you may use Object Oriented Input and Output by using I/O stream library.
* Its additional features support the programming style known as Object Oriented Programming (OOP).
* Object oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Data Abstraction, Encapsulation etc.
* Encapsulation is the process of binding up of data and function into a single entity
* Abstraction is the process of representing only the essential features hiding the background details.
* Inheritance is the capability of one class to inherit the properties from another class.
* Polymorphism is the characteristic of being able to assign a different meaning or usage to something in different contexts.





HEADERFILES

* #include<fstream.h> - To include the header file “fstream” in the program. fstream is used to manage and handle binary and text files and invoke the commonly used functions like cout,cin in a C++ program.
* #include<conio.h> - This is used to provide console Input and

Output.

* #include<string.h> - This is used to manipulate string or character array.
* #include<stdlib.h> - This is used for file related Input OutputFunctions.
* #include<stdio.h> - This is used to provide Standard Input Output functions.
* #include<ctype.h> - This is used to provide Standard Input Output functions. The ctype.h header file of the C Standard Library declares several functions that are useful for testing and mapping characters.
* #include<time.h> - This is used for manipulation of current date and time.
* #include<dos.h> - It contains functions for handling interrupts, producing sound, date and time functions/
* #include<graphics.h> -This is used to invoke all the graphics functions in c++.
* #include<iomanip.h> - This is used to set the field of the output .

CLASSES AND THEIR OBJECTS

CLASS NAME: persondatabase.

CLASS DATA MEMBERS: uname[20],pw[20], temp,ptemp[20],email[40], recv.

CLASS MEMBER FUNCTIONS: signinpw( ), login(int a), sandr( ),retname( ), disp( ),

retemail( ), retrecv(), retpw().

CLASS OBJECTS: p,user.

CLASS NAME: generalproduct.

CLASS DATA MEMBERS: type1[30],desp1[100].

CLASS MEMBER FUNCTIONS: input1( ),output1( ).

CLASS OBJECTS : -

CLASS NAME: sortedproduct

CLASS DATA MEMBERS: type2[30],desp2[100],name[30],code[10] , price,flagdisp.

CLASS MEMBER FUNCTIONS:rettype1( ),rettype2( ),output2( ),input2(),retflag( ),retname( ),

retprice( ),flagup( ),sortedproduct( ).

CLASS OBJECTS:p, sp[30], temp, run.

CLASS NAME: cartitems.

CLASS DATA MEMBERS:no\_ofitems, subtotal, price,name[30].

CLASS MEMBER FUNCTIONS: discart( ),checkdisp( ),

copy(float p,char \*na,int noitm),

editnoi(intnoi),subt( ),retname( ),retprice( ),retsubt( ),retnoi().

CLASS OBJECTS:c,ck,temp[20].



SOURCE CODE

#include<fstream.h>

#include<conio.h>

#include<ctype.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<dos.h>

#include<iomanip.h>

void addproducts();

void sortproducts(int flag);

class persondatabase

{

private:

char uname[20],pw[20];

char temp,ptemp[20];

char email[40];

long recv;

public:

void signinpw();

int login(int a);

void sandr();

char\* retname()

{

return uname;

}

void disp()

{

cout<<"\n";

cout<<"USER NAME:"<<uname<<"\t PASSWORD"<<pw;

cout<<"\nEMAIL ID:";

cout<<email;

cout<<"\t RECOVERY CODE:"<<recv;

}

char\* retemail()

{

return email;

}

long retrecv()

{

return recv;

}

char\* retpw()

{

return pw;

}

};

int persondatabase::login(int adcode)

{

int flag=0;

char ch[30],ph[30];

char ad[10];

if(adcode==0)

{

cout<<"\n ENTER THE ADMIN CODE:";

gets(ad);

if(strcmpi(ad,"AD@2019")==0)

;

else

return flag;

}

persondatabase log;

ifstream fi("USERID.DAT",ios::binary);

cout<<"\n \n \n \n \t PLEASE ENTER THE USERNAME:";

gets(ch);

char temp,pw[30];int i=0;

cout<<"\n \t PLEASE ENTER THE PASSWORD:";

while(temp!='\r')

{

temp=getch();

pw[i]=temp;

cout<<"\*";

i++;

}

pw[i-1]='\0';

strcpy(ph,pw);

while(fi.read((char\*)&log,sizeof(log)))

{

if(strcmp(ch,log.retname())==0)

{

if(strcmp(ph,log.retpw())==0)

{

flag=1;

}

}

}

fi.close();

if(flag==0)

cout<<"\n \t NO ID FOUND OR INCORRECT LOGIN DETAILS";

return flag;

}

void persondatabase::sandr()

{

//searching the file

clrscr();

ifstream fi("USERID.DAT",ios::binary);

long v;

persondatabase pc;

char gn[100];

cout<<"\n Please enter the recovery mail:";

gets(gn);

cout<<"\n Please enter the 6 digit recovery code by pressing enter after each number:";

cin>>v;

int flag=0;

if(!fi)

cout<<"Error in recovering";

while(fi.read((char\*)&pc,sizeof(pc)))

{

if(strcmp(gn,pc.retemail())==0)

{

if(v==pc.retrecv())

{

flag=1;

}

}

}

if(flag ==1)

pc.disp();

else if(flag ==0)

cout<<"\n INCORRECT DETAILS:";

getch();

fi.close();

}

void persondatabase::signinpw()

{

persondatabase p;

m:

clrscr();

cout<<"\n ENTER YOU REGISTERED EMAIL ID [For password recovery]:";

gets(email);

cout<<"\n PLEASE ENTER A NEW USER NAME :";

gets(uname);

strcpy(pw,NULL);temp=NULL;

cout<<"\n PLEASE ENTER A PASSWORD OF SIZE BETWEEN 8-15:";

int i=0;

int flag=0;

while(temp!='\r')

{

temp=getch();

pw[i]=temp;

cout<<"\*";

i++;

}

pw[i-1]='\0';

char c;

cout<<"\n WANT TO SEE YOUR TYPED PASSWORD (Y/N) ?";

cin>>c;

c=tolower(c);

if(c=='y')

{ cout<<"\t"<<pw;}

int j=0; temp=NULL;

strcpy(ptemp,NULL);

if(strlen(pw)>=8 && strlen(pw)<=15)

{

cout<<"\n PLEASE RE ENTER TO CONFIRM YOUR PASSWORD:";

while(temp!='\r')

{

temp=getch();

ptemp[j]=temp;

cout<<"\*";

j++;

}

ptemp[j-1]='\0';

if(strcmp(pw,ptemp)==0)

{ cout<<"\n Password matched !";

flag=1;

getch();

}

else

{

cout<<"\n Incorrect password";

flag=0;

getch();

}

}

else

{

cout<<"\n PASSWORD HAS LESS CHARACTERS --> ";

flag =0;

}

if(flag==1)

{

randomize();

long a;

a=random(1000000)+100000;

cout<<"\n PLEASE TAKE A MOMENT TO NOTE DOWN THE ACCOUNT RECOVERY CODE !!";

recv=a;

cout<<"\n RECOVERY CODE:"<<recv;

getch();

cout<<"\n YOUR ID HAS BEEN SUCCESSFULLY CREATED !";

getch();

}

else if(flag==0)

{

clrscr();

goto m;

}

}

class generalproduct

{

protected:

char type1[30],desp1[100];

public:

void input1()

{

cout<<"\n GENERAL TYPE:";

gets(type1);

}

void output1()

{

cout<<"\n GENERAL TYPE:"<<type1;

}

};

class sortedproduct:protected generalproduct

{

protected:

char type2[30],desp2[100],name[30],code[10];

float price;

int flagdisp;

public:

sortedproduct()

{

strcpy(type1,"-");

strcpy(type2,"-");

strcpy(desp2,"-");

strcpy(name,"-");

strcpy(code,"0000");

strcpy(desp1,"-");

price=0.0;

flagdisp=0;

}

int retflag()

{

return flagdisp;

}

char\* retname()

{

return name;

}

float retprice()

{

return price;

}

void flagup()

{

flagdisp=1;

}

void input2()

{

input1();

cout<<"\n SPECIFIC TYPE:";

gets(type2);

cout<<"\n PRODUCT CODE:";

gets(code);

cout<<"\n NAME OF THE PRODUCT:";

gets(name);

cout<<"\n PRICE:";

cin>>price;

cout<<"\n BRIEF DESCRIPTION:";

gets(desp2);

}

void output2()

{

output1();

cout<<"\n CATEGORY:"<<type2;

cout<<"\n NAME OF THE PRODUCT:"<<name;

cout<<"\n BRIEF DESCRIPTION:"<<desp2;

cout<<"\n PRICE:"<<price;

}

char\* rettype1()

{

return type1;

}

char\* rettype2()

{

return type2;

}

};

class cartitems

{

int no\_ofitems;

float subtotal;

float price;

char name[30];

public:

void discart()

{

cout<<"\n \t NAME OF THE PRODUCT:"<<name;

cout<<"\n \t PRICE:"<<price<<" X "<<no\_ofitems;

subt();

}

void checkdisp()

{

cout<<"NAME:"<<name<<"\t PRICE:"<<price<<"x"<<no\_ofitems;

}

void copy(float p,char \*na,int noitm)

{

price =p;

strcpy(name,na);

no\_ofitems=noitm;

subtotal=(price\*no\_ofitems);

}

void editnoi(int noi)

{

no\_ofitems=noi;

subtotal=(price\*no\_ofitems);

}

void subt()

{

cout<<"\n SUBTOTAL:"<<subtotal;

}

char\* retname()

{

return name;

}

float retprice()

{

return price;

}

float retsubt()

{

return subtotal;

}

int retnoi()

{

return no\_ofitems;

}

};

void bill(float p,persondatabase user);

inline void erase\_cart()

{

ofstream fo("cart.dat",ios::binary|ios::trunc);

fo.close();

}

void addlist(sortedproduct tc,int n)

{

clrscr();

cout<<"\n \n \n \n \n \n\t ADDING TO CART :";

for(int s=0;s<15;s++)

{

delay(150);

cout<<"|||";

}

getch();

ofstream fo("cart.dat",ios::binary|ios::app);

cartitems c;

float p;

char na[30];

p=tc.retprice();

strcpy(na,tc.retname());

c.copy(p,na,n);

fo.write((char\*)&c,sizeof(c));

fo.close();

}

void edit(cartitems ced[20],float &ta,int n)

{

char ch;

cartitems ck;

ofstream fo("tempcart.dat",ios::binary|ios::trunc);

ifstream fi("cart.dat",ios::binary);

cartitems temp[20];

int i,flag=0;

int noi; int num;

clrscr();

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

{

ced[i].discart();

cout<<"\n";

}

char nam[20];

cout<<"\n ENTER THE PRODUCT NAME TO BE EDITED/DELETED:";

gets(nam);

cout<<"\n HOW WOULD YOU LIKE TO EDIT ?:";

cout<<"\n 1.Change the number of item(s) \n 2.Delete the Product from cart";

cout<<"\n ENTER YOUR CHOICE PLEASE:";

cin>>num;

getch();

if(num==1)

{

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

{

if(strcmpi(nam,ced[i].retname())==0)

{

flag=1;

cout<<"\n Current number of item(s):"<<ced[i].retnoi();

cout<<"\n Enter the new quantity :";

cin>>noi;

ced[i].editnoi(noi);

cout<<"\n CHANGES MADE!";

getch();

break;

}

}

}

else if(num==2)

{

while(fi.read((char\*)&ck,sizeof(ck)))

{

if(strcmpi(nam,ck.retname())==0)

{

flag=1;

cout<<"\n";

cout<<ck.retname()<<"--> Item deleted \n";

ta=ta-ck.retsubt();

}

else

{

fo.write((char\*)&ck,sizeof(ck));

}

}

fo.close();

remove("cart.dat");

rename("tempcart.dat","cart.dat");

}

if(flag==1&&num==1)

{

cartitems t;

ofstream fo("cart.dat",ios::binary|ios::trunc);

if(!fo)

cout<<"\n Cart editing failed";

for(int i=0;i<n;i++)

{

t=ced[i];

fo.write((char\*)&t,sizeof(t));

}

fo.close();

}

else if(flag==0)

cout<<"\n Item not found in you cart!";getch();

}

void editcart(cartitems ced[20],float &ta,int n)

{

int num;

char ch;

cout<<"\n Would you like to edit your cart (y/n) ?";

cin>>ch;

do{

ch=tolower(ch);

if(ch=='y')

{

edit(ced,ta,n);

break;

}

else if(ch=='n')

;

else

{

cout<<"\n Please enter the correct character !";

cin>>ch;

}

} while(ch!='n');

}

void checkout(float p,persondatabase user)

{

clrscr();

float tot=0;

ifstream f1("cart.dat",ios::binary);

cartitems c;

textcolor(YELLOW);

clrscr();

cout<<"\n===================================================\n";

while(f1.read((char\*)&c,sizeof(c)))

{

c.checkdisp();

tot+=c.retsubt();

cout<<"\n";

}

cout<<"\nTOTAL AMOUNT:"<<tot;

cout<<"\n===================================================";

cout<<"\n DO YOU WISH TO CHECK OUT ?(Y/N)";

char ch;

cin>>ch;

ch=tolower(ch);

if(ch=='y')

{

clrscr();

textcolor(RED+128);

clrscr();

cout<<"\n \t \t PAY AMOUNT OF:"<<tot<<"Rs PLEASE";

bill(p,user);

clrscr();

}

}

float dispcart() //called from mainfunc

{

clrscr();

cout<<"\n \n \n \n \n \n \t LOADING CART:";

for(int s=0;s<15;s++)

{

delay(150);

cout<<"|||";

}

getch();

clrscr();

cartitems c,ced[20];

float totalamt=0;

ifstream f1("cart.dat",ios::binary);

if(!f1)

{

cout<<"\Cart failed to open";

}

int i=0;

while(f1.read((char\*)&c,sizeof(c)))

{

ced[i]=c;i++;

c.discart();

totalamt+=c.retsubt();

cout<<"\n";

}

f1.close();

cout<<"\n TOTAL AMOUNT --->:"<<totalamt;

editcart(ced,totalamt,i);

// checkout();

return totalamt;

}

void bill(float totalamt,persondatabase user)

{

getch();

ofstream fi("bill.txt",ios::trunc);

char name[30]; int n;

float price,subtot;

ifstream f1("cart.dat",ios::binary);

cartitems c;

fi<<"====================================================================";

fi<<"\n";

fi<<"\n \t User ID:"<<user.retname();

getch();

if(!fi)

{ cout<<" BILL cant be opened "; getch();}

if(!f1)

{ cout<<" CART cant be opened";getch();}

while(f1.read((char\*)&c,sizeof(c)))

{

price=c.retprice();

strcpy(name,c.retname());

subtot+=c.retsubt();

n=c.retnoi();

fi<<"\tName:"<<name<<"\tPrice:"<<price<<" x "<<n<<" "<<"\n";

}

fi<<"\tTotal:"<<totalamt<<"\n";

fi<<"====================================================================";

f1.close();

fi.close();

}

void disp(char type2[30])

{

sortproducts(3);

int i,st=0,end=5,ct=0,e=5;

int input;

ifstream fi("Products.dat",ios::binary);

if(!fi)

{ cout<<" file failed to open"; getch(); }

sortedproduct temp[20],run;

while(fi.read((char\*)&run,sizeof(run)))

{

if(strcmp(run.rettype2(),type2)==0)

{

temp[ct]=run;

ct++;

}

}

input=end+5;

do

{

if(input>=st&&input<=end)

{

clrscr();

temp[input-1].output2();

int b;

cout<<"\n 03 -->TO SELECT";

cout<<"\n ANY NUMBER other than 03-->GO BACK";

cout<<"\n Enter your choice:";

cin>>b;

if(b==03)

{

int noi;

cout<<"\n NO OF ITEMS TO BE ADDED TO CART:";

cin>>noi;

sortedproduct tc;

tc=temp[input-1];

addlist(tc,noi);

cout<<"\n \t \tITEM(S) ADDED TO CART!";

getch();

textcolor(CYAN);

clrscr();

}

}

else if(input==end+1)

{

st+=e;

end+=e;

}

else if(input==end+2)

{

st-=e;

end-=e;

}

textcolor(CYAN);

clrscr();

cout<<"\n \t \t PRODUCTS: \n";

for(i=st;i<end;i++)

{

{

cout<<i+1<<". \t"<<temp[i].retname();

cout<<"\n";

}

}

if(i>e-1)

cout<<"\n"<<end+1<<"-->FOR NEXT";

else if(end>e)

{ cout<<"\n"<<end+2<<"-->FOR PREVIOUS";}

cout<<"\n0 -->FOR EXIT";

cout<<"\n ENTER THE CHOICE:";

cin>>input;

}while(input!=0);

fi.close();

return;

}

void displaytype2(char type1[30])

{

textcolor(CYAN);

clrscr();

sortproducts(2);

int i,st=0,end=5;

int e=5;

int count=0;

int input;

ifstream fi("Products.dat",ios::binary);

if(!fi)

{

cout<<" file failed to open";

getch();

}

sortedproduct sp[10],temp;

count=0;

while(fi.read((char\*)&temp,sizeof(temp)))

{

if(strcmpi(temp.rettype1(),type1)==0)

{

sp[0]=temp;

count++;

break;

}

}

fi.seekg(0,ios::beg);

while(fi.read((char\*)&temp,sizeof(temp)))

{

if(strcmpi(temp.rettype1(),type1)==0)

{

if(strcmpi(sp[count-1].rettype2(),temp.rettype2())!=0)

{

sp[count]=temp;

count++;

}

}

}

getch();

fi.close();

clrscr();

do

{

clrscr();

if(input>=st&&input<=end)

{

char typo[30];

strcpy(typo,sp[input-1].rettype2());

getch();

disp(typo);

getch();

}

else if(input==end+1)

{

st+=e;

end+=e;

}

else if(input==end+2)

{

st-=e;

end-=e;

}

clrscr();

cout<<"\n \t \t CATEGORIES: \n";

for(i=st;i<end;i++)

{

cout<<i+1<<'.'<<"\t"<<sp[i].rettype2();

cout<<"\n";

}

if(i<count)

cout<<"\n"<<end+1<<"-->FOR NEXT";

if(end>e)

{ cout<<"\n"<<end+2<<"-->FOR PREVIOUS";}

cout<<"\n0 -->FOR EXIT";

cout<<"\n Enter your choice :";

cin>>input;

getch();

}while(input!=0);

}

void displaytype1()

{

sortproducts(1);

ifstream fi("Products.dat",ios::binary);

sortedproduct sp[30],temp;

fi.read((char\*)&temp,sizeof(temp));

sp[0]=temp;

fi.seekg(0,ios::beg);

int n=0;

char t[50][30];

int flag;

while(fi.read((char\*)&temp,sizeof(temp)))

{

if(strcmpi(sp[n].rettype1(),temp.rettype1())!=0)

{

sp[n+1]=temp;

n++;

}

}

getch();

fi.close(); int s,e=5;

int i,st=0,end=5,input;

clrscr();

do

{

clrscr();

if(input>=st&&input<=end)

{

char type1[30];

cout<<sp[input-1].rettype1();

getch();

strcpy(type1,sp[input-1].rettype1());

displaytype2(type1);

textcolor(MAGENTA);

clrscr();

}

else if(input==end+1)

{

st+=e;

end+=e;

}

else if(input==end+2)

{

st-=e;

end-=e;

}

textcolor(CYAN);

clrscr();

cout<<"\n \t \t TYPES: \n";

for(i=st;i<end;i++)

{

cout<<i+1<<'.'<<"\t"<<sp[i].rettype1();

cout<<"\n";

}

if(i<n)

cout<<"\n"<<end+1<<"-->FOR NEXT";

if(end>e)

{ cout<<"\n"<<end+2<<"-->FOR PREVIOUS";}

cout<<"\n0 -->FOR EXIT";

cout<<"\n Enter your choice :";

cin>>input;

}while(input!=0);

}

void sortproducts(int flag)

{

clrscr();

sortedproduct sp[30],temp;

int n=0;

fstream f("Products.dat",ios::binary|ios::nocreate|ios::in|ios::out);

while(f.read((char\*)&temp,sizeof(temp)))

{

sp[n]=temp;

n++;

}

f.close();

for(int i=0;i<n;i++)

{

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

{

if(flag==3)

{ if(strcmp(sp[i].retname(),sp[j].retname())<0)

{

temp=sp[i];

sp[i]=sp[j];

sp[j]=temp;

}

}

else if(flag==1)

{ if(strcmp(sp[i].rettype1(),sp[j].rettype1())<0)

{

temp=sp[i];

sp[i]=sp[j];

sp[j]=temp;

}

}

else if(flag==2)

{

if(strcmp(sp[i].rettype2(),sp[j].rettype2())<0)

{ temp=sp[i];

sp[i]=sp[j];

sp[j]=temp;

}

}

}

}

f.open("Products.dat",ios::binary|ios::in|ios::out);

for(int k=0;k<n;k++)

{

temp=sp[k];

f.write((char\*)&temp,sizeof(temp));

}

f.close();

return;

}

void deleteproducts()

{

clrscr();

sortedproduct p;

char c;

char ch[30];

int flag=0;

ifstream fi("Products.dat",ios::binary);

ofstream fo;

fo.open("prodc.dat",ios::binary|ios::trunc);

textcolor(MAGENTA);

clrscr();

strcpy(ch,NULL);

int totalcount=0;

cout<<"\n"<<"TYPE1"<<"\t\t\t\t"<<"TYPE2"<<"\t\t\t"<<"NAME";

while(fi.read((char\*)&p,sizeof(p)))

{

totalcount++;

cout<<"\n"<<p.rettype1()<<setw(30)<<p.rettype2()<<setw(35)<<p.retname();

}

fi.close();

fi.open("Products.dat",ios::binary);

cout<<"\n Enter the correct name/category/type of the product(s) to be deleted:";

gets(ch);

int count=0;

int delcount=0;

while(fi.read((char\*)&p,sizeof(p)))

{

if(strcmp(ch,p.retname())==0)

{

cout<<"\nTHE PRODUCT:\t"<<p.retname()<<"\t HAS BEEN DELETED !";

flag=1;

delcount++;

}

if(strcmp(ch,p.rettype1())==0)

{

cout<<"\nTHE TYPE 1:\t"<<p.retname()<<"\t HAS BEEN DELETED !";

flag=1;

delcount++;

}

if(strcmp(ch,p.rettype2())==0)

{

cout<<"\nTHE TYPE 2:\t"<<p.retname()<<"\t HAS BEEN DELETED !";

flag=1;

delcount++;

}

else

{

fo.write((char\*)&p,sizeof(sortedproduct));

count++;

}

}

if(flag==0)

cout<<"\n\nPRODUCT/CATEGORY/TYPE NOT FOUND!";

fo.close();

fi.close();

remove("Products.dat");

if(rename("prodc.dat","Products.dat")==0)

cout<<"\n FILE RENAMED";

else

cout<<"\n FILE RENAMING FAILED";

getch();

textcolor(RED);

fi.open("Products.dat",ios::binary);

clrscr();

cout<<"\n============================================================";

while(fi.read((char\*)&p,sizeof(p)))

{

cout<<"\n NAME:";

cout<<p.retname()<<"\t CATEGORY:"<<p.rettype2()<<"\t TYPE:"<<p.rettype1();

}

cout<<"\n============================================================";

getch();

textcolor(LIGHTGREEN);

clrscr();

cout<<"\n \t NO OF RECORDS BEFORE DELETION:"<<totalcount;

cout<<"\n \t NO OF RECORDS REMAINING AFTER DELETION: "<<count;

cout<<"\n \t NO OF RECORDS DELETED:"<<delcount;

cout<<"\n PRESS ENTER TO CONTINUE";

getch();

textcolor(MAGENTA);

clrscr();

fi.close();

}

void addproducts()

{

clrscr();

sortedproduct p;

char c;

ofstream fo("Products.dat",ios::binary|ios::app);

do

{

cout<<"\n Enter the detailsof the new product:";

p.input2();

fo.write((char\*)&p,sizeof(sortedproduct));

cout<<"\n Do you want to continue adding products ?(y/n):";

cin>>c;

c=tolower(c);

if(c=='y')

continue;

}while(c!='n');

fo.close();

}

void mainmenu()

{

int num;

char ch;

clrscr();

float p;

persondatabase user;

do

{

textcolor(GREEN);

clrscr();

cout<<"\n \t Hi there!";

cout<<"\n \t What would you like to do ?";

cout<<"\n \t 1.Start to pick out \n \t 2.Check,what you have added to cart \n \t 3.Clear the cart or Delete cart \n \t 4.Checkout the cart \n \t 5.Exit ";

cout<<"\n \t Enter the your choice to move on:";

cin>>num;

switch(num)

{

case 1:

clrscr();

// int flag=1;

int flag=user.login(1);

if(flag==1)

displaytype1();

else

{ cout<<"\n \n\t PLEASE Log in OR sign up BEFORE SHOPPING";

getch();

}

clrscr();

break;

case 2:

clrscr();

p=dispcart();

break;

case 3:

clrscr();

cout<<"\n \t ARE YOU SURE THAT YOU WANT TO DELETE YOUR SELECTED ITEMS ?(Y/N)";

cin>>ch;

ch=tolower(ch);

del:

if(ch=='y')

{ erase\_cart(); cout<<"\n CART DELETED!";getch(); }

else if(ch=='n')

break;

else

{ cout<<"\n \t It seems you have entered wrong character other than y/n !";

cout<<"\n \t Please type again correctly!";

cin>>ch;

goto del;

}

break;

case 4:

checkout(p,user);

break;

case 5:

cout<<"\n \t Thanks for shopping!";

getch();

break;

default:

cout<<"\n \t OOPS!! YOU HAVE ENTERED A WRONG OPTION";

getch();

break;

}

}while(num!=5);

}

void circleintro()

{

clrscr();

int gdriver = DETECT,gmode;

int x,y,i;

initgraph(&gdriver,&gmode,"C:\\TC\\BGI");

x=getmaxx()/2;

y=getmaxy()/2;

for(i=40;i<200;i++)

{

settextstyle(TRIPLEX\_FONT, HORIZ\_DIR, 4);

delay(100);

setcolor(i/10);

circle(x,y,i-10);

outtextxy(x-35,y-20,"A.D.S");

setcolor(i/2);

settextstyle(6, HORIZ\_DIR, 4);

outtextxy(x-160,y+120,"THE A-Z SHOPPING SITE");

}

getch();

closegraph();

}

void main()

{

circleintro();

persondatabase p;

int cho;

char c;

do

{ textcolor(RED);

clrscr();

cout<<"\n \t \t WELCOME TO THE A.D.S SHOPPING SITE";

cout<<"\n \t \t PLEASE SELECT THE OPTION BELOW TO PROCEED";

cout<<"\n \t 1.GO IN FOR SHOPPING \n \t 2.NEW TO THIS ? ->SIGN UP \n \t 3.ADMIN MODE \n \t 4.FORGOTTEN USER NAME / PASSWORD \n \t 5.EXIT!";

cout<<"\n ENTER YOUR CHOICE:";

cin>>cho;

switch(cho)

{

case 1:

clrscr();

mainmenu();

break;

case 2:

clrscr();

ofstream fo("USERID.DAT",ios::binary|ios::app);

p.signinpw();

fo.write((char\*)&p,sizeof(persondatabase));

fo.close();

break;

case 3:

int ad;

cout<<"\n TO ADD / DELETE SPECIFIC PRODUCT(S) / DELETE ALL PRODUCTS ?:(1/2/3):";

cin>>ad;

int flag=p.login(0);

if(flag==1&&ad==1)

{ addproducts();break;}

else if(flag==1&&ad==2)

{

do

{

clrscr();

deleteproducts();

cout<<"\n Do you want to continue deleting product(s) products ?(y / n ):";

cin>>c;

c=tolower(c);

if(c=='n')

break;

}while(c=='y');

}

else if(ad==3&&flag==1)

{

ofstream fo("Products.dat",ios::binary|ios::trunc);

fo.close();

}

else if(flag==0)

{

textcolor(RED+128);

clrscr();

cout<<"\n \t \t \t \t ACCESS DENIED!";

getch();

break;

}

break;

case 4:

clrscr();

p.sandr();

break;

case 5:

cout<<"\n THANK YOU FOR VISTING US !";

getch();

break;

default:

cout<<"\n WRONG OPTION";

}

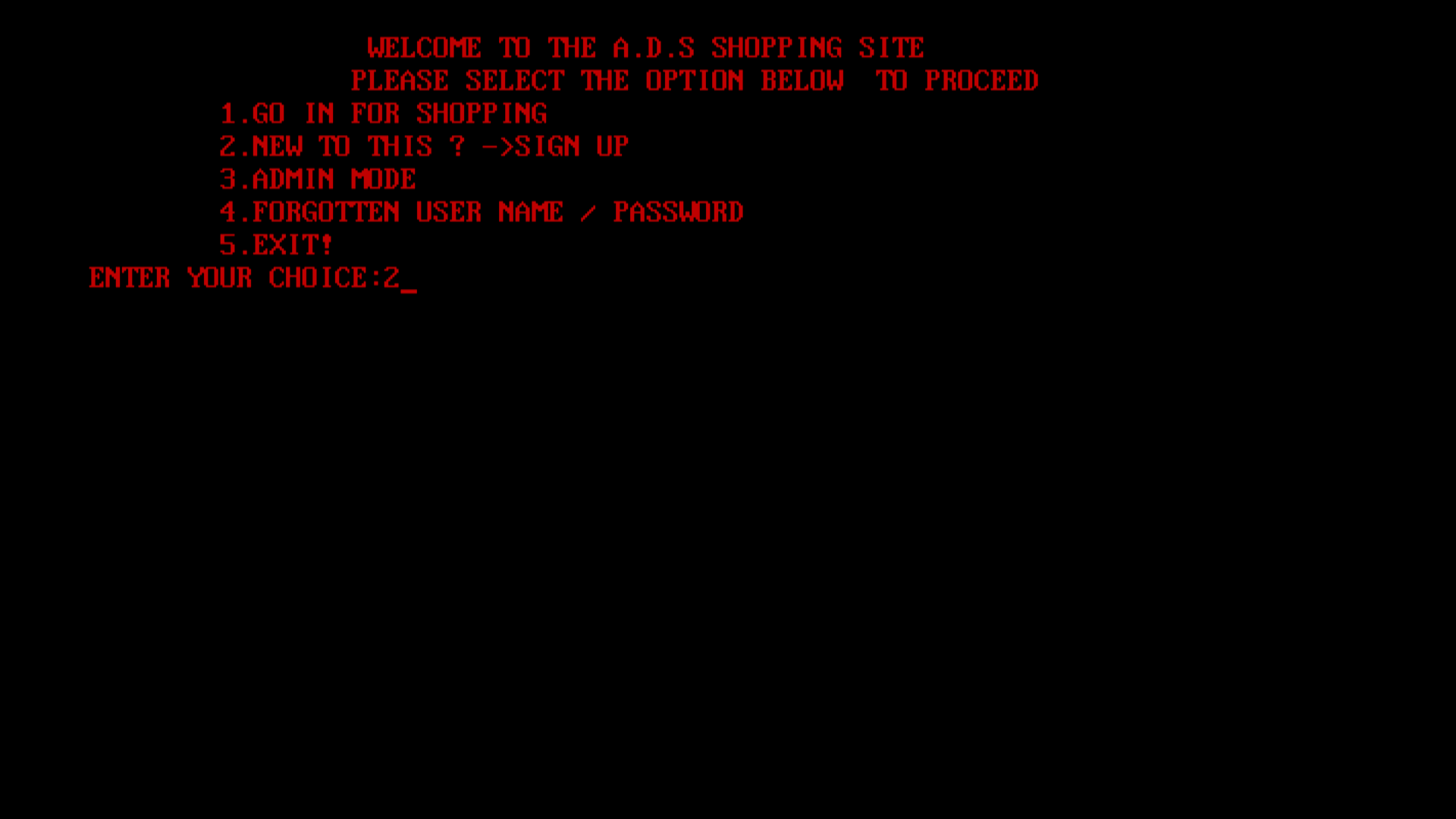
}while(cho!=5);

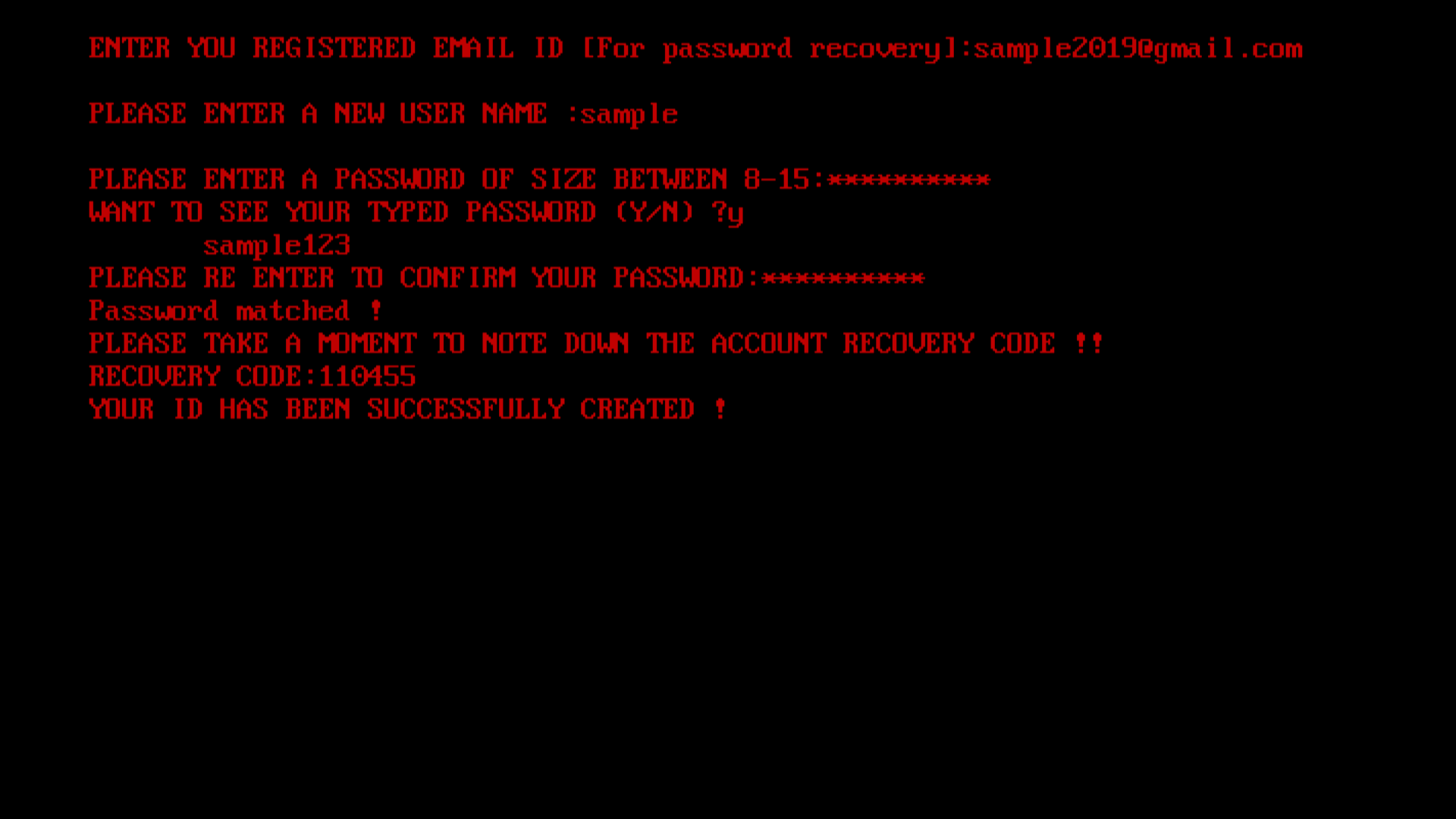
clrscr();

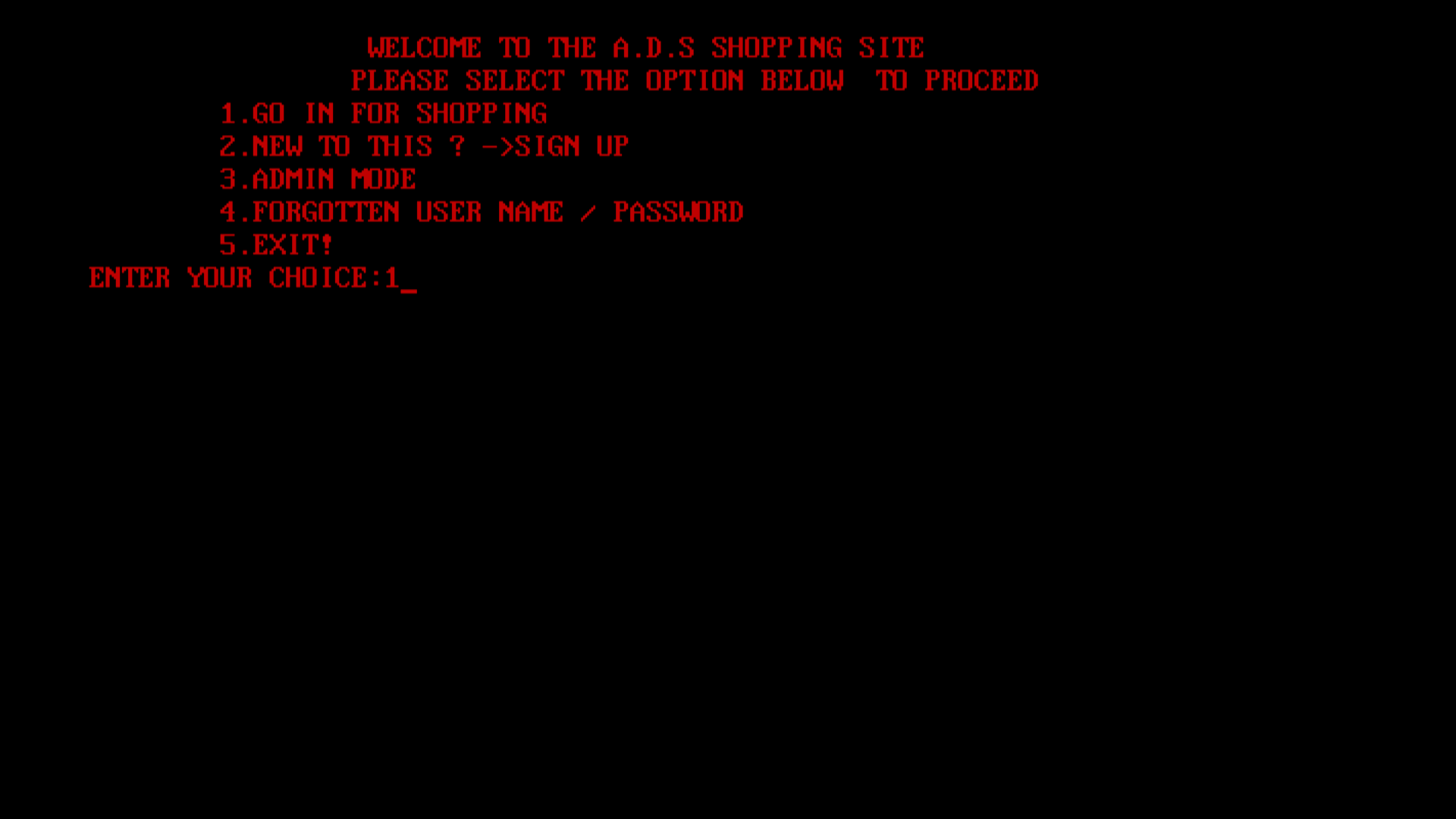
}

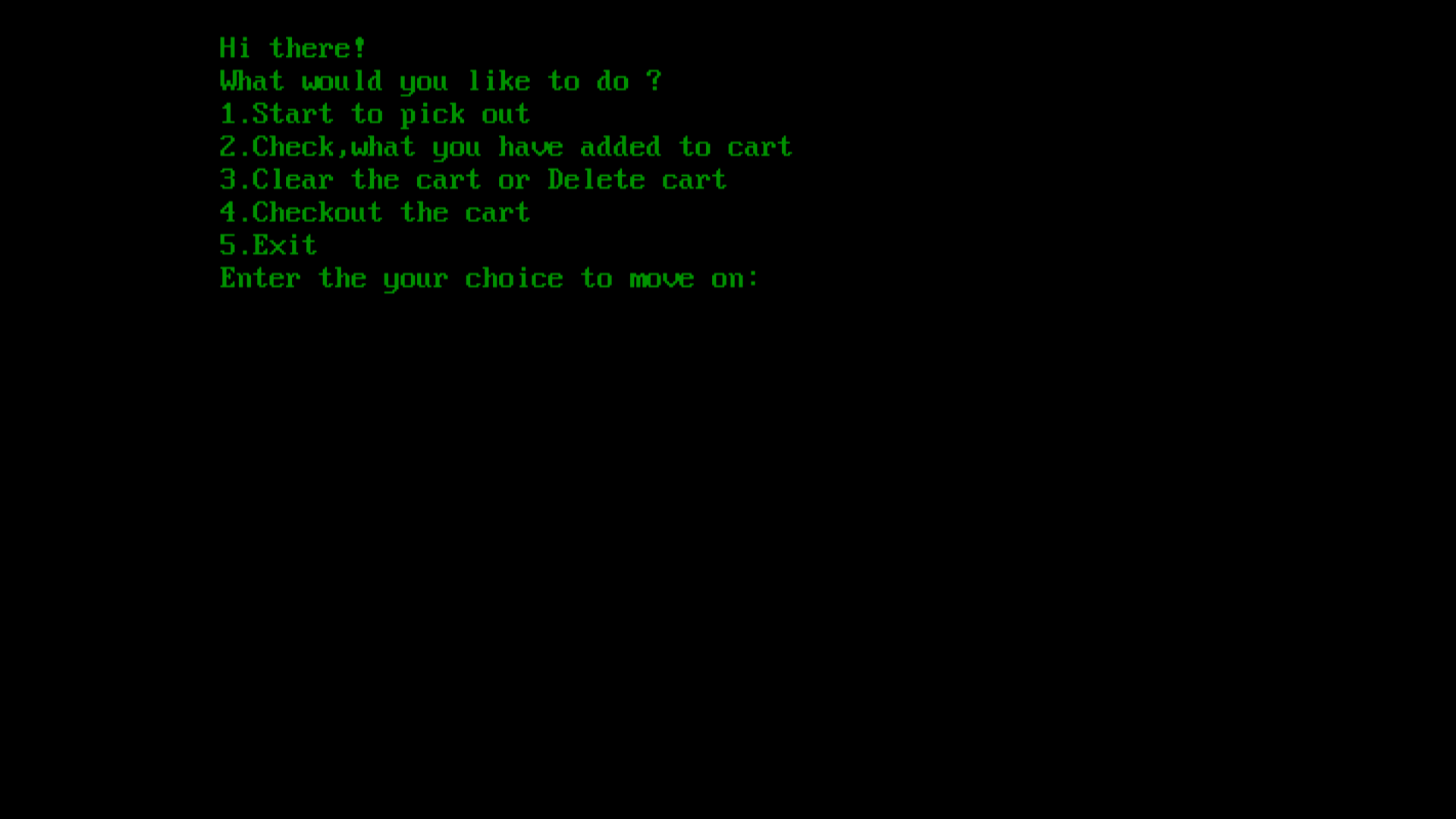
OUTPUT

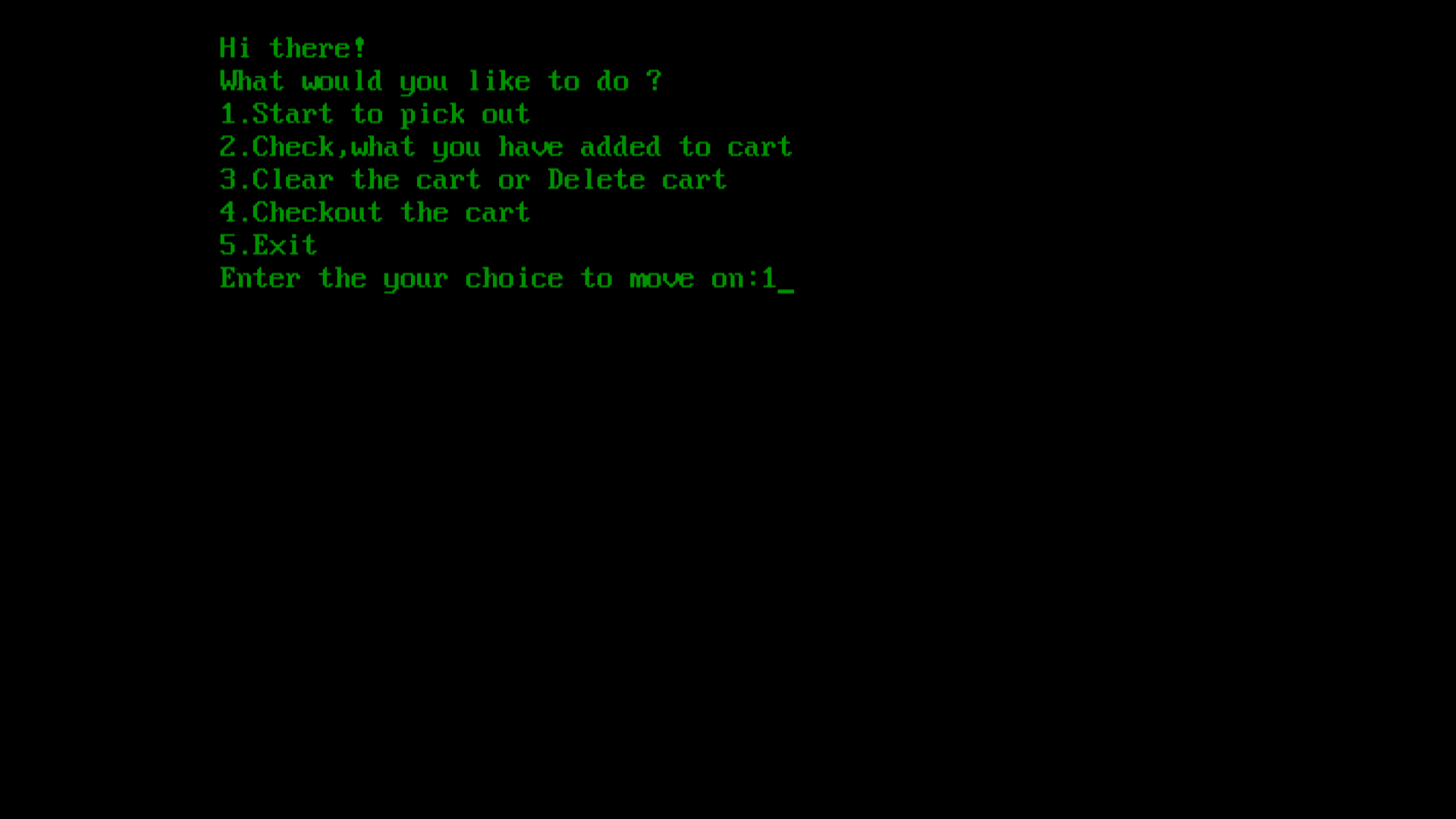




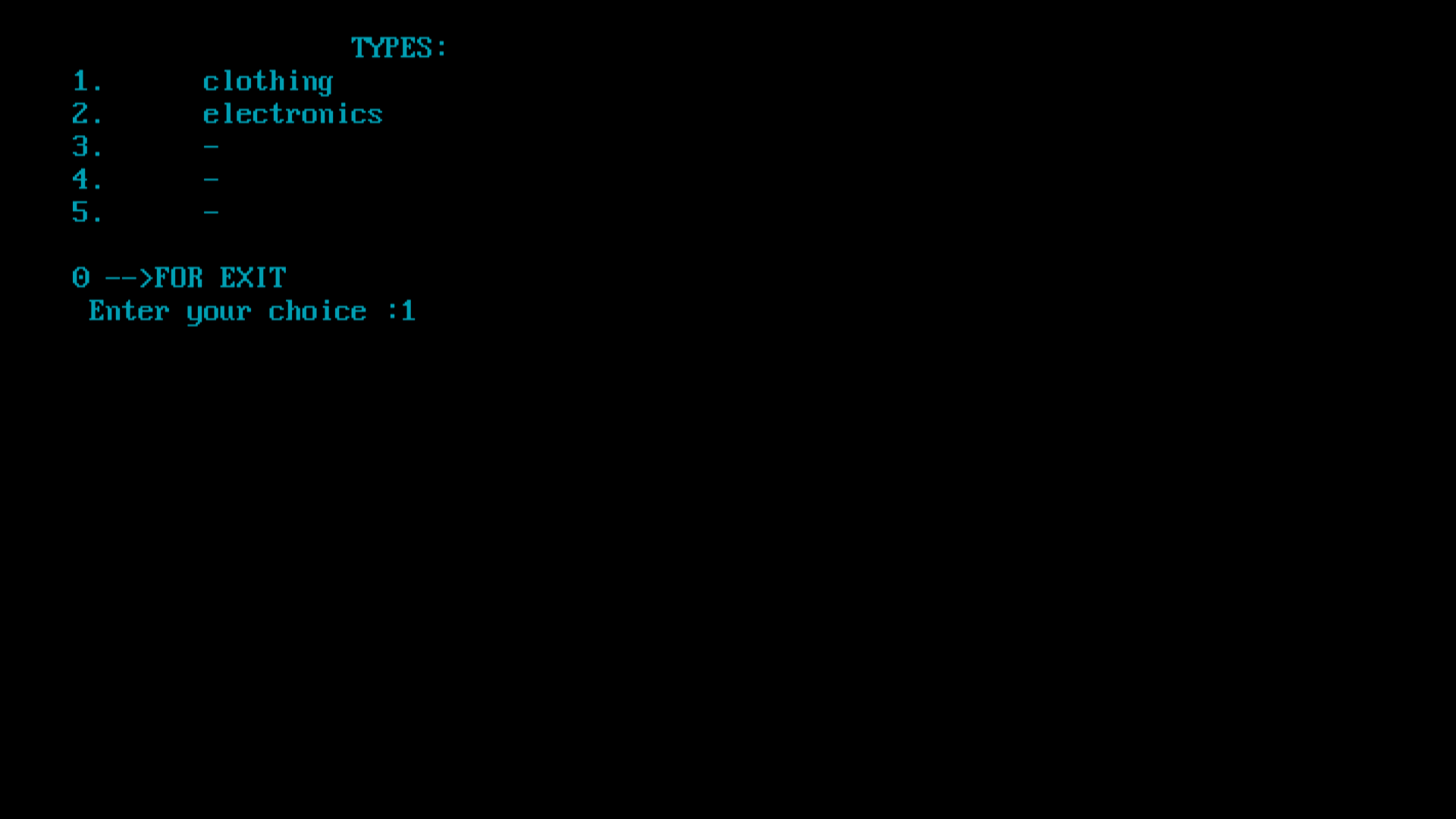


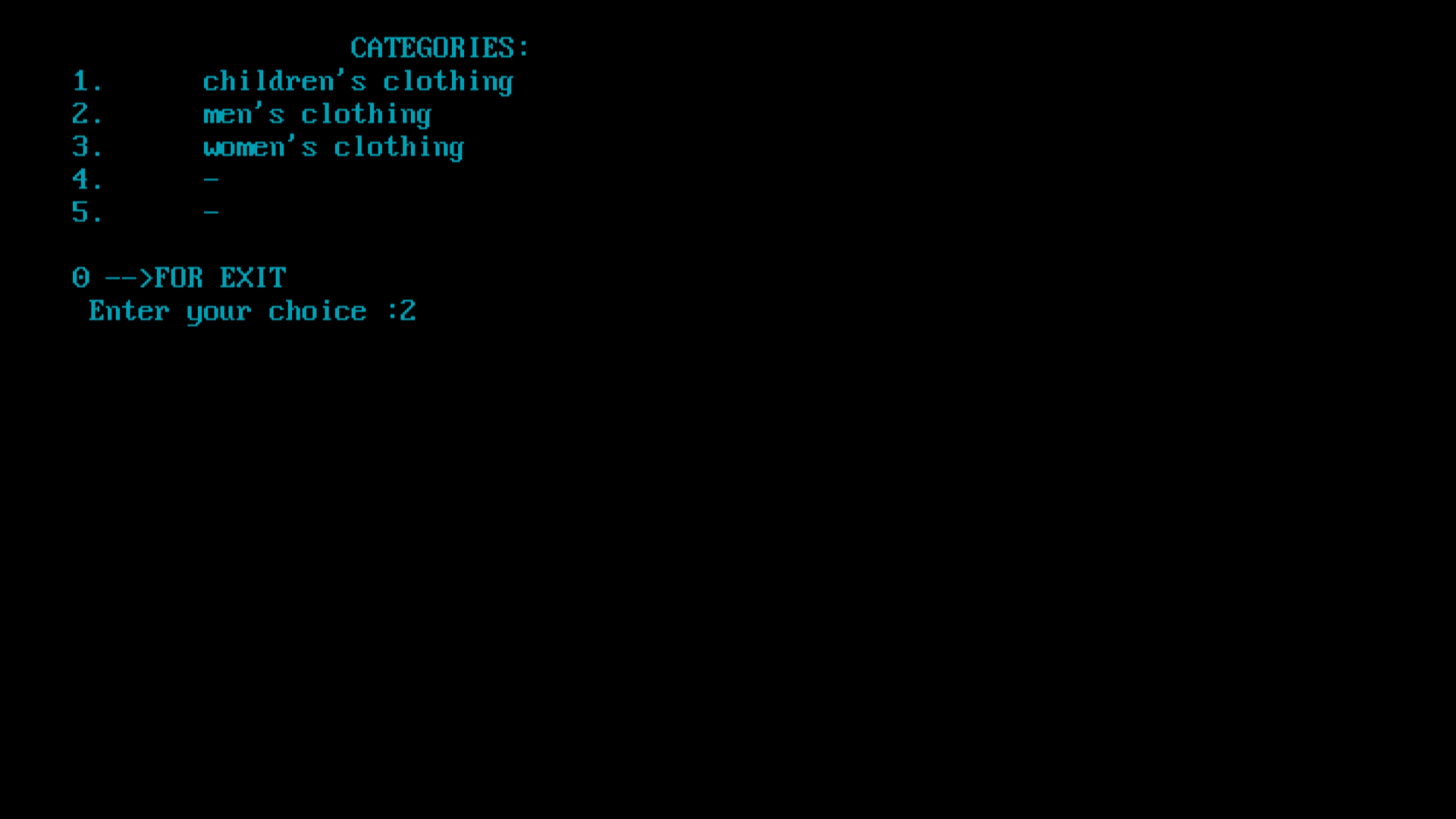


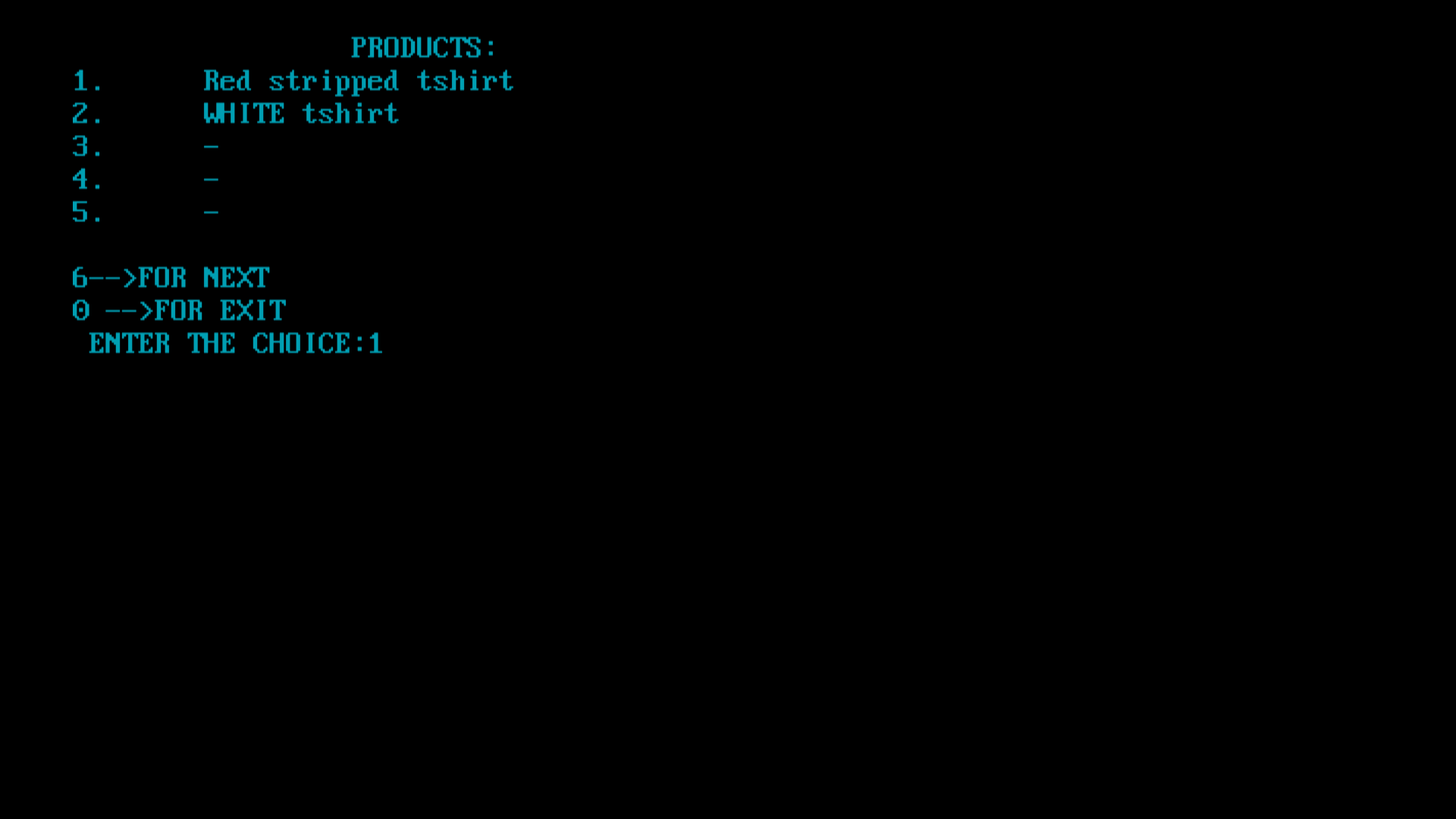


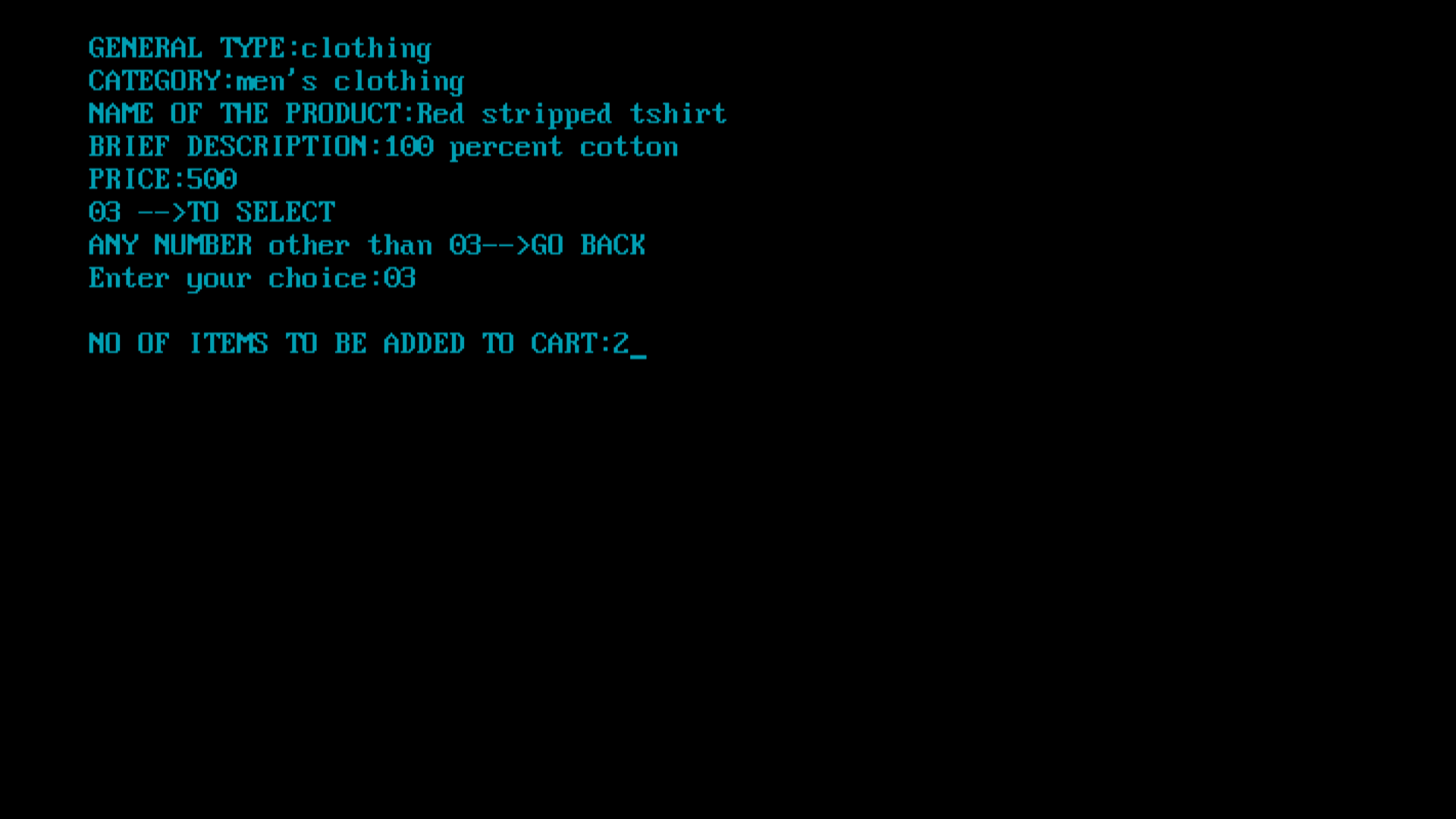


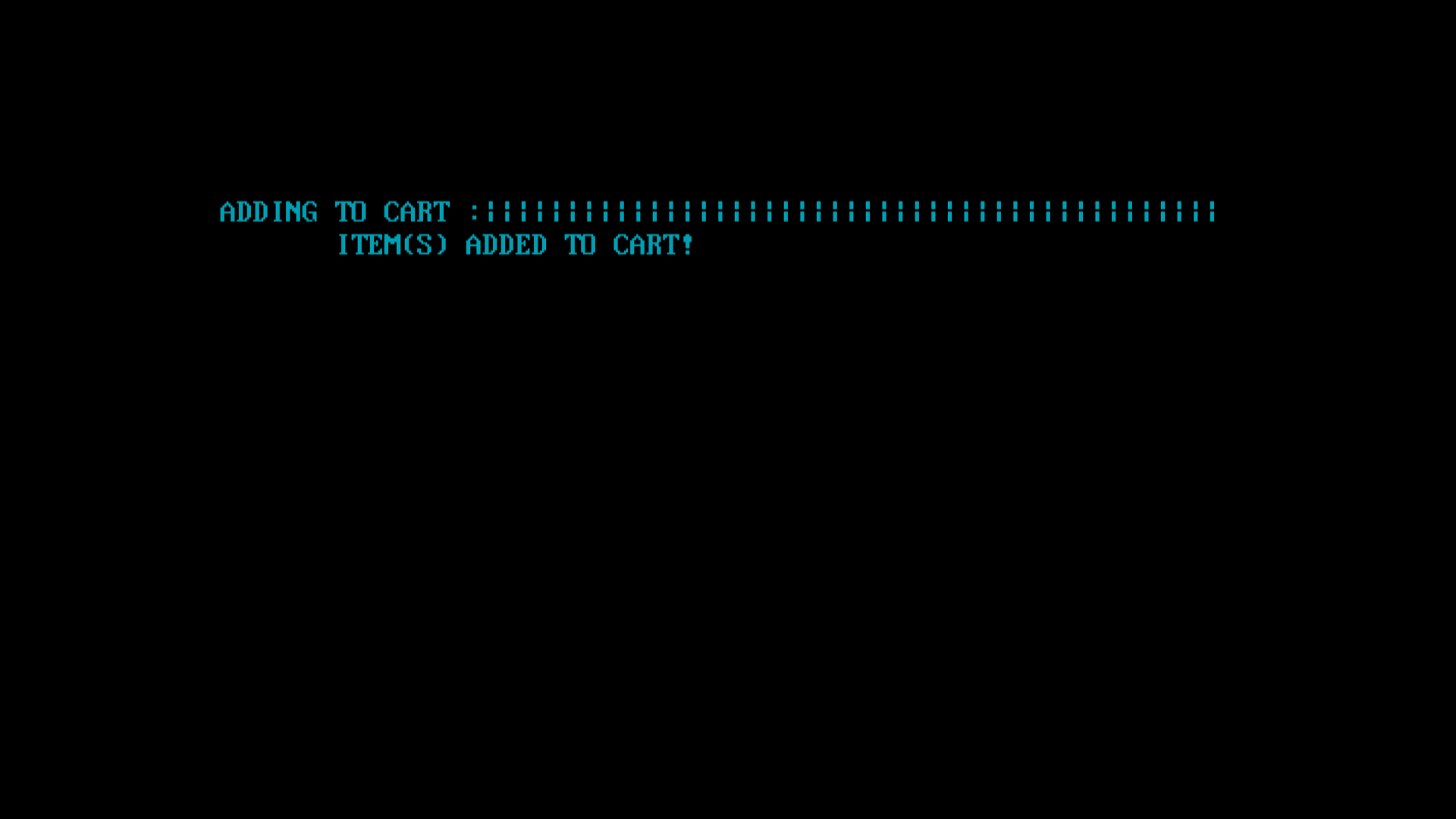


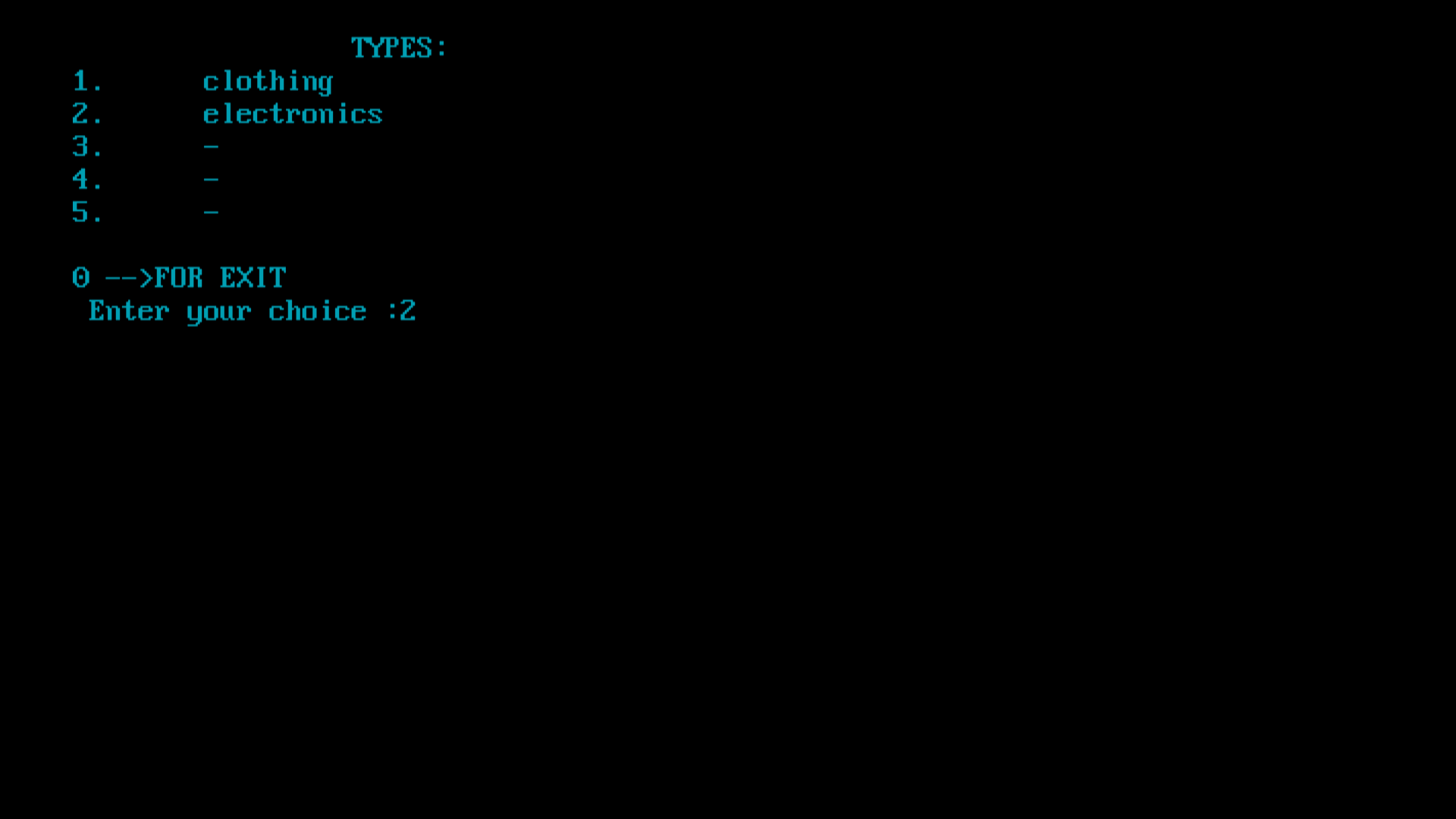


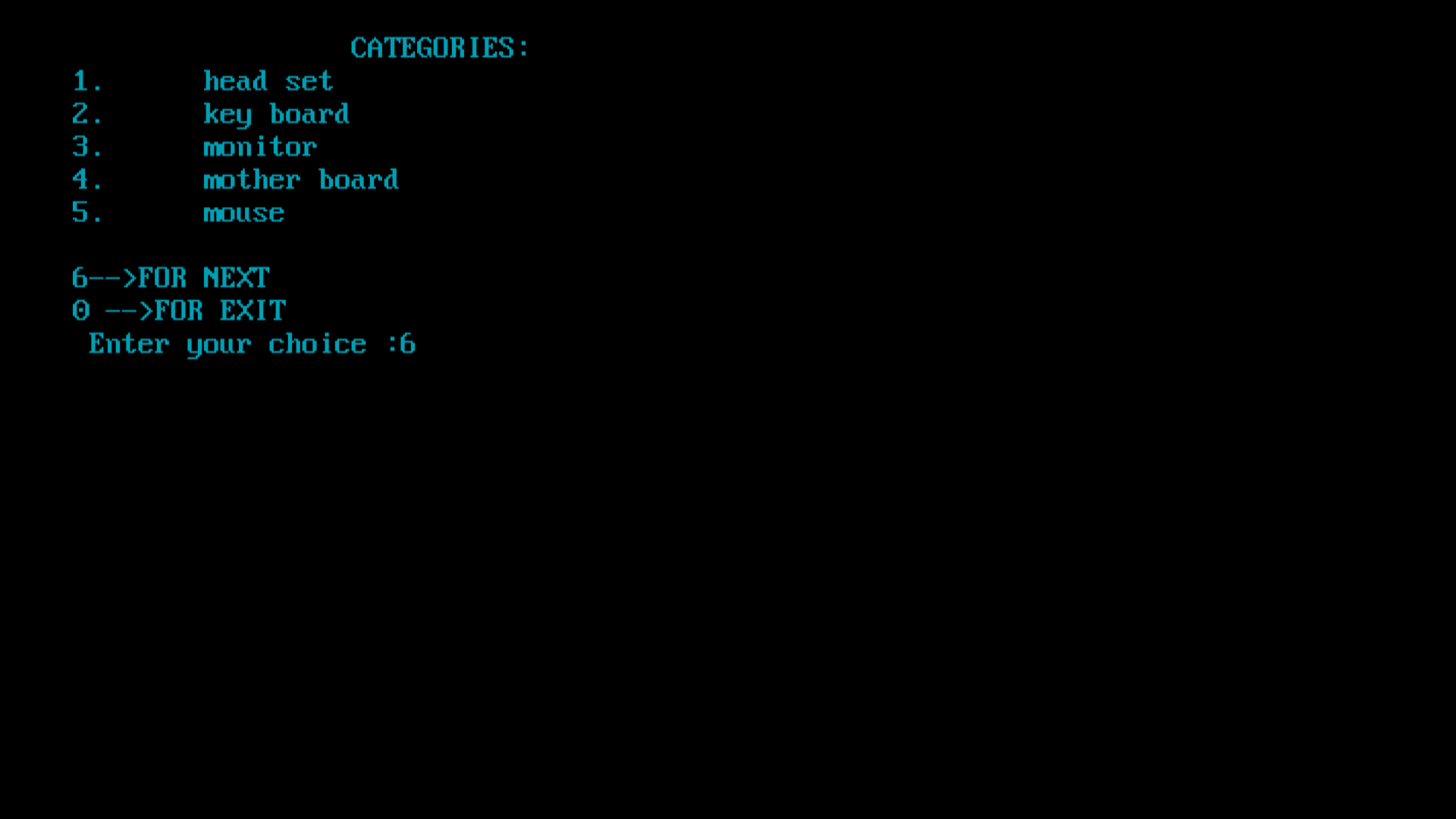


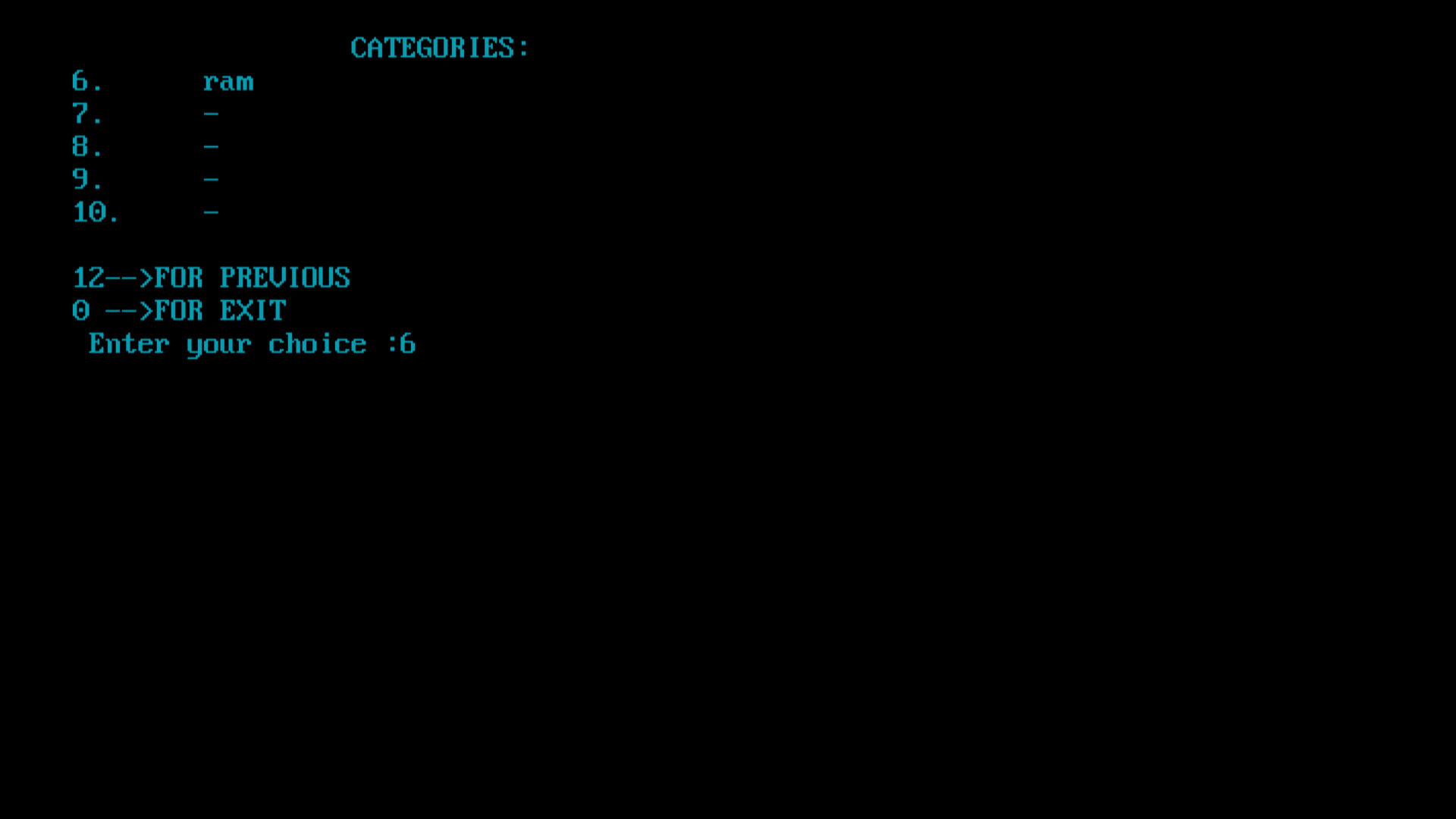


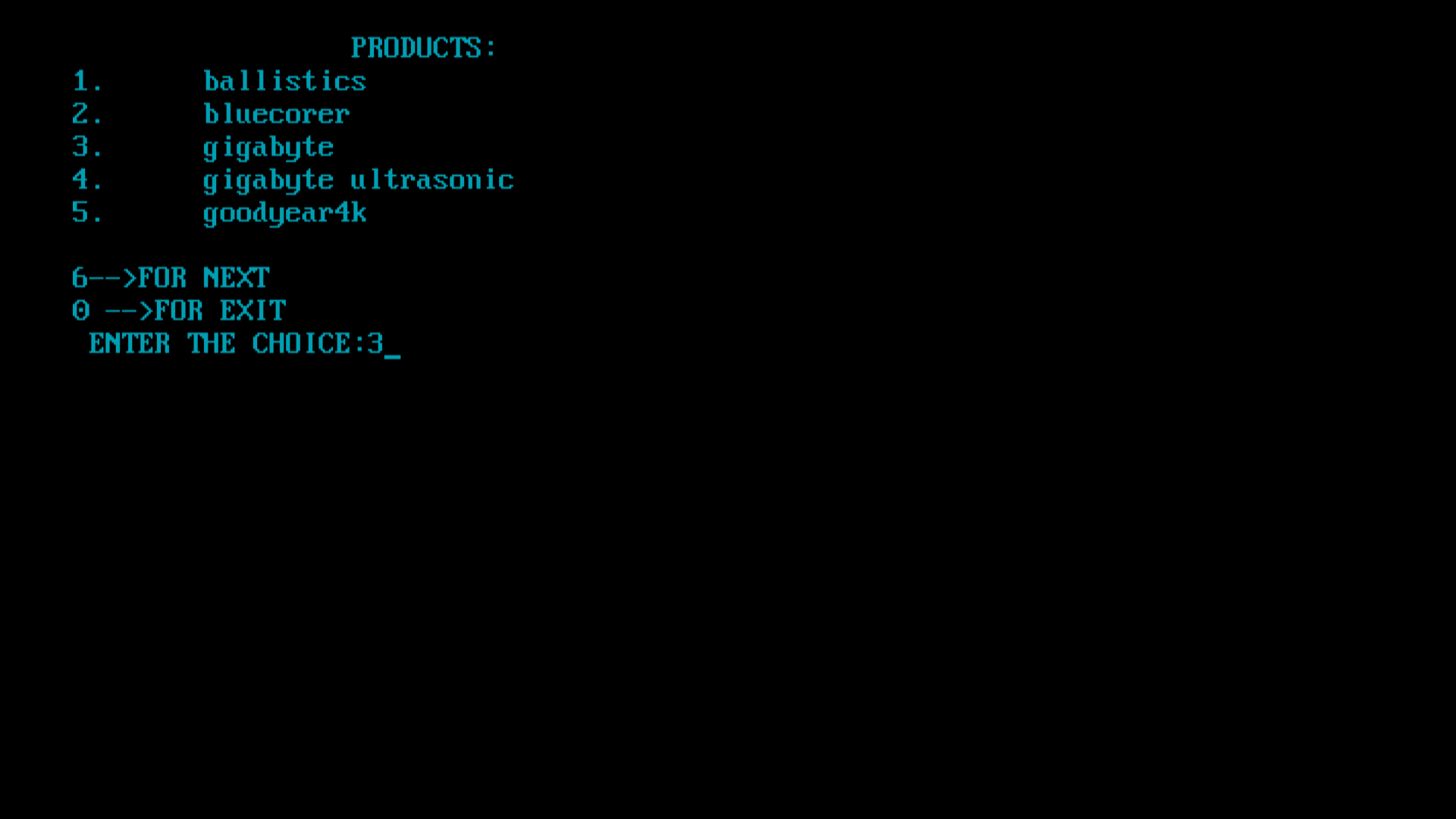


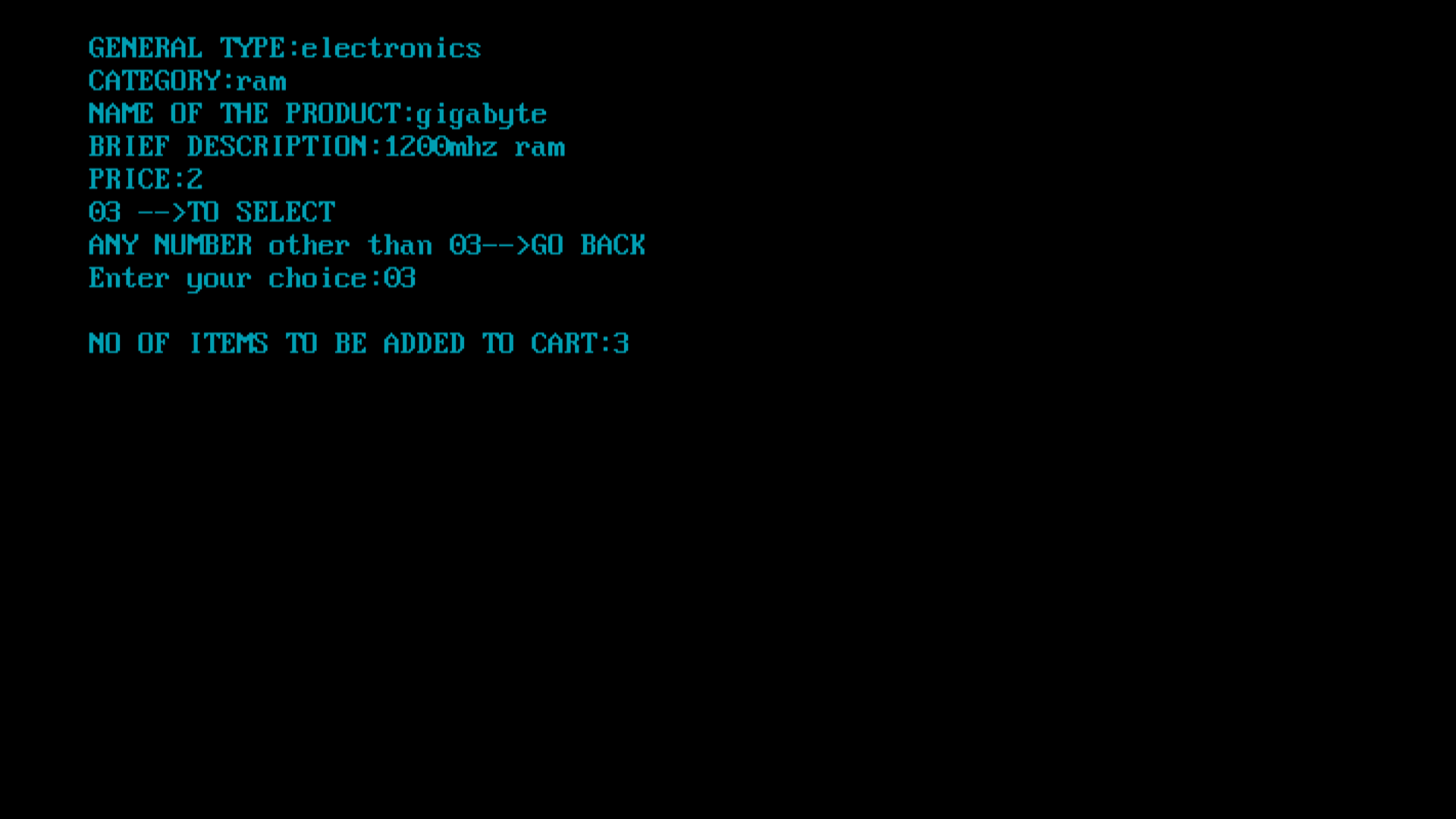


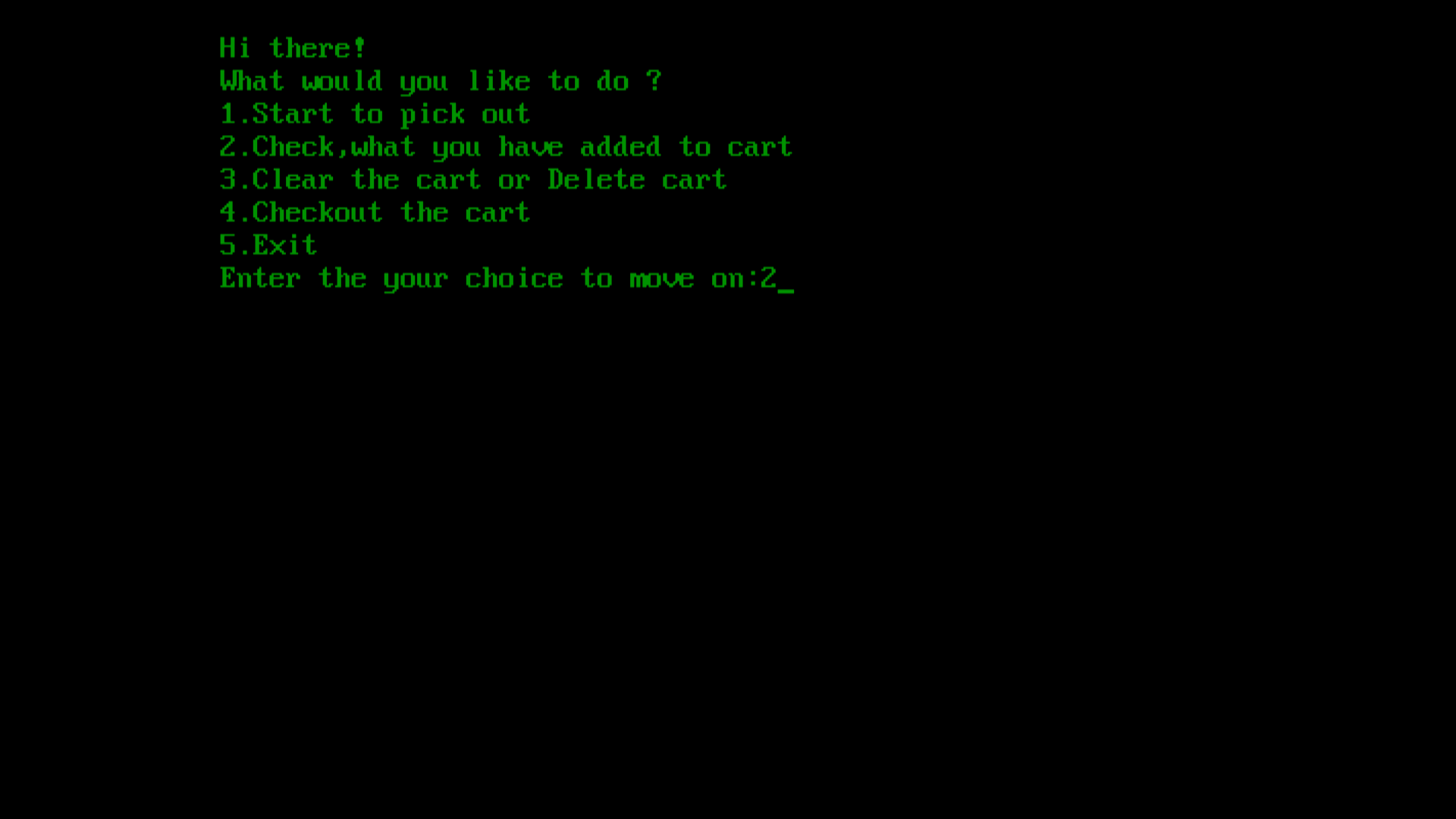


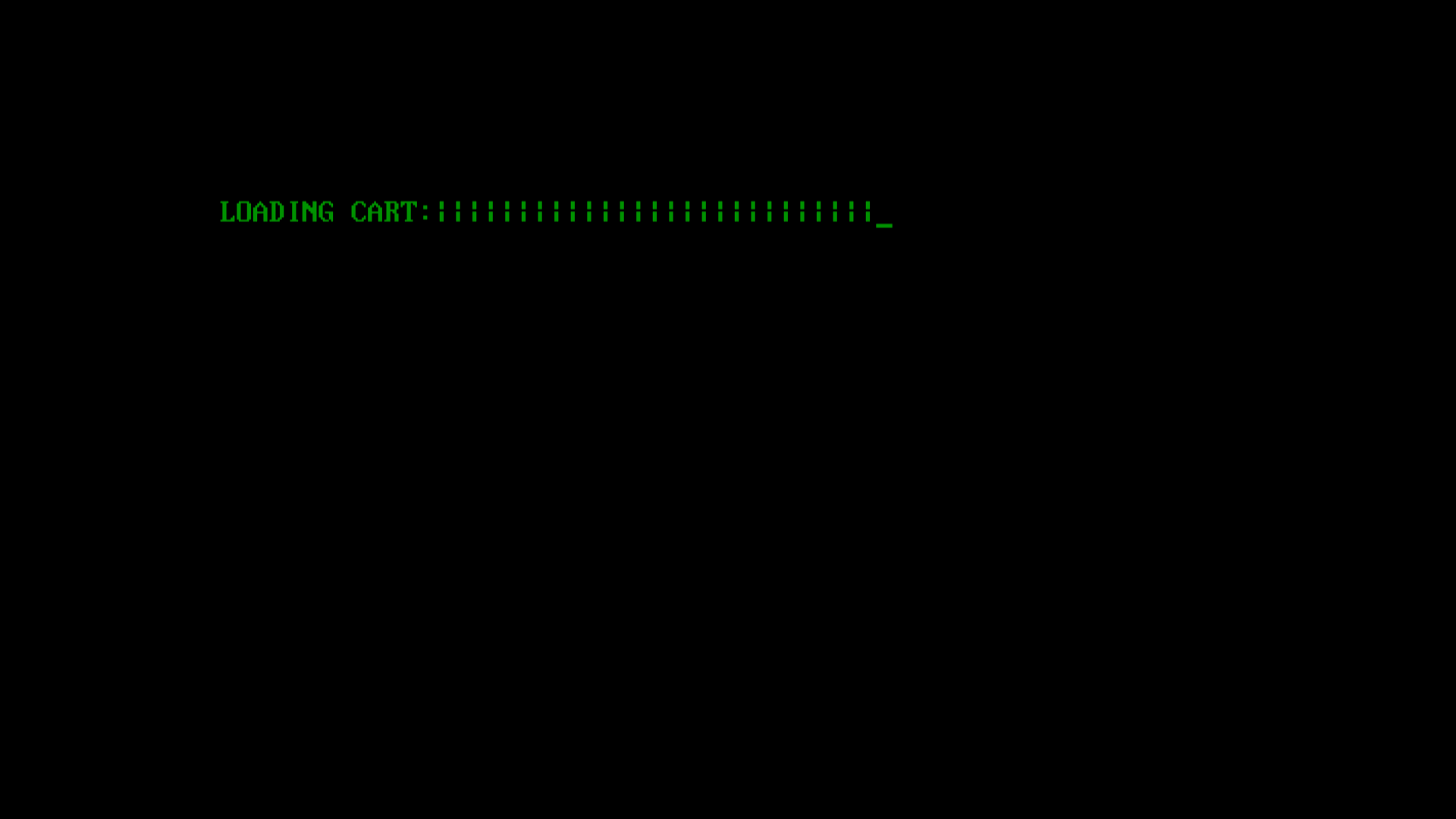


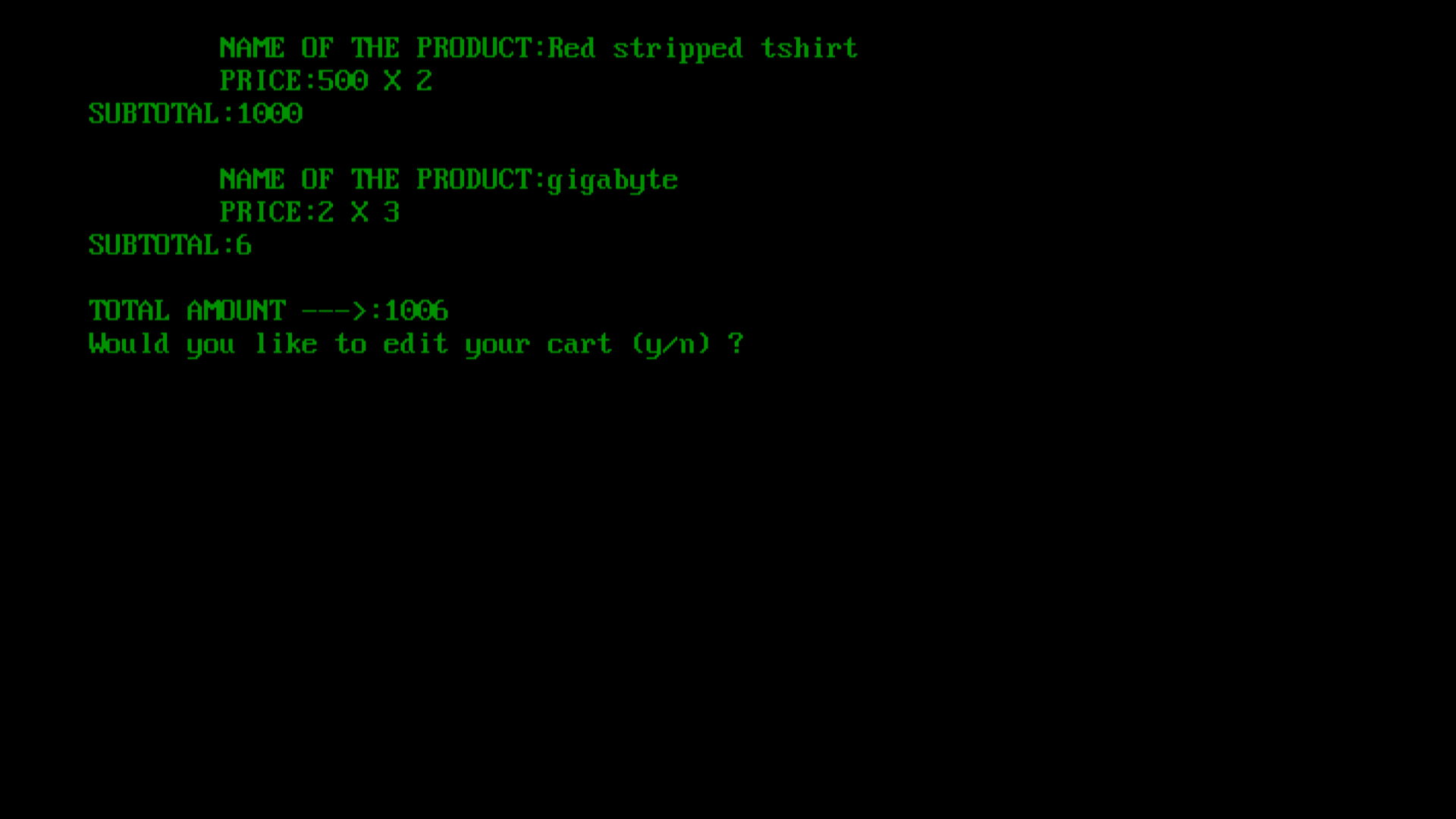


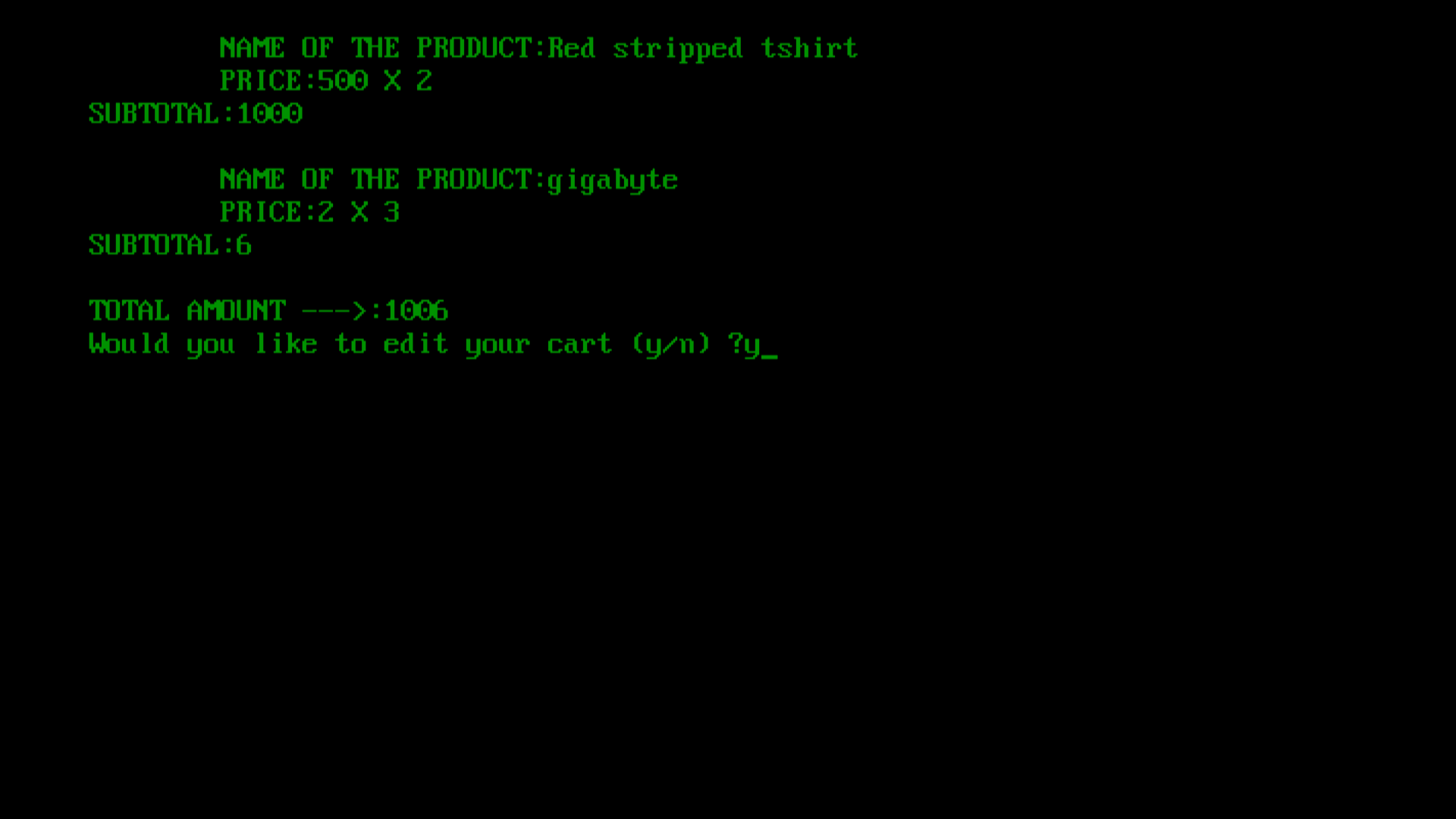


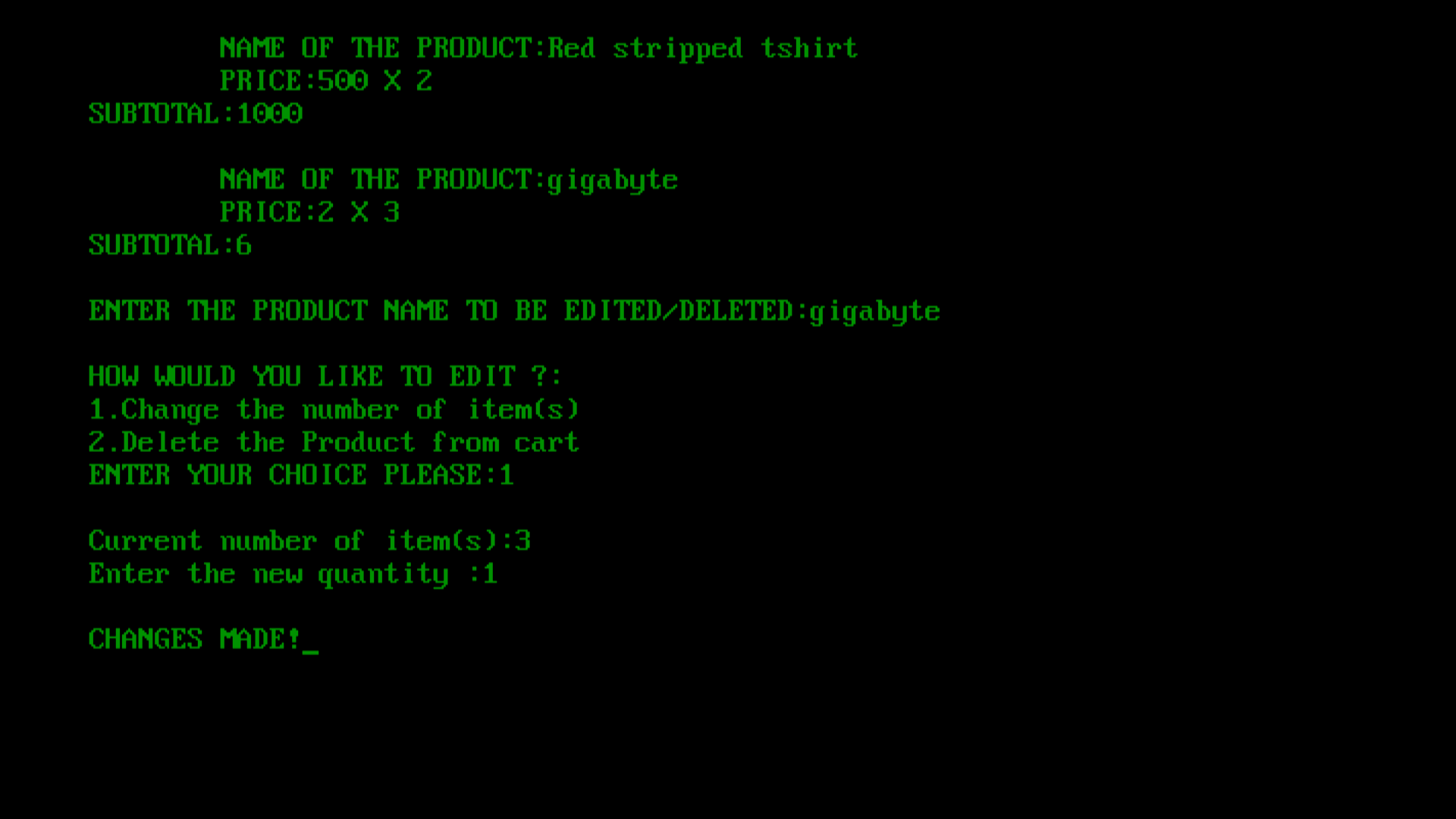


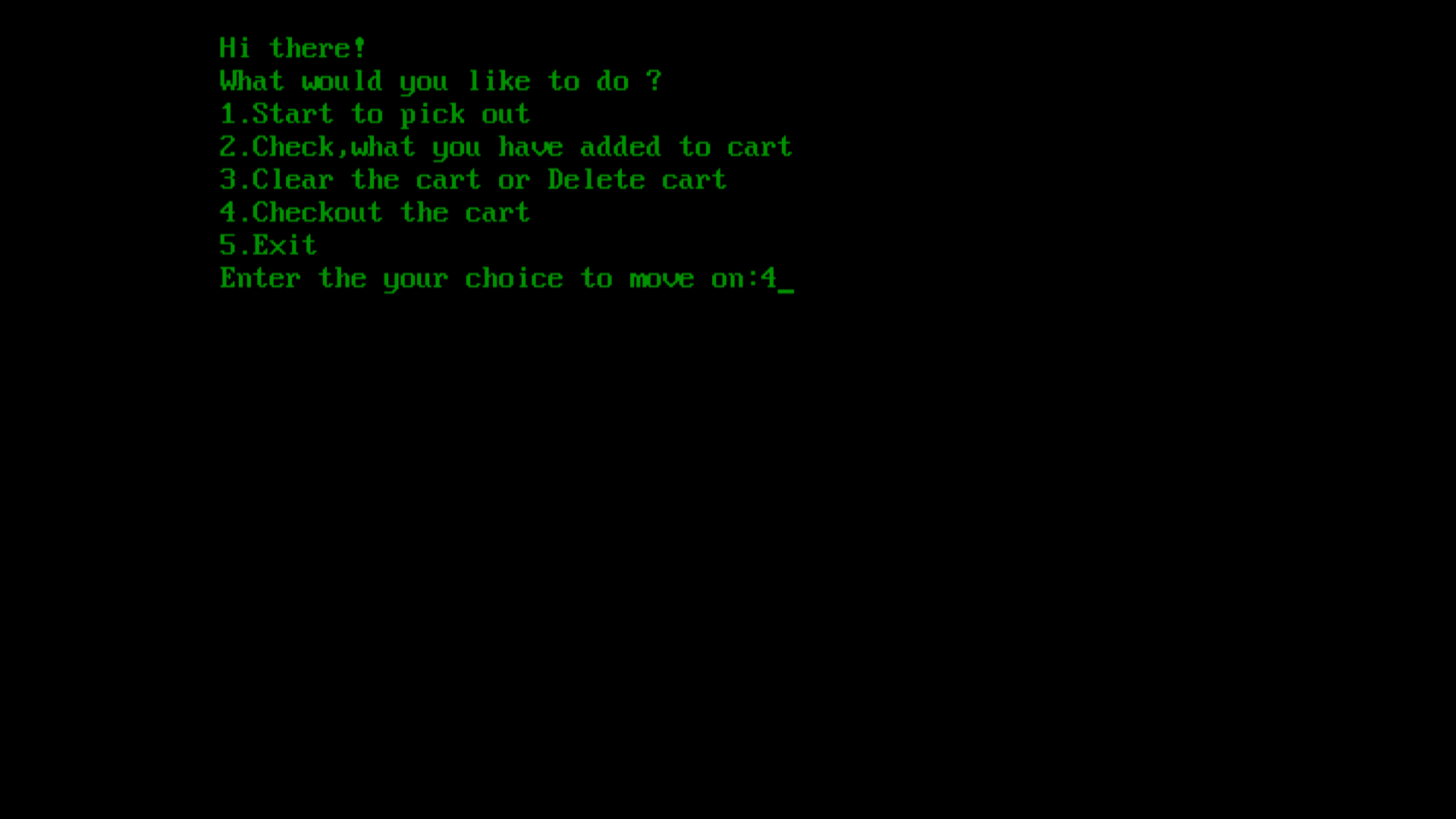


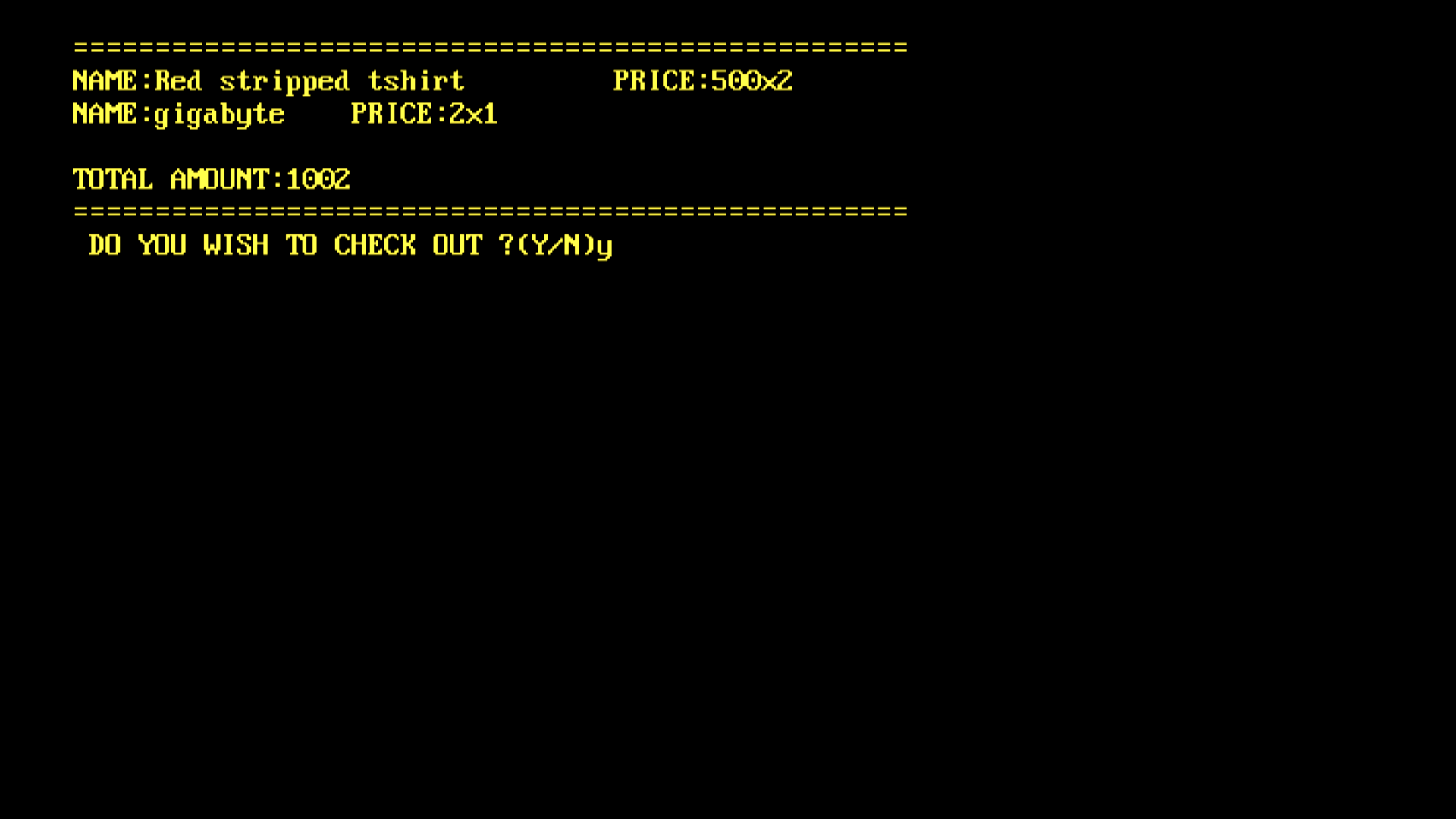




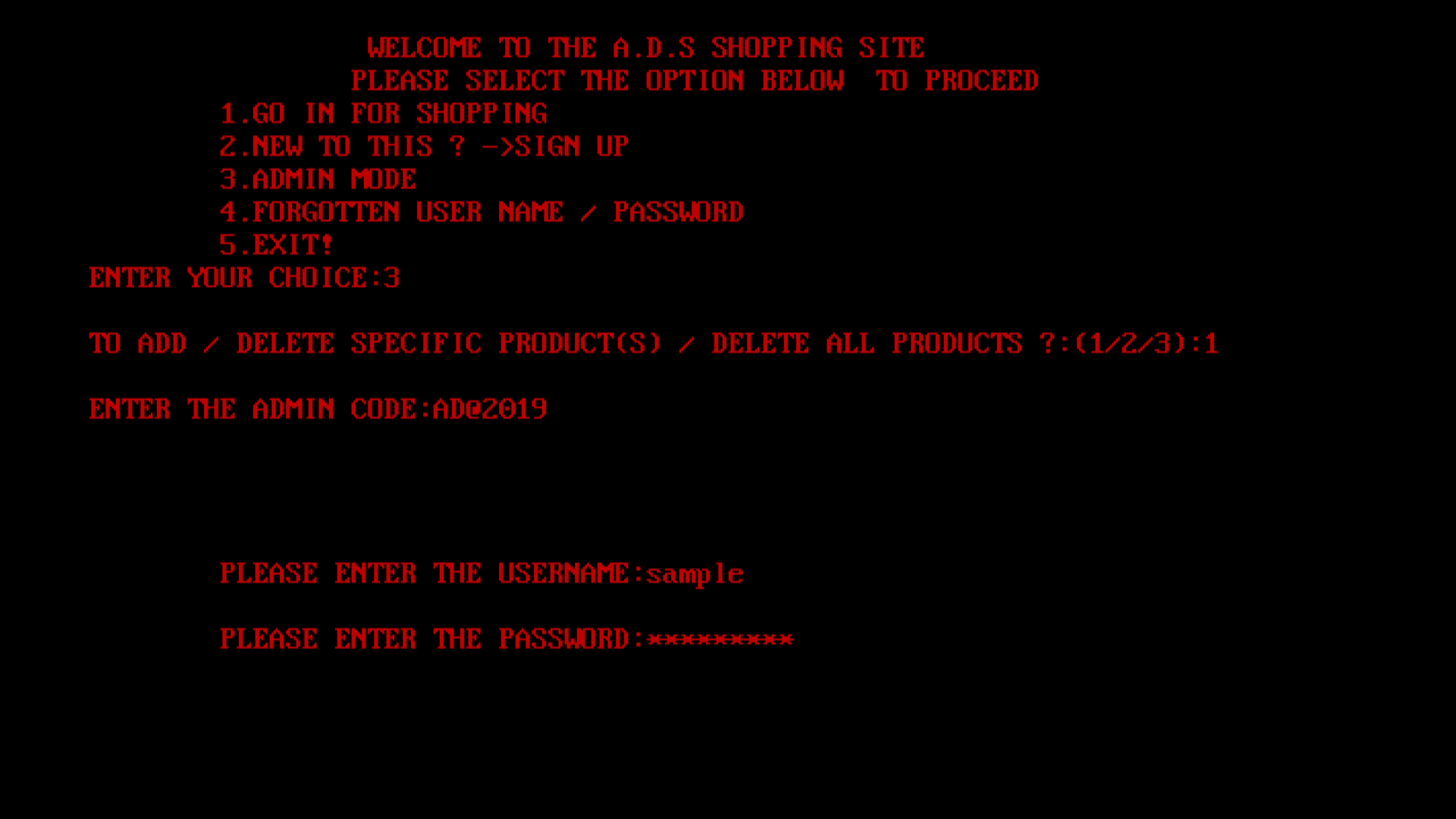


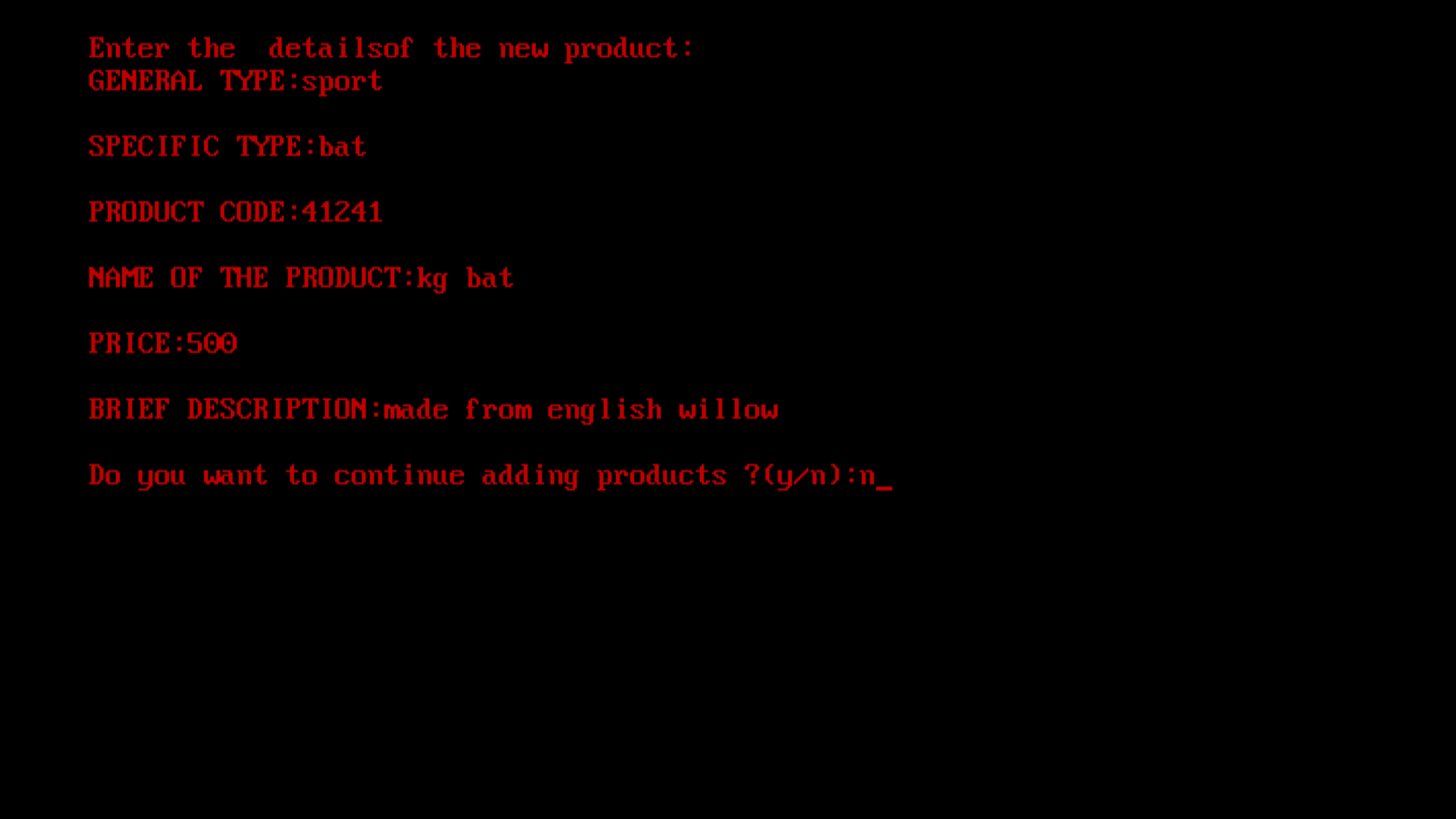


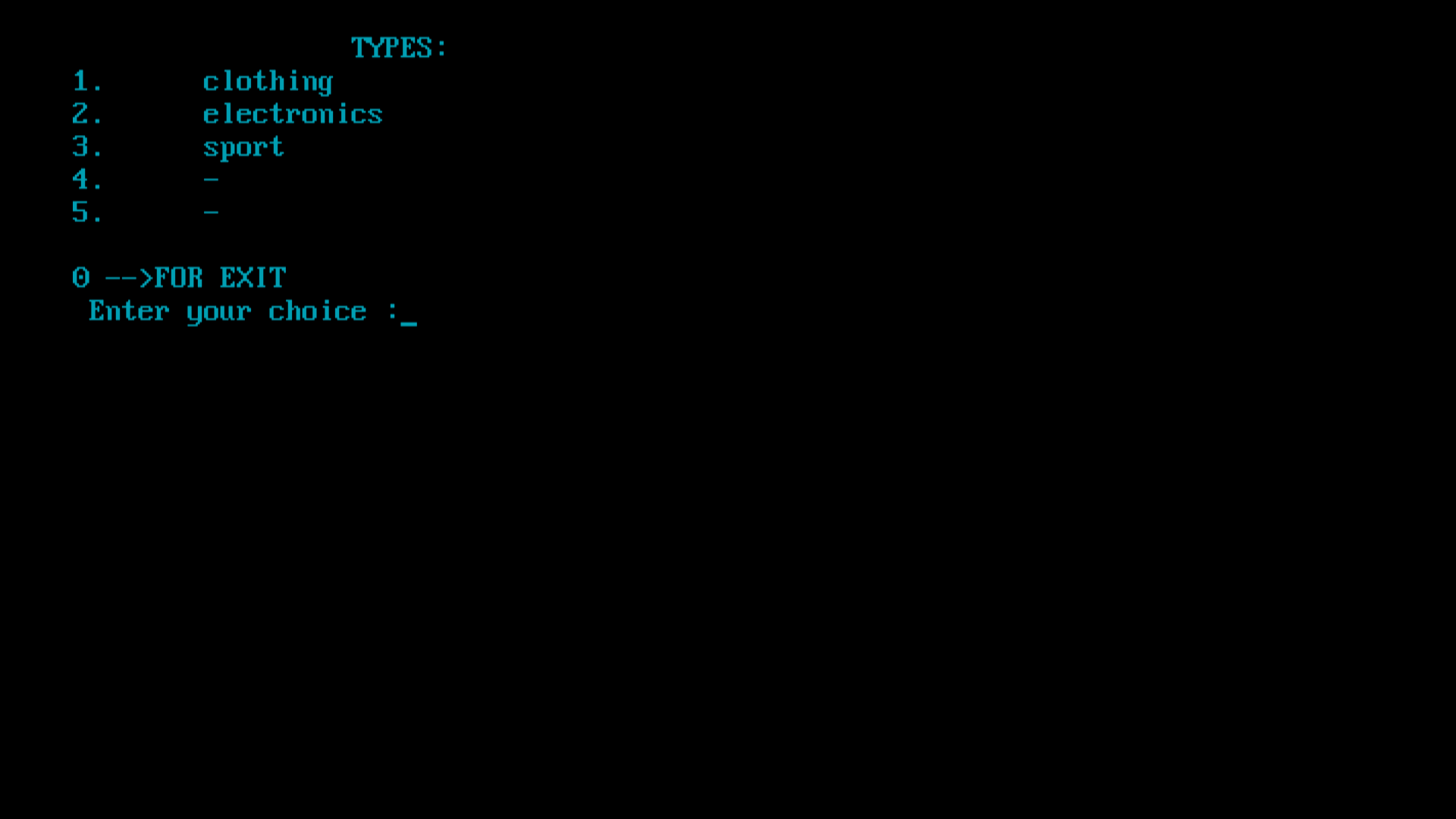


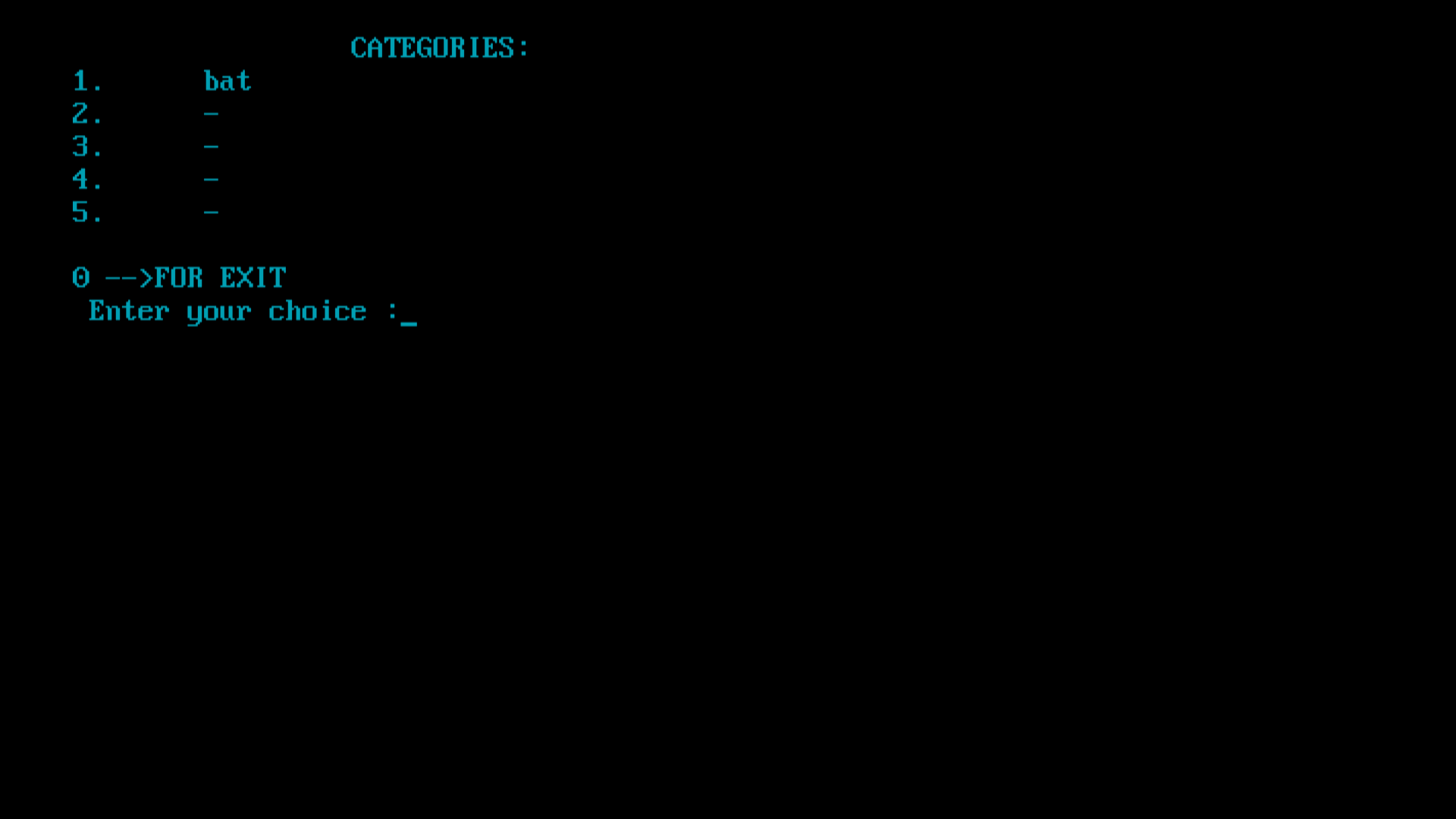


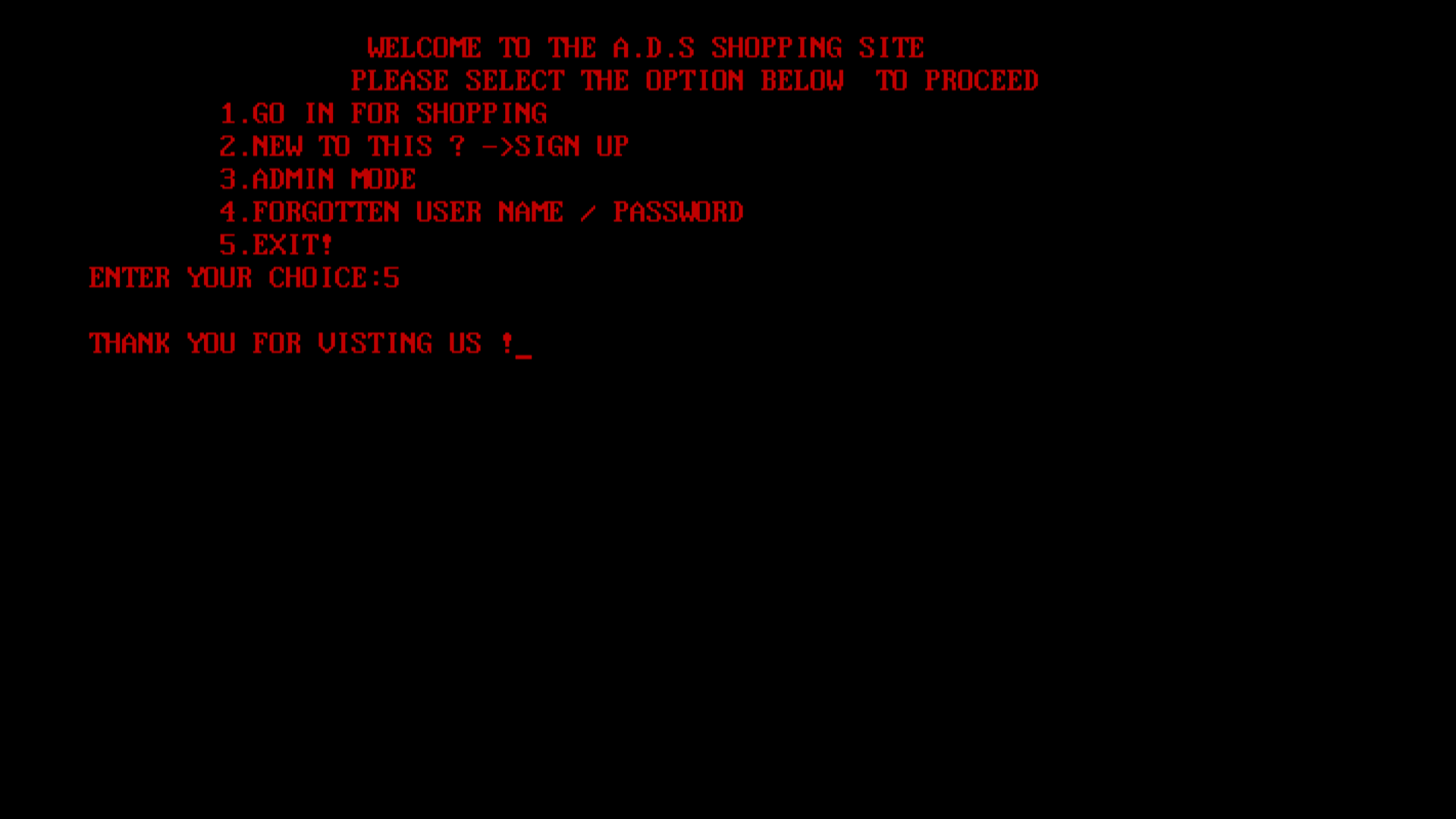












 FUTURE ENHANCEMENTS

* “THIS APPLICATION” is just a prototype of the actual full version of the ONLINE SELF SHOPPING . The further updates includes more features like enhanced user interface and smart selecting of products based on the users searches and needs .
* The next version of this program can be used by the regular people to self checkout their products ,when they shop at malls.
* The future updates enables the user to store their payment details and many more details , which can be used for ease of payment.

REFERENCE

* Computer science C++ for class XI book by SUMITA ARORA.