****

**INF212**

**ALGORITHMS AND PROGRAMMING II**

**PROJECT-1**

**Sale Program**

**Deadline is April 1, 2021 at 23:00.**

**Projects that are not delivered on time are not accepted.**

**Upload the project to the Project 1 assignment section of the INF212 class.**

**The questions can be asked to course lecturer Dr. Tuba GÖZEL and teaching assistant Süleyman TUNCEL and Mehmet Türker TAKCI.**

**Complete your project, fill the report by using attached word file   
(INF212\_Project\_1\_Report\_Template.docx).**

**Upload the project codes and the report.**

**You can compress the project codes if number of the files is more than ten.**

**The file name of report should be as *P1\_ID\_FirstLetterofName\_.docx*   
(i.e. *P1\_141024001\_YKZ.docx* ).**

****

**INF 212 Algorithms and Programming II**

**2020-2021 Spring**

**Electronic Engineering**

**Proje ID# Project Name**

**\_Program Name\_**

|  |  |  |
| --- | --- | --- |
| **School ID** | **Name** | **Surname** |
| **200102002043** | **Senanur** | **Ağaç** |

|  |
| --- |
| PROJECT OBJECTIVE  Briefly describe the objectives of your project |
| Projemizde amaç bir satış otomasyon sistemi kurmaktır. Projede satılan ürünlerin, müşterilerin belirli bilgileri bağlı listeler özelliği ile tutulmaktadır. Bu bilgilere göre menü kısmından yapılacak işlem seçilir ve bilgiler ekrana yazdırılır. Bu sistem iş yükünü azaltıp ve bilgilerin bir arada tutulmasını sağlar ve bu bilgilere göre analiz yaparak bu meslek grubuna yardımcı olur. |
| PROBLEM  Briefly describe the problem you given to solve |
| Bir satış otomasyon sistemi kurarak verilerin elde tutulmasını kolaylaştırmak ve analiz yaparak kullanıcıya yardımcı olduk. |
| ANALYSIS  Analyze of the problem |
| * Öncelikle sistemimizde menü kısmıyla kullanıcıdan işlem seçmesini istiyoruz. * (1-Müşteri bilgisi 2-Ürün Bilgisi 3-Müsteri Analizleri 4-Ürün Analizleri 5-Çıkış) * Daha sonra kullanıcı 1-Müşteri bilgisi seçeneğini seçtiği takdirde müşteri bilgisini istediği seçenekleri sunuyoruz.Bunlar 1.1 Tüm müşteri bilgileri-1.2Müşteri tipine göre(kurumsal veya bireysel) müşteri bilgileri-1.3 Bir müşteri bilgisi * İstediği seçeneği seçtiğinde buna göre müşteri bilgileri ekrana yazdırılır. * Eğer 2- Ürün bilgisi seçeneğini seçerse ürün bilgisini istediği seçenekleri sunuyoruz. Bunlar 2.1-Tüm ürün bilgileri 2.2- Ürün tipine göre ürün bilgileri 2.3-Bir ürünün bilgileri * İstediği seçeneği seçtiğinde buna göre ürün bilgileri ekrana yazdırılır. * Eğer 3- Müşteri Analizleri seçeneğini seçerse müşteri analizleri için istediği seçeneği soruyoruz. Bunlar 3.1- Bir müşterinin satın aldığı ürünler 3.2-Bir müşterinin satın aldığı toplam tutar 3.3- Tüm müşterilerin satın aldığı ürünlerin toplam tutarı 3.4-Müşterilerin kargo ücreti * İstediği seçeneği seçtiğinde buna göre müşteri analizi ekrana bastırılır. * Eğer 4- Ürün Analizleri seçeneğini seçerse ürün analizi için istediği seçeneği soruyoruz. Bunlar 4.1-Bir ürünün toplam satış tutarı 4.2-Bir müşterinin satın aldığı toplam tutar 4.3-Tüm müşterilerin satın aldığı ürünlerin toplam tutarı 4.4-Müşterilerin kargo ücreti * İstediği seçeneği seçtiğinde buna göre ürün analizi ekrana bastırılır. * Eğer 5 seçeneğini seçerse bir işlem yapmadan sistemden çıkılır. |
| DATA REQUIREMENTS  Specify all data requirements of the problem |
| **Structured Data Type** |
| struct musterinode{  int ID;  char name[50];  int tip ;  double x;  double y;  struct musteri \*sonraki;  };  struct urunbilgisi{  int urunID;  char urunname[50];  int uruntip;  double urunucret;  struct urun \*sonrakiurun;  };  struct parcabilgisi{  int ID;  int invoice\_ID;  int customer\_ID;  int product\_ID;  double cost;  struct Purchased \*nextPurchasedPtr;  }; |
| **Problem Constants** |
| Kullanmadım |
| **Problem Inputs** |
| * Müşteri adı-ID numarası-X koordinatı-Y koordinatı-Sonraki müşterinin bellek adresi * Ürün adı-Ürün numarası(ID)-Ürün tipi()-Ürün fiyatı-Sonraki ürünün bellek adresi * Parça numarası-Fatura numarası-Satış yapılan müşteri numarası-Satılan ürün tutarı-Bir sonraki satış yapılan ürünün bellek adresi * Genel menü işlem seçimi * Ara menü işlem seçimi |
| **Problem Output** |
| * Satış menüsü * Müşteri bilgileri * Ürün bilgileri * Müşterinin satın aldığı ürünler * Müşterinin ödediği toplam ücret * Tüm müşterilerin ödediği toplam ücret * Müşterilerin kargo ücreti * Ürünün satış tutarı * Ürün tipinin satış tutarı * Tüm ürünlerin satış tutarı |
| DESIGN  ALGORITHM  Design algorithm of the problem |
| * Menüden istediği bilginin işlemini seçmesi(Müşteri bilgisi-Ürün bilgisi-Müşteri analizi-Ürün analizi-Çıkış) * **Eğer** müşteri bilgisi seçerse;   1. Tüm müşteriler   1.2 Müşteri tipine göre  1.2 Bir müşteri seçenekleri sunulur.   * + Eğer Tüm Müşterileri seçerse tüm müşterilerin isim-soyisim , ID numaraları,coğrafi bilgi sistemindeki x ve y koordinat bilgileri ekrana basılır   + Eğer Müşteri tipine göre seçeneğini seçerse öncelikle (0-Bireysel 1-Kurumsal) Seçtiği müşteri tipine göre müşterilerin isim-soyisim ,ID numaraları , coğrafi bilgi sistemindeki x ve y koordinat bilgileri ekrana basılır.   + Eğer Bir müşteri seçeneğini seçerse ID numarasını girdiği müşterinin isim-soyisim, ID numaraları ,coğrafi bilgi sistemindeki x ve y koordinat bilgileri ekrana basılır.   + **Eğer** Ürün Bilgisi seçerse ;   2.1 Tüm ürünler  2.2 Ürün tipine göre  2.3 Bir ürün seçenekleri sunulur.   * + Eğer Tüm ürünler seçeneğini seçerse tüm ürünlerin isim, ID numarası, Fiyat, Tip gibi özellikleri ekrana bastırılır.   + Eğer Ürün tipine göre seçeneğini seçerse öncelikle (0-Et ürünleri 1- Sebze 2-Meyve 3-İçecek) seçtiği ürün tipine göre ürünlerin isim, ID numarası, Fiyat, Tip gibi özellikleri ekrana bastırılır.   + Eğer bir ürün seçeneğini seçerse ID numarasını girdiği ürünün isim, ID numarası, Fiyat, Tip gibi özellikleri ekrana bastırılır.   + **Eğer** Müşteri Analizleri seçeneğini seçerse ;   3.1 Bir müşterinin satın aldığı ürünler  3.2 Bir müşterinin satın aldığı toplam tutar  3.3 Tüm müşterinin satın aldığı ürünlerin toplam tutarı  3.4 Müşterilerin kargo ücreti seçenekleri sunulur.   * + Eğer bir müşterinin satın aldığı ürünler seçeneğini seçerse önce istediği müşterinin ID numarası sorulur ve müşterinin satın aldığı ürünün ID numarası ekrana basılır.   + Eğer Bir müşterinin satın aldığı toplam tutar seçeneğini seçerse önce istediği müşterinin ID numarası sorulur ve müşterinin aldığı tutar ekrana bastırılır.   + Eğer Tüm müşterinin satın aldığı ürünlerin toplam tutarı seçerse tüm müşterilerin aldığı ürünlerin tutarı önce tek tek ekrana basılır ve sonra toplamı verilir.   + Eğer Müşterilerin kargo ücreti seçeneğini seçerse önce tüm müşterilerin x ve y koordinat bilgilerine göre mesafesi hesaplanır ve km başına 5 tl ile kargo ücretleri tek tek hesaplanıp ekrana basılır.   + **Eğer** Ürün Analizleri seçeneğini seçerse ;   4.1 Bir ürünün toplam satış tutarı  4.2 Bir ürün tipinin toplam satış tutarı  4.3 Tüm ürünlerin toplam satış tutarı seçenekleri sunulur.   * + Eğer Bir ürünün toplam satış tutarı seçeneğini seçerse ID numarasını girdiği ürünün toplam satış tutarı ekrana basılır.   + Eğer Bir ürün tipinin toplam satış tutarı seçeneğini seçerse önce ürün tipi sorulur. Seçtiği ürün tipindeki ürünlerin satış tutarları ekrana bastırılır.   + Eğer Tüm ürünlerin toplam satış tutarı seçeneğini seçerse tüm ürünlerin satış tutarı bilgileri ekrana bastırılır.   + Eğer Çıkış seçeneğini seçerse sistemden çıkılır. |
| HEADER FILE(S) |
| #ifndef senaprojee  #define senaprojee  struct musterinode{  int ID;  char name[50];  int tip ;  double x;  double y;  struct musteri \*sonraki;  };  struct urunbilgisi{  int urunID;  char urunname[50];  int uruntip;  double urunucret;  struct urun \*sonrakiurun;  };  struct parcabilgisi{  int ID;  int invoice\_ID;  int customer\_ID;  int product\_ID;  double cost;  struct Purchased \*nextPurchasedPtr;  };    void instructions( void );  void tumusteribilgileri(struct musterinode \*a);  void tipmusteri(struct musterinode \*b);  void birmusteri(struct musterinode \*c);  void tumurunbilgileri(struct urunbilgisi \*a);  void tipurun(struct urunbilgisi \*b);  void birurun(struct urunbilgisi \*c);  void birmusteriurun(struct parcabilgisi \*urun);  void birmusteritutar(struct parcabilgisi \*tutar);  void tummusteritutar(struct parcabilgisi \*toptutar);  void kargoucreti(struct musterinode \*kargo);  void urunsatis(struct parcabilgisi \*urunsatis);  void tipsatis(struct urunbilgisi \*tipsatis,struct parcabilgisi \*tut);  void tumsatis (struct parcabilgisi \*tum);  #endif |
| IMPLEMENTATION  Explain the method applied for the problem solution |
| Öncelikle menümüz için swich-case yapısını oluşturdum. Bu yapının içinde de seçenek seçmek için birden fazla if-else kullandım. Fonksiyonlarda while içinde if else yapısı kullanarak işlemleri gerçekleştirdim. Ürünlerin-Müşterilerin bilgilerini ise 3 farlı struct yapısı ile (bağlı liste ile) tuttum. |
| TESTING  Briefly describe how you test your code |
| Doğru sonuç alınması için ürünlerin fiyatı TL cinsinden alınmalıdır.  X ve Y koordinatlarına göre mesafe hesaplaması KM cinsinden kabul edilecektir. |
| USER’S GUIDE  (Describe briefly how the user can use your program (input parameters of program, output of program, possible limitations, possible errors, etc.) |
| Öncelikle sistemimizde menü kısmıyla kullanıcıdan işlem seçmesini istiyoruz.  (1-Müşteri bilgisi 2-Ürün Bilgisi 3-Müsteri Analizleri 4-Ürün Analizleri 5-Çıkış)  Daha sonra kullanıcı 1-Müşteri bilgisi seçeneğini seçtiği takdirde müşteri bilgisini istediği seçenekleri sunuyoruz.Bunlar 1.1 Tüm müşteri bilgileri-1.2Müşteri tipine göre(kurumsal veya bireysel) müşteri bilgileri-1.3 Bir müşteri bilgisi  İstediği seçeneği seçtiğinde buna göre müşteri bilgileri ekrana yazdırılır.  Eğer 2- Ürün bilgisi seçeneğini seçerse ürün bilgisini istediği seçenekleri sunuyoruz. Bunlar 2.1-Tüm ürün bilgileri 2.2- Ürün tipine göre ürün bilgileri 2.3-Bir ürünün bilgileri  İstediği seçeneği seçtiğinde buna göre ürün bilgileri ekrana yazdırılır.  Eğer 3- Müşteri Analizleri seçeneğini seçerse müşteri analizleri için istediği seçeneği soruyoruz. Bunlar 3.1- Bir müşterinin satın aldığı ürünler 3.2-Bir müşterinin satın aldığı toplam tutar 3.3- Tüm müşterilerin satın aldığı ürünlerin toplam tutarı 3.4-Müşterilerin kargo ücreti  İstediği seçeneği seçtiğinde buna göre müşteri analizi ekrana bastırılır.  Eğer 4- Ürün Analizleri seçeneğini seçerse ürün analizi için istediği seçeneği soruyoruz. Bunlar 4.1-Bir ürünün toplam satış tutarı 4.2-Bir müşterinin satın aldığı toplam tutar 4.3-Tüm müşterilerin satın aldığı ürünlerin toplam tutarı 4.4-Müşterilerin kargo ücreti  İstediği seçeneği seçtiğinde buna göre ürün analizi ekrana bastırılır.  Eğer 5 seçeneğini seçerse bir işlem yapmadan sistemden çıkılır. |
| GENERAL FLOWCHART OF THE PROJECT |
|  |
| CONCLUSION AND REMARKS  You can or can(not) put any remarks on your work. For example;  is your program works or not, if not why  which troubles during performance of project you faced with and how you overcome them;  is your program user friendly and how you can improve it;  what you achieved during performance of project; and so |
| Projeyi yaparken çok fazla zorlandım. Birden çok kaynaktan ve videolardan yararlanmama rağmen insert fonksiyonunu bir türlü yapamadım. Aşağıya hata veren insert fonksiyon kodlarımı koyacağım. Hata verdiği için projeye eklemedim. Onun dışında projede sadece yapabildiğim kadar kısım var malsef bazı kısımlrını anlayamadım. Açıkcası Linked List konusunda lab yapılmadığı için konu tam olarak oturmadı ve ek kaynaklara göre yapmaya çalıştım.  İNSERT --   1. struct bagliListe\* dugumOlustur(int ID,int tip,double x,double y){   struct bagliListe\* yeniDugum= (struct bagliListe\*)malloc(sizeof(struct bagliListe\*));  yeniDugum->ID=ID;  yeniDugum->tip=tip;  yeniDugum->x=x;  yeniDugum->y=y;  yeniDugum->sonrakiDugum=NULL;  return yeniDugum;  }  2-void insertCustomer( CustomerPtr \*cPtr, int c\_ID, char\* c\_name, int c\_type, double c\_x, double c\_y) {  CustomerPtr besinci = (Customer \*)malloc(sizeof (Customer));  besinci -> ID = c\_ID;  strcpy (besinci->name,c\_name);  besinci -> type = c\_type;  besinci -> x\_coord = c\_x;  besinci -> y\_coord = c\_y;  besinci -> nextCusPtr = NULL;    if (startPtr == NULL) {  \*startPtr=besinci;  }    else {  q = startPtr;    while(q->(\*(nextCusPtr)) != NULL) {  q = q->(\*(nextCusPtr)) ;    }  q -> (\*(nextCusPtr)) = besinci;  }  }  3- |
| REFERENCES  Put the list of references and sources (books, links to websites, videos, etc), which you used for project. |
| <https://www.youtube.com/watch?v=r3uOBb3BM-0>  <https://www.bilgigunlugum.net/prog/cprog/c_yapi>  <https://www.geeksforgeeks.org/linked-list-set-1-introduction/>  <https://meminoglu.medium.com/tek-y%C3%B6nl%C3%BC-ba%C4%9Fl%C4%B1-listeler-hakk%C4%B1nda-her-%C5%9Fey-1-2-a7b0ab95a0a0>  <https://stackoverflow.com/>  <https://youtube.com/playlist?list=PLUUSh8du1azZEsdngPxMQGLnatId0V4h9> |

|  |
| --- |
| APPENDIX |
| PROJECT CODES  Put here code of your project. |
| MAİN.C  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <math.h>  #include "senaprojee.h"  /\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/  int main(int argc, char \*argv[]) {  printf("SISTEME HOSGELDINIZ\n");  struct musterinode \*birinci=(struct musterinode \*)malloc(sizeof(struct musterinode));  struct musterinode \*ikinci=(struct musterinode \*)malloc(sizeof(struct musterinode));  struct musterinode \*ucuncu=(struct musterinode \*)malloc(sizeof(struct musterinode));  struct musterinode \*dorduncu=(struct musterinode \*)malloc(sizeof(struct musterinode));  struct musterinode \*besinci=(struct musterinode \*)malloc(sizeof(struct musterinode));  birinci->ID = 1;  strcpy (birinci->name,"Senanur Agac");  birinci->tip=1;  birinci->x=3.0;  birinci->y=4.0;  birinci->sonraki= ikinci;    ikinci->ID =2;  strcpy(ikinci->name,"Ervanur Cetinkol");  ikinci->tip=0;  ikinci->x=7.0;  ikinci->y=24.0;  ikinci->sonraki = ucuncu;    ucuncu->ID=3;  strcpy(ucuncu->name,"Omer Asaf Sunger");  ucuncu->tip=0;  ucuncu->x=6.0;  ucuncu->y=8.0;  ucuncu->sonraki= dorduncu;    dorduncu->ID=4;  strcpy(dorduncu->name,"Beyza Cil");  dorduncu->tip=1;  dorduncu->x=12.0;  dorduncu->y=16.0;  dorduncu->sonraki = besinci;    besinci->ID=5;  strcpy(besinci->name,"Kubra Yildiz");  besinci->tip=1;  besinci->x=9.0;  besinci->y=12.0;  besinci->sonraki = NULL;  struct urunbilgisi \*birinciurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*ikinciurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*ucuncuurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*dorduncuurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*besinciurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*altinciurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  struct urunbilgisi \*yedinciurun=(struct urunbilgisi \*)malloc(sizeof(struct urunbilgisi \*));  /\*  id numaraları=  et ürünleri =0 sebze=1 meyve =3 içecek=4    \*/ birinciurun->urunID=1 ;  strcpy(birinciurun->urunname,"Balik");  birinciurun->uruntip=0;  birinciurun->urunucret=3.0;  birinciurun->sonrakiurun=ikinciurun;  ikinciurun->urunID=2;  strcpy(ikinciurun->urunname,"Domates");  ikinciurun->uruntip=1;  ikinciurun->urunucret=2.5;  ikinciurun->sonrakiurun=ucuncuurun;    ucuncuurun->urunID=3;  strcpy(ucuncuurun->urunname,"Cilek");  ucuncuurun->uruntip=2;  ucuncuurun->urunucret=5.0;  ucuncuurun->sonrakiurun=dorduncuurun;  dorduncuurun->urunID=4;  strcpy(dorduncuurun->urunname,"Ayran");  dorduncuurun->uruntip=3;  dorduncuurun->urunucret=3.0;  dorduncuurun->sonrakiurun=besinciurun;  besinciurun->urunID=5;  strcpy(besinciurun->urunname,"Tavuk");  besinciurun->uruntip=0;  besinciurun->urunucret=25.0;  besinciurun->sonrakiurun=altinciurun;    altinciurun->urunID=6;  strcpy(altinciurun->urunname,"Patlican");  altinciurun->uruntip=1;  altinciurun->urunucret=8.0;  altinciurun->sonrakiurun=yedinciurun;    yedinciurun->urunID=7;  strcpy(yedinciurun->urunname,"Soda");  yedinciurun->uruntip=3;  yedinciurun->urunucret=2.0;  yedinciurun->sonrakiurun=NULL;    struct parcabilgisi \*p1=(struct parcabilgisi \*)malloc(sizeof(struct parcabilgisi \*));  struct parcabilgisi \*p2=(struct parcabilgisi \*)malloc(sizeof(struct parcabilgisi \*));  struct parcabilgisi \*p3=(struct parcabilgisi \*)malloc(sizeof(struct parcabilgisi \*));  struct parcabilgisi \*p4=(struct parcabilgisi \*)malloc(sizeof(struct parcabilgisi \*));  struct parcabilgisi \*p5=(struct parcabilgisi \*)malloc(sizeof(struct parcabilgisi \*));  p1->ID=1;  p1->invoice\_ID=1;  p1->customer\_ID=2;  p1->product\_ID=3;  p1->cost=25.0;  p1->nextPurchasedPtr=p2;  p2->ID=2;  p2->invoice\_ID=1;  p2->customer\_ID=3;  p2->product\_ID=5;  p2->cost=150.0;  p2->nextPurchasedPtr=p3;  p3->ID=3;  p3->invoice\_ID=2;  p3->customer\_ID=4;  p3->product\_ID=7;  p3->cost=18;  p3->nextPurchasedPtr=p4;  p4->ID=4;  p4->invoice\_ID=3;  p4->customer\_ID=1;  p4->product\_ID=4;  p4->cost=24;  p4->nextPurchasedPtr=p5;  p5->ID=5;  p5->invoice\_ID=5;  p5->customer\_ID=5;;  p5->product\_ID=6;  p5->cost=48;  p5->nextPurchasedPtr = NULL;  int choice;  instructions();  printf("\nYukarida verilen tabloya gore yapmak istediginiz islemi seciniz : ");  scanf( "%d", &choice );  int a,b,t,f;  switch (choice){  case 1:  printf("--Musteri Bilgileri--\n");  printf("1-Tum musteriler\n2-Musteri tipine gore \n3-Bir musteri\nIslem seciniz--> ");  scanf("%d",&a);  if(a==1){  tumusteribilgileri(birinci);  }  else if(a==2){  tipmusteri(birinci);  }  else if(a==3){  birmusteri(birinci);  }  else {  printf("Yanlis secim!");  }  break;  case 2:  printf("--Urun Bilgileri--\n");  printf("1-Tum urunler\n2-Urun tipine gore\n3-Bir urun\nIslem seciniz-->");  scanf("%d",&b);  if(b==1){  tumurunbilgileri(birinciurun);  }  else if(b==2){  tipurun(birinciurun);  }  else if(b==3){  birurun(birinciurun);  }  else {  printf("Yanlis secim!");  }  break;    case 3:  printf("-- Müşteri Analizleri--\n");  printf("Musteri Analizleri\n1-Bir musterinin satin aldigi urunler\n2-Bir musterinin satin aldigi toplam tutar\n3.Tum musterinin satin aldigi urunlerin toplam tutari\n4-Musterilerin kargo ucreti");  printf("\nIslem seciniz -->");  scanf("%d",&t);  if(t==1){  birmusteriurun(p1);  }  else if(t==2) {  birmusteritutar(p1);  }  else if(t==3){  tummusteritutar(p1);  }  else if(t==4){  kargoucreti(birinci);  }  else {  printf("Yanlis secim!!");  }  break;  case 4:  printf("--Urun Analizleri--\n");  printf("1-Bir urunun toplam satis tutari\n2-Bir urun tipinin toplam satis tutari\n3-Tum urunlerin toplam satis tutari\n");  scanf("%d",&f);  if (f==1){  urunsatis(p1);  }  else if(f==2){  tipsatis(birinciurun,p1);  }  else if(f==3){  tumsatis(p1);  }  break;  case 5:  printf("Sistemden Cikabilirsiniz");  break;    }      return 0;  }  Proje.c  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <math.h>  #include "senaprojee.h"  void instructions( void )  {  printf( "----MENU----\n1.Musteri Bilgisi\n2.Urun Bilgisi\n3.Musteri Analizleri\n4.Urun Analizleri\n5.Cikis\n------------");    }  void tumusteribilgileri(struct musterinode \*a){  int n=0;  while (a != NULL) {  n++;  printf("%d. Musteri ",n);  printf("ID = %d\tIsim-Soyisim = %s\tTip = %d\tX koordinati = %1.f\tY koordinati =%1.f\n",a->ID,a->name,a->tip,a->x,a->y);  a= a->sonraki;  }  }  void tipmusteri(struct musterinode \*b) {  int n =0;  int s;  printf("Musteri tipleri\n 0=Bireysel\t1=Kurumsal\nMusteri tipi seciniz: ");  scanf("%d",&s);    while (b != NULL) {  n++;    if(s==0){  if(b->tip==0){  printf("ID = %d\tIsim-Soyisim = %s\tTip = %d\tX koordinati = %1.f\tY koordinati =%1.f\n",b->ID,b->name,b->tip,b->x,b->y);  }  }  else if(s==1){  if(b->tip==1){  printf("ID = %d\tIsim-Soyisim = %s\tTip = %d\tX koordinati = %1.f\tY koordinati =%1.f\n",b->ID,b->name,b->tip,b->x,b->y);  }  }  else {  printf("Yanlış seçim!");  }  b= b->sonraki;  }  }  void birmusteri(struct musterinode \*c) {  int s,n=0;  printf("Secmek istediginiz musterinin ID numarasini giriniz ");  scanf("%d",&s);  while (c != NULL) {  n++;  if(s == c->ID){  printf("ID = %d\tIsim-Soyisim = %s\tTip = %d\tX koordinati = %1.f\tY koordinati =%1.f\n",c->ID,c->name,c->tip,c->x,c->y);  }  c= c->sonraki;  }  }  void tumurunbilgileri(struct urunbilgisi \*a){  struct urunbilgisi \*temp;  temp=a;  while(temp->sonrakiurun != NULL) {  printf("Urun adi = %s\n",temp->urunname);  temp=temp->urunname;  }  printf("Urun adi = %s",temp->urunname);  }  void tipurun(struct urunbilgisi \*b) {  int n =0;  int s;  printf("Urun tipleri\nEt urunleri =0\NSebze=1\nMeyve =2\nIcecek=3\nMusteri tipi seciniz: ");  scanf("%d",&s);    while (b != NULL) {  n++;    if(s==0){  if(b->uruntip==0){  printf("Urun ID = %d\tUrun ismi %s\tUrun tipi=%d\tUrun fiyati=%f\n",b->urunID,b->urunname,b->uruntip,b->urunucret);  }  }  else if(s==1){  if(b->uruntip==1){  printf("Urun ID = %d\tUrun ismi %s\tUrun tipi=%d\tUrun fiyati=%f\n",b->urunID,b->urunname,b->uruntip,b->urunucret);  }  }  else if(s==2){  if(b->uruntip==2){  printf("Urun ID = %d\tUrun ismi %s\tUrun tipi=%d\tUrun fiyati=%f\n",b->urunID,b->urunname,b->uruntip,b->urunucret);  }  }  else if(s==3){  if(b->uruntip==3){  printf("Urun ID = %d\tUrun ismi %s\tUrun tipi=%d\tUrun fiyati=%f\n",b->urunID,b->urunname,b->uruntip,b->urunucret);  }  }  else {  printf("Yanlış seçim!");  }  b= b->sonrakiurun;  }  }  void birurun (struct urunbilgisi \*c){  int s,n=0;  printf("Secmek istediginiz urunun ID numarasini giriniz ");  scanf("%d",&s);  while (c != NULL) {  n++;  if(s == c->urunID) {  printf("Urun ID = %d\tUrun ismi %s\tUrun tipi=%d\tUrun fiyati=%f\n",c->urunID,c->urunname,c->uruntip,c->urunucret);  }  c = c->sonrakiurun;  }  }  void birmusteriurun(struct parcabilgisi \*urun){  int n=0;  int q;  printf("Secmek istediginiz musterinin ID numarasini giriniz ");  scanf("%d",&q);  while(urun !=NULL){  n++;  if(q== urun->customer\_ID){  printf("Musterinin satin aldigi urunun id numarasi : %d",urun->product\_ID);  }  urun = urun->nextPurchasedPtr;  }  }  void birmusteritutar(struct parcabilgisi \*tutar){  int n=0,p;  printf("Secmek istediginiz musterinin ID numarasini giriniz ");  scanf("%d",&p);  while(tutar !=NULL){  n++;  if(p== tutar->customer\_ID){  printf("Musterinin satin aldigi urununlerin tutari : %f",tutar->cost);  }  tutar = tutar->nextPurchasedPtr;  }  }  void tummusteritutar(struct parcabilgisi \*toptutar){  double a=0;  int n=0;  while(toptutar !=NULL){  n++;  printf("ID numarasi %d olan musterinin tutari = %f\n",toptutar->customer\_ID,toptutar->cost);  a= toptutar->cost + a;  toptutar = toptutar->nextPurchasedPtr;  }  printf("Toplam tutar = %f",a);    }  void kargoucreti(struct musterinode \*kargo){  int n=0;  double z=0,l=0,p=0;  printf("1 km basina yol ucreti 5 tldir!!\n");  while(kargo != NULL){  n++;  z=(kargo->x\*kargo->x)+(kargo->y\*kargo->y);  l=sqrt(z);  p= 5\*l;  printf("ID numarasi %d olan musterinin kargo ucreti = %f\n",kargo->ID,p);  kargo=kargo->sonraki;  }  }  void urunsatis(struct parcabilgisi \*urunsatis){  int n=0;  while(urunsatis != NULL){  n++;  printf("Id numarasi %d olan urunun satis tutari = %f\n",urunsatis->product\_ID,urunsatis->cost);  urunsatis=urunsatis->nextPurchasedPtr;  }  }  void tipsatis(struct urunbilgisi \*tipsatis,struct parcabilgisi \*tut){  int n=0;  int s;  printf("Urun tipleri\nEt urunleri =0\nSebze=1\nMeyve =2\nIcecek=3\nMusteri tipi seciniz: ");  scanf("%d",&s);  while (tipsatis != NULL) {  n++;    if(s==0){  if(tipsatis->uruntip==0){  printf("Urun ID = %d\tUrun fiyati=%f\n",tut->cost);  }  }  else if(s==1){  if(tipsatis->uruntip==1){  printf("Urun ID = %d\tUrun fiyati=%f\n",tut->cost);  }  }  else if(s==2){  if(tipsatis->uruntip==2){  printf("Urun ID = %d\tUrun fiyati=%f\n",tut->cost);  }  }  else if(s==3){  if(tipsatis->uruntip==3){  printf("Urun ID = %d\tUrun fiyati=%f\n",tut->cost);  }  }  else {  printf("Yanlış seçim!");  }  tipsatis= tipsatis->sonrakiurun;  tut=tut->nextPurchasedPtr;  }  }  void tumsatis (struct parcabilgisi \*tum){  int n=0;  double a=0;  while(tum !=NULL){  printf("ID numarasi %d olan urunun tutari = %f\n",tum->product\_ID,tum->cost);  a= tum->cost + a;  tum = tum->nextPurchasedPtr;  }  printf("Urunleri toplam tutari = %f",a);  }  /\*void ekle (char isim[50],int ID,double x,double y,int tip ) {  struct musterinode \*ucuncu=(struct musterinode \*)malloc(sizeof(struct musterinode));  if(basla==NULL) {  basla=ucuncu;  }    else {  temp=basla;  while (temp->sonraki!=NULL){  temp=temp->sonraki;    }    temp->sonraki=ucuncu;    }    }  void yazdir() {  int n =0;  struct musterinode \*a = basla;  while(a != 3){    ++n;  printf("%d. dugum =\n ",n);  printf("ID numarasi = %d",a->ID);  }  }  \*/  Proje.h  #ifndef senaprojee  #define senaprojee  struct musterinode{  int ID;  char name[50];  int tip ;  double x;  double y;  struct musteri \*sonraki;  };  struct urunbilgisi{  int urunID;  char urunname[50];  int uruntip;  double urunucret;  struct urun \*sonrakiurun;  };  struct parcabilgisi{  int ID;  int invoice\_ID;  int customer\_ID;  int product\_ID;  double cost;  struct Purchased \*nextPurchasedPtr;  };    void instructions( void );  void tumusteribilgileri(struct musterinode \*a);  void tipmusteri(struct musterinode \*b);  void birmusteri(struct musterinode \*c);  void tumurunbilgileri(struct urunbilgisi \*a);  void tipurun(struct urunbilgisi \*b);  void birurun(struct urunbilgisi \*c);  void birmusteriurun(struct parcabilgisi \*urun);  void birmusteritutar(struct parcabilgisi \*tutar);  void tummusteritutar(struct parcabilgisi \*toptutar);  void kargoucreti(struct musterinode \*kargo);  void urunsatis(struct parcabilgisi \*urunsatis);  void tipsatis(struct urunbilgisi \*tipsatis,struct parcabilgisi \*tut);  void tumsatis (struct parcabilgisi \*tum);  #endif |