INF212
ALGORITHMS AND PROGRAMMING II

PROJECT-1

Sale Program

Deadline is April 2, 2021 at 17:00.

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.

Information about the project is in the following. The code that will help in the project is attached.
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*).

PROJECT 1

Sale Program

Create a sales automation project that sells products with the help of product information, customer information. The analysis of the purchase products should be made and shown according to the customer, product type, quantity and amount. Keep data with linked lists feature. You can use the attached code from Deitel Fig 12.3.

In sales automation, product information consists of::

product number,
product name,
product type (i.e. 1 for fruit, 2 for vegetable, 3 for meat etc.),
price and
next product object's address (pointer)

```
struct product {
        int ID;
        char name[50];
        int type;
        double price;
        struct Product *nextProPtr;
};
```

Customer information consists of:

customer number,
customer name,
customer type (i.e. 1 for personal, 2 for commercial),
x coordinate of customer location,
x coordinate of customer location and,
next customer object's address (pointer)

```
struct customer {
        int ID;
        char name[50];
        unsigned int type: 1;
        double x_coord;
        double y_coord;
        struct Customer *nextCusPtr;
};
```

Purchased item information consist of as follows:

purchased item number,
invoice number,
customer number,
product number,
product cost and
next purchased item object's address (pointer)

```
struct purchased {
        int ID;
        int invoice_ID;
        int customer_ID;
        int product_ID;
        double cost;
        struct Purchased *nextPurchasedPtr;
};
```

Sales automation menu can be designed as follows:

1. Sales
2. Customer Information
3. Product Information
4. Customer Analysis
5. Product Analysis

After selecting the sales menu, first the last invoice number should be determined, then the purchasing process should be started by entering the customer number.

In product purchase, the product number and quantity should be entered then the product cost should be calculated and the purchase data should be stored. After that the next product purchase process can be passed.

After the purchased items are finished, the shipping fee should be calculated and the purchase process should be completed. The shipping fee should be calculated according to the distance of the customer and the shop.

1. Sales
    - 1.1 Preprocessing
        - 1.1.2 Invoice number
        - 1.1.1 Customer number
    - 1.2 Product sales
        - 1.2.1 Find product number
        - 1.2.2 Enter amount
    - 1.3 Invoice finalization

2. Customer Information
    - 2.1 All customers
    - 2.2 According to the customer type
    - 2.3 Single customer

In the Customer Information section, the information of all customers, customers according to customer type or a single customer should be listed. The choice can be made from the menu.

3. Product Information
    - 3.1 All products
    - 3.2 According to the product type
    - 3.3 Single product

In the Product Information section, the information of all products, products according to product type or a single product should be listed. The choice can be made from the menu.

4. Customer Analysis
    - 4.1 Products purchased by a customer
    - 4.2 Total amount of purchased by a customer
    - 4.3 Total amount of products purchased by all customer
    - 4.4 Customers shipping fee

In the Customer Analysis section, the products purchased by a customer should be listed and the most preferred product and the least preferred product should be determined and shown.
The total amount of the products purchased by a customer's invoice should be listed and the maximum invoice amount and the minimum invoice amount should be obtained and shown.
The total amount of the products purchased by all the customers should be listed.
The total shipping costs of the customers should be listed.

5. Product Analysis
    - 5.1 Total purchase quantity of a product
    - 5.2 Total purchase amount of a product type
    - 5.3 Total purchase amount for all products

In the Product Analysis section, the total purchase amount of the product determined by the user should be listed and shown its quantity .
The total purchase amount of a product type should be listed and shown the most preferred product.
The total purchase amount for all products should be listed.

The header file example for the project:

```
#ifndef productsale_h_
#define productsale_h_

struct customer {
        int ID;
        char name[50];
        unsigned int type: 1;
        double x_coord;
        double y_coord;
        struct Customer *nextCusPtr;
};
typedef struct customer Customer;
typedef Customer * CustomerPtr;

struct product {
        int ID;
        char name[50];
        int type;
        double price;
        struct Product *nextProPtr;
};
typedef struct product Product;
typedef Product * ProductPtr;

struct purchased {
        int ID;
        int invoice_ID;
        int customer_ID;
        int product_ID;
        double cost;
        struct Purchased *nextPurchasedPtr;
};
typedef struct purchased Purchased;
typedef Purchased * PurchasedPtr;

void insertCustomer( CustomerPtr *cPtr, int c_ID, char* c_name, int c_type, double c_x, double c_y);
int deleteCustomer( CustomerPtr *cPtr, int c_ID);
int isEmptyCustomer( CustomerPtr cPtr );
void printCustomer( CustomerPtr currentcPtr);

int getCustomerID( CustomerPtr *cPtr, char* c_name);
char* getCustomerName( CustomerPtr *cPtr, int c_ID);
int getCustomerType( CustomerPtr *cPtr, int c_ID);
double getCustomerGpsX( CustomerPtr *cPtr, int c_ID);
double getCustomerGpsY( CustomerPtr *cPtr, int c_ID);
....
....

#endif
```

PROJE 1

Satış Programı

Ürün bilgileri, müşteri bilgileri yardımı ile ürün satışı yapan satış otomasyon projesi oluşturun.
Projede satılan ürünlerin analizi müşteriye, ürün tipine, miktarına ve tutarına göre yapılıp gösterilsin.
Veriler bağlı listeler özelliği ile tutulsun. Deitel kitabı Şekil 12.3 deki koddan yaralanabilirsiniz.

Satış otomasyonunda ürün bilgileri şunlardan oluşsun:

ürün numarası,
ürün adı,
ürün tipi (örneğin meyve ise 1, sebze ise 2, et ise 3 vb),
fiyatı ve
bir sonraki ürün bilgisinin bulunduğu bellek adresi

```
struct product {
        int ID;
        char name[50];
        unsigned int type : 4;
        double price;
        struct Product *nextProPtr;
};
```

Müşteri bilgileri aşağıdakilerden oluşsun:

müşteri numarası,
müşteri adı,
müşteri tipi (örneğin bireysel müşteri ise 0, ticari müşteri ise 1),
müşterinin coğrafi bilgi sistemindeki x koordinatı,
müşterinin coğrafi bilgi sistemindeki y koordinatı ve
bir sonraki müşteri bilgisinin bulunduğu bellek adresi

```
struct customer {
        int ID;
        char name[50];
        unsigned int type: 1;
        double x_coord;
        double y_coord;
        struct Customer *nextCusPtr;
};
```

Satış yapılan her ürün (parça) ile ilgili bilgiler de aşağıdaki gibi saklansın:

satılan parça numarası,
fatura numarası,
satış yapılan müşterinin numarası,
satılan ürünün ürün numarası,
satılan ürünün tutarı ve
bir sonraki satış yapılan ürün ile ilgili bilgilerin bulunduğu bellek adresi

```
struct purchased {
        int ID;
        int invoice_ID;
        int customer_ID;
        int product_ID;
        double cost;
        struct Purchased *nextPurchasedPtr;
};
```

Satış otomasyon menüsü aşağıdaki gibi olabilir.

1. Satış
2. Müşteri Bilgisi
3. Ürün Bilgisi
4. Müşteri Analizleri
5. Ürün Analizleri

Menüden Satış seçildiğinde öncelikle son fatura numarası bulunmalı sonra müşteri numarası girilerek ürün satış işlemine başlanmalıdır.

Ürün satışta ise ürün numarası ve miktarı girilerek tutar hesaplanıp ilgili veriler hafıza depolanmalı.

Sonraki ürünün satış işlemi gerçekleşmelidir

Satın alınan ürünler bittikten kargo ücreti hesaplanıp satış işlemi bitirilmelidir. Kargo ücreti müşteri ile ürün satılan yerin uzaklığı ile orantılı olarak hesaplanmalıdır.

1. Satış
    - 1.1 Satış Ön İşlem
        - 1.1.2 Son Fatura Numarası Bul
        - 1.1.1 Müşteri Numarası Bul
    - 1.2 Ürün Satış
        - 1.2.1 Ürün No Bul
        - 1.2.2 Miktar Bilgisi Gir
    - 1.3 Satış Son İşlem (Kargo Tutarı Hesapla

2. Müşteri Bilgisi
    - 2.1 Tüm müşteriler
    - 2.2 Müşteri tipine göre
    - 2.3 Bir müşteri

Müşteri Bilgisi kısmında tüm müşterilerin, müşteri tipine göre müşterilerin veya tek bir müşterinin bilgileri listelenmelidir. Seçimi menüden yapılabilsin.

3. Ürün Bilgisi
    - 3.1 Tüm ürünler
    - 3.2 Ürün tipine göre
    - 3.3 Bir ürün

Ürün Bilgisi kısmında tüm ürünlerin, ürün tipine göre ürünlerin veya tek bir ürünün bilgileri listelenmelidir. Seçimi menüden yapılabilsin.

4. Müşteri Analizleri
    - 4.1 Bir müşterinin satın aldığı ürünler
    - 4.2 Bir müşterinin satın aldığı toplam tutar
    - 4.3 Tüm müşterinin satın aldığı ürünlerin toplam tutarı
    - 4.4 Müşterilerin kargo ücreti

Müşteri analizleri kısmında ise bir müşterinin satın aldığı ürünleri listelesin, en fazla tercih ettiği ürünü ve en az tercih ettiği ürünü göstersin.
Bir müşterinin satın aldığı ürünlerin toplam tutarını listelesin, en fazla fatura bedelini ve en az fatura bedelini göstersin.
Tüm müşterinin satın aldığı ürünlerin toplam tutarı listelesin.
Müşterilerin toplam kargo ücretini listelesin.

5. Ürün Analizleri
    - 5.1 Bir ürünün toplam satış tutarı
    - 5.2 Bir ürün tipinin toplam satış tutarı
    - 5.3 Tüm ürünlerin toplam satış tutarı

Ürün analizleri kısmında ise bir ürünün toplam satın alınan tutarı listelesin. Satın alınan miktarı göstersin.
Bir ürün tipinin toplam satın alınan tutarı listelesin.
En fazla hangi ürünün tercih edildiğini göstersin.
Tüm ürünlerin toplam satın alınan tutarı listelesin.

Proje için aşağıdaki header file örneğinden yararlanabilrsiniz.

```
#ifndef productsale_h_
#define productsale_h_

struct customer {
        int ID;
        char name[50];
        unsigned int type: 1;
        double x_coord;
        double y_coord;
        struct Customer *nextCusPtr;
};
typedef struct customer Customer;
typedef Customer * CustomerPtr;

struct product {
        int ID;
        char name[50];
        unsigned int type : 4;
        double price;
        struct Product *nextProPtr;
};
typedef struct product Product;
typedef Product * ProductPtr;

struct purchased {
        int ID;
        int invoice_ID;
        int customer_ID;
        int product_ID;
        double cost;
        struct Purchased *nextPurchasedPtr;
};
typedef struct purchased Purchased;
typedef Purchased * PurchasedPtr;

void insertCustomer( CustomerPtr *cPtr, int c_ID, char* c_name, int c_type, double c_x, double c_y);
int deleteCustomer( CustomerPtr *cPtr, int c_ID);
int isEmptyCustomer( CustomerPtr cPtr );
void printCustomer( CustomerPtr currentcPtr);

int getCustomerID( CustomerPtr *cPtr, char* c_name);
char* getCustomerName( CustomerPtr *cPtr, int c_ID);
int getCustomerType( CustomerPtr *cPtr, int c_ID);
double getCustomerGpsX( CustomerPtr *cPtr, int c_ID);
double getCustomerGpsY( CustomerPtr *cPtr, int c_ID);
....
....

#endif
```