Soru - 1:

NYP dersinin ilk haftasında işlenen, C dili ile yazılmış Nokta programını kullanarak belirtilen iki noktanın bir koordinat sistemde kare ifade edip etmediğini belirten bir işlev ekleyiniz.

#include <stdio.h>

#include <stdlib.h>

struct point {

int x;

int y;

};

struct point \*

point\_create(int x, int y)

{

struct point \*this =

(struct point \*)malloc(sizeof(struct point));

this->x = x;

this->y = y;

return this;

}

int

point\_isright(const struct point \*this, const struct point \*that)

{

return this->x > that->x;

}

int

point\_isleft(const struct point \*this, const struct point \*that)

{

return this->x < that->x;

}

int

point\_isabove(const struct point \*this, const struct point \*that)

{

return this->y > that->y;

}

int

point\_isbelow(const struct point \*this, const struct point \*that)

{

return this->y < that->y;

}

int

point\_isequal(const struct point \*this, const struct point \*that)

{

return this->x == that->x && this->x == that->y;

}

void

point\_move(struct point \*this, int x, int y)

{

this->x += x;

this->y += y;

}

void

point\_print(const struct point \*this)

{

printf("(%d,%d)", this->x, this->y);

}

void

point\_destroy(struct point \*this)

{

free(this);

}

int

main(void)

{

struct point \*p = point\_create(3, 5);

struct point \*q = point\_create(5, 7);

point\_print(p), printf(" ile ");

point\_print(q), printf(" noktalari kare ");

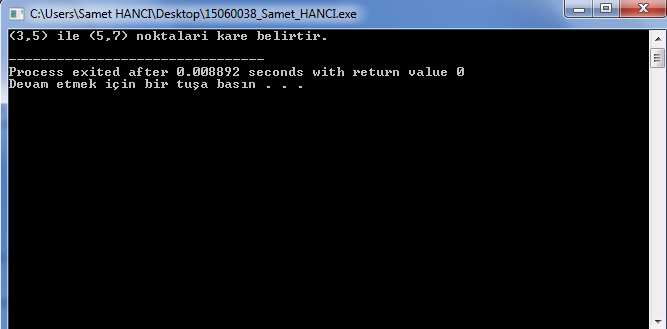
printf("%s.\n", point\_isleft(p, q) ? "belirtir" : "belirtmez");

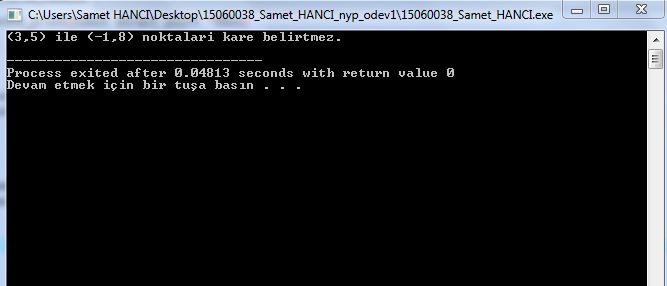
point\_destroy(p);

point\_destroy(q);

return 0;

}





Soru - 2:

Bir araba galerisi tüm araç bilgilerine bir noktadan hakim olmak için bir yazılım istemektedir. Bu konuyla ilgili ilk adımlar için bazı bilgileri tutmak amacıyla sınıf yazmanız gerekmektedir. Bu sınıf arabanın tipi (taksi, minibüs, kamyon vb.), bulunduğu il ve fiyat bilgilerini tutulacaktır. Bu değişkenler üzerinde değişiklik yapılmayacağı için sadece okunabilir olması yeterlidir. Sınıfın tüm değişkenlerine ilk değer atayabilecek bir kurucu metodu bulunmalıdır. Tüm bu bilgilere göre 2 tane nesne oluşturup bu nesneler ile şu işlemler yapılmak istenmektedir:

- Fiyat bilgilerini pahalı, ucuz veya eşit yönleriyle karşılaştırma.

- Verilen bir miktara göre indirim veya zam yapma.

Bütün bu işlemleri yapacak kodu Ruby dili ile gerçekleyiniz.

class Karsilastirma

attr\_reader :aractipi, :bulunduğuil, :fiyat

def initialize(aractipi, bulunduğuil, fiyat)

@aractipi, @bulunduğuil, @fiyat = aractipi, bulunduğuil, fiyat

end

def >(other)

@fiyat > other.fiyat

end

def <(other)

@fiyat < other.fiyat

end

def ==(other)

@fiyat == other.fiyat

end

def to\_s

"(#{@fiyat},#{@fiyat})"

end

def zam

@fiyat \* 1.1

end

def indirim

@fiyat \* 0.9

end

end

n1 = Karsilastirma.new("taksi","Samsun",20000)

n2 = Karsilastirma.new("kamyon","Kirsehir",80000)

if !(n1 == n2)

puts("İki aracın fiyatları eşit değil.")

end

if n2 > n1

puts("İki aracın fiyatlarından Samsun'daki Taksi fiyatı diğerine göre daha pahalı değil.")

end

if n1 < n2

puts("İki aracın fiyatlarından Samsun'daki Taksi fiyatı diğerine göre daha ucuz.")

end

puts "Samsun'daki Taksi'nin indirimli fiyatı #{n1.indirim.to\_i}"

puts "Samsun'daki Taksi'nin zamlı fiyatı #{n1.zam.to\_i}"

gets

