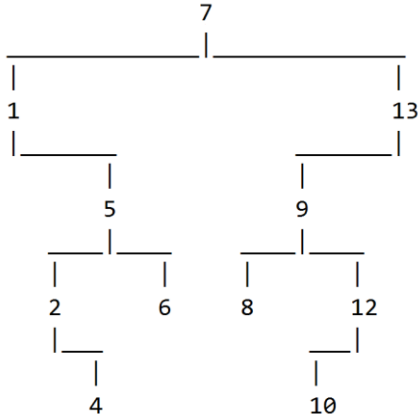




NUMARA :	AD SOYAD :	DEĞERLENDİRME	
Sınavda Uyulması Gereken Kurallar	İMZA :	[.....]
1. Cep telefonlarının saate bakmak için bile olsa herhangi bir amaçla kullanılması yasaktır. Telefon kapalı ve cepte olmalıdır. 2. Sınavın başında sorular kısaca açıklanacaktır. Öğrencilerin soruları cevaplandıktan sonra sınav boyunca soru sormak yasaktır. 3. Soru kağıdına numaranızı ve isminizi yazıp imzalamayı unutmayınız.			

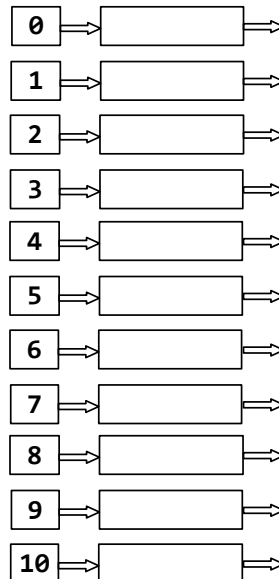


1. Yukarıdaki splay ağacına 3'ü ekleyiniz. (30P)

<pre>int Hash (char* key) { int sum = 0; for (int j=0; j<4; j += 2) sum += 4*key[j] + key[j+1]; sum = sum % 11 ; return sum; }</pre>	<p>dictionary.txt</p> <table><tr><td>list</td><td>liste</td></tr><tr><td>nuts</td><td>findik</td></tr><tr><td>object</td><td>nesne</td></tr><tr><td>order</td><td>duzen</td></tr><tr><td>parent</td><td>baba</td></tr><tr><td>queue</td><td>kuyruk</td></tr><tr><td>stack</td><td>yigin</td></tr><tr><td>tree</td><td>agac</td></tr></table>	list	liste	nuts	findik	object	nesne	order	duzen	parent	baba	queue	kuyruk	stack	yigin	tree	agac
list	liste																
nuts	findik																
object	nesne																
order	duzen																
parent	baba																
queue	kuyruk																
stack	yigin																
tree	agac																

2. Yukarıda dictionary.txt'de verilen kelimeleri Hash() fonksiyonunu ve çakışma çözümü yöntemi olarak linear probing'i kullanarak relative.txt'ye yazınız. Ayrıca synonym chaining yöntemine göre bağlı listelere ilgili kayıtları ekleyiniz. (30P)

relative.txt		a-97	n-110
0		b-98	o-111
1		c-99	p-112
2		d-100	q-113
3		e-101	r-114
4		f-102	s-115
5		g-103	t-116
6		h-104	u-117
7		i-105	v-118
8		j-106	w-119
9		k-107	x-120
10		l-108	y-121
ASCII Tablo →		m-109	z-122



```

bool empty()
{
    return (header->next == trailer);
}

void addFront(const int& i)
{
    add(header->next, i);
}

void add(DoublyNode* v, int& i)
{
    DoublyNode* u = new DoublyNode;
    u->score = i;
    .....
    .....
    .....
    .....
}

void main()
{
    DoublyLinkedList list;
    list.addFront(750);
    list.addFront(720);
}

```

3. Yukarıdaki programda add() fonksiyonunda ile temsil edilen satırlar:

i) (20P)

(Yanlış cevaptan 5P kırılacaktır)

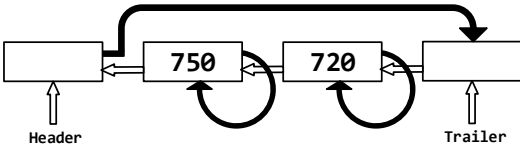
```

v->prev->next = u;
v->prev      = u;
u->prev      = v->prev;
u->next      = v;

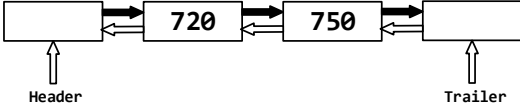
```

olduğunda listenin son hali :

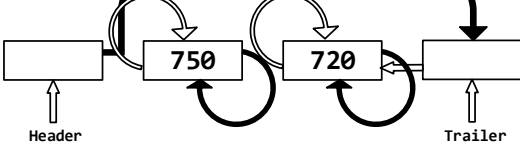
(A)



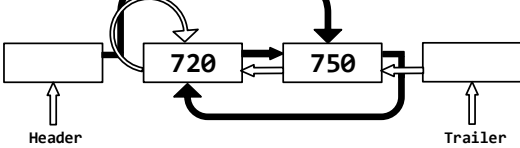
(B)



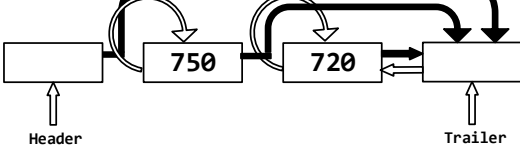
(C)



(D)



(E)



ii) (20P)

(Yanlış cevaptan 5P kırılacaktır)

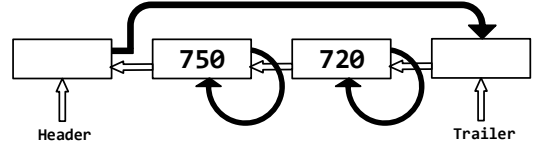
```

v->prev      = u;
v->prev->next = u;
u->prev      = v->prev;
u->next      = v;

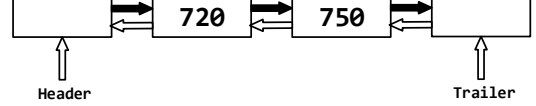
```

olduğunda listenin son hali :

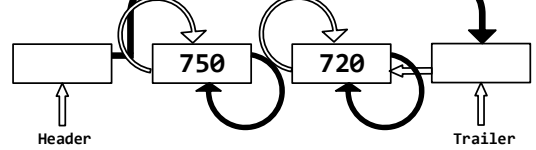
(A)



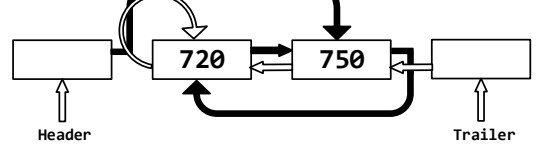
(B)



(C)



(D)



(E)

