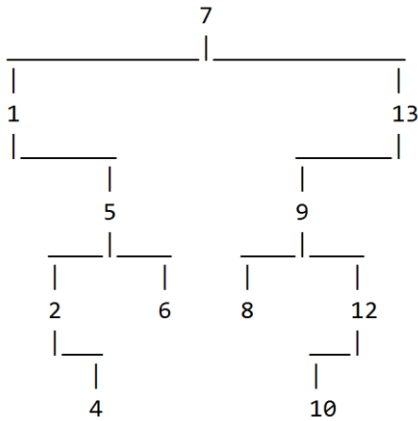




<b>NUMBER :</b> .....	<b>NAME :</b> .....	<b>EXAM GRADE</b>	
<b>Rules to be Obeyed During the Exam</b>		[.....]	.....
<b>SIGNATURE :</b> .....			

1. Cell phones are not allowed to be used as a calculator or a watch. They must be switched off and placed in the pocket.  
 2. Brief information about the exam will be given at the beginning, then no one is not allowed to ask a question during the exam.  
 3. Do not to forget to sign this paper after writing your number and name.

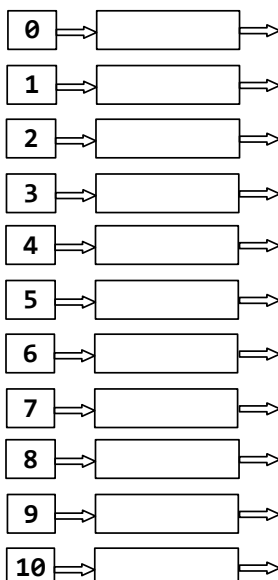


1. Insert **11** into the splay tree above. (30P)

<pre> int Hash (char* key) {   int sum = 0;   for (int j=0; j&lt;4; j += 2)     sum += 3*key[j] + key[j+1];    sum = sum % 11 ;   return sum; }         </pre>	<p style="text-align: center;"><u>dictionary.txt</u></p> <table style="width:100%; border-collapse: collapse;"> <tr><td>array</td><td>dizi</td></tr> <tr><td>binary</td><td>ikili</td></tr> <tr><td>child</td><td>cocuk</td></tr> <tr><td>circuit</td><td>devre</td></tr> <tr><td>class</td><td>sinif</td></tr> <tr><td>client</td><td>istemci</td></tr> <tr><td>gate</td><td>kapi</td></tr> <tr><td>root</td><td>kok</td></tr> </table>	array	dizi	binary	ikili	child	cocuk	circuit	devre	class	sinif	client	istemci	gate	kapi	root	kok
array	dizi																
binary	ikili																
child	cocuk																
circuit	devre																
class	sinif																
client	istemci																
gate	kapi																
root	kok																

2. Write words from dictionary.txt to relative.txt using Hash() function to calculate relative addresses and linear probing as a collision resolving method. In addition, add related words to linked lists using synonym chaining method. (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 Table →		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. Taking into account the lines represented by ..... in the function add() answer the following choices :

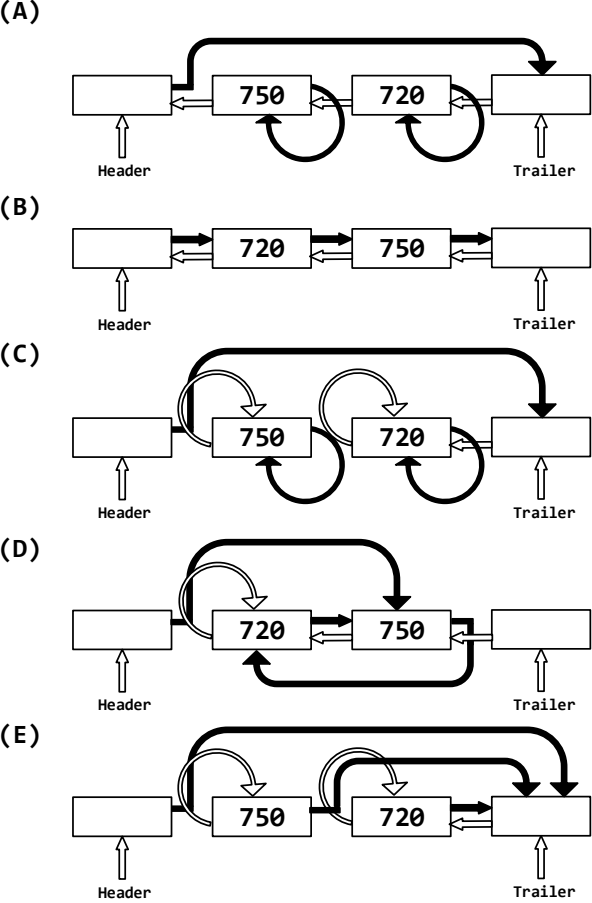
i) (20P) (You'll loose 5Ps from wrong answer)  
 If the lines are like these

```

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

```

the linked list will be :



ii) (20P) (You'll loose 5Ps from wrong answer)

If the lines are like these

```

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

```

the linked list will be :

