**VERİ ANALİZİ FİNAL SINAVI ( 30 Nisan 2014)**

**SORU 1)**

Bir tek yönlü bağlı liste bilgisayar belleğinde bulunmaktadır (ilk düğümün işaretçisi biliniyor). Herbir düğüm tamsayı değerler içermektedir.

**a)(10)** Listede tek sayı ve çift sayı içeren düğümlerin sayısını bulan C fonksiyonu yazınız.

**b) (15)** Aynı listenin tersini, yığıt yapısı kullanarak, yeni bir bağlı liste olarak oluşturan C fonksiyonu yazınız.

**SORU 2)**

Her bir düğümünde tamsayı değerler bulunan bir ikili arama ağacı bilgisayar belleğinde bulunmaktadır ( kökün işaretcisi biliniyor).

**a) (10)** Ağacın yapraklarındaki değerlerin aritmatik ortalamsını bulan C fonksiyonu yazınız.

**b) (15)** Aynı ağaçta kökte bulunan değerden büyük en küçük ve kökte bulunan değerden küçük en büyük değeri bulan C fonksiyonu yazınız.

**SORU 3) (20)**

Arka sayfada verilen C programını adım adım açıklayarak yazdırılan değerleri belirleyiniz.

**SORU 4)**

48, 76, 68, 23, 58, 42, 11, 60, 70, 80, 55, 50 değerlerini sırası ile kullanarak

**a) (10)** AVL ağacı oluşturunuz.

**b) (10)** Heap oluşturunuz ve kökünü bir kez siliniz.

**SORU 5)**

**a) (10) ( X – Y \* ( Z + H / M )^K ) / P – R** infiks ifadesini yığıt yapısı kullanarak postfiks ifadeye dönüştürünüz.

**b) (10)** Aynı ifadenin bir ikili ağaç gösterilimini oluşturunuz. Bu ağacın LRN (postfiks) taramasını yapınız.

typedef struct yapi{

int deger;

struct yapi \*sonraki;

}YAP;

YAP\*ON=NULL,\*ARKA = NULL;

int par = 0;

**int main ()**

{ int i, m;

void yaz( YAP\* ptr );

void ekle ( int );

int sil( void );

void olustur ( int );

printf("\n no son rakami..? ");

scanf("%d", &m);

m = m%2 == 1 ? 3:5;

while( par < 5 )

{ olustur( m + par );

par++;

}

yaz ( ON );

while ( ON -> sonraki != NULL )

{ ekle( sil() + sil() );

yaz ( ON );

} }// main sonu

**void olustur( int x )**

{

YAP\* ptr;

if( ON == NULL )

{

ON=(YAP\*) malloc(sizeof(YAP ));

ON -> deger = x;

ON -> sonraki = NULL;

ARKA = ON;

return;

}

ARKA -> sonraki = ptr = ( YAP\* ) malloc (sizeof( YAP ));

ptr -> deger = x;

ptr -> sonraki = NULL;

ARKA = ptr;

}//oluştur sonu

**int sil()**

{

int x;

YAP\* ptr = ON;

x = ON -> deger;

ON = ptr -> sonraki;

free( ptr );

return ( x );

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void ekle ( int n )**

{

YAP\* ptr;

ptr = ( YAP\* ) malloc (sizeof( YAP ));

ptr -> deger = n;

if( n%2 == 1 )

{

ptr -> sonraki = ON;

ON = ptr;

}

else

{

ptr->sonraki = NULL;

ARKA -> sonraki = ptr;

}

}

void yaz ( YAP \*ptr )

{

printf("\n");

while( ptr != NULL)

{

printf ("%d ", ptr -> deger );

ptr = ptr -> sonraki;

}

}