



MUH + +++++ MNH

## VERI ANALIZI- ARASINAV

## 30 Nisan 2009

SORU 1) Her bir düğümünde tamsayı değerler bulunan bir bağlı liste bilgisayar belleğinde bulunmaktadır (ilk dugümün işaretcisi biliniyor). Bu listedeki düğüm sayısını n yi bulan ve listenin n/2. düğümünü silen C fonksiyonu yazınız.

SORU 2) Düğümlerinde tamsayı değerler bulunan bir ikili arama ağacı bilgisayar belleğinde bulunmaktadır ( kökün işaretcisi biliniyor).

- a) Ağaçtaki düğüm sayısını
- b) Ağaçtaki en büyük ve en küçük değerleri içeren düğümlerin babalarının değerlerini bulan C fonksiyonu yazınız.

SORU 3)

88, 210, 20, 100, 46, 78, 500, 5, 66, 90, 420, 30, 120, 620, 50, 180, 95, 300, 250

değerlerini sırasıyla kullanarak 5.Mertebe B – Ağacı oluşturunuz. Bu ağaçtan sırasıyla 30, 120 ve 100 değerlerini siliniz.

SORU 4) 48, 125, 90, 5, 56, 77, 16, 100, 200, 60, 120 değerlerini kullanarak

- a) Dengeli ikili arama ağacı oluşturunuz.
- b) Heap oluşturunuz ve kökünü siliniz.

SORU 5) Arka sayfada verilen C programını adım açıklayarak yazdırılan değerleri belirleyiniz.



```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
typedef struct yapi{
int deger;
struct yapi *sonraki;
}YAP;
YAP *ON = NULL, *ARKA = NULL;
YAP*yeni(void);
int ust = 0;
int main(void)
   int i, m;
   void yaz( YAP*ptr );
   void ekle( int );
   int sil( void );
   void olustur( int );
   printf("\n numaranizin son rakami?");
   scanf( "%d", &m );
   m = m\%2 == 1?6:9;
  while (ust < 5)
   { olustur( 21\%(m + ust + 1));
    ust++; }
   yaz(ON);
  while(ON -> sonraki != NULL)
     { ekle( sil() - sil() );
     yaz(ON); }
    }\\ main sonu
void olustur( int x )
{ YAP*ptr;
 if( ust == 0 )
  \{ ON = yeni(); 
   ON \rightarrow deger = x;
```

ON -> sonraki = NULL;

```
ARKA = ON; return;
      ARKA -> sonraki = ptr = yeni();
    ptr -> deger = x;
    ptr -> sonraki = NULL; ARKA = ptr;
 } \\oluştur sonu
int sil()
 {YAF*ptr = ON;}
   int x = ON \rightarrow deger;
   ON = ptr -> sonraki;
                           return(x); }\\sil sonu
void ekle( int n )
   YAP*ptr;
  ptr = yeni();
  ptr \rightarrow deger = n;
   if(n \% 2 == 1)
   { ptr -> sonraki = ON;
    ON = ptr;
    return; }
 else
   { ARKA -> sonraki = ptr;
    ptr -> sonraki = NULL;
    ARKA = ptr;
} \\ekle sonu
YAP* yeni()
{ return(( YAP* )malloc( sizeof( YAP ) ) ); }
void yaz(YAP*ptr)
   printf("\n");
  while(pir != NULL)
  { printf(" %d",ptr->deger);
    ptr = ptr -> sonraki; } \\yaz sonu
```