

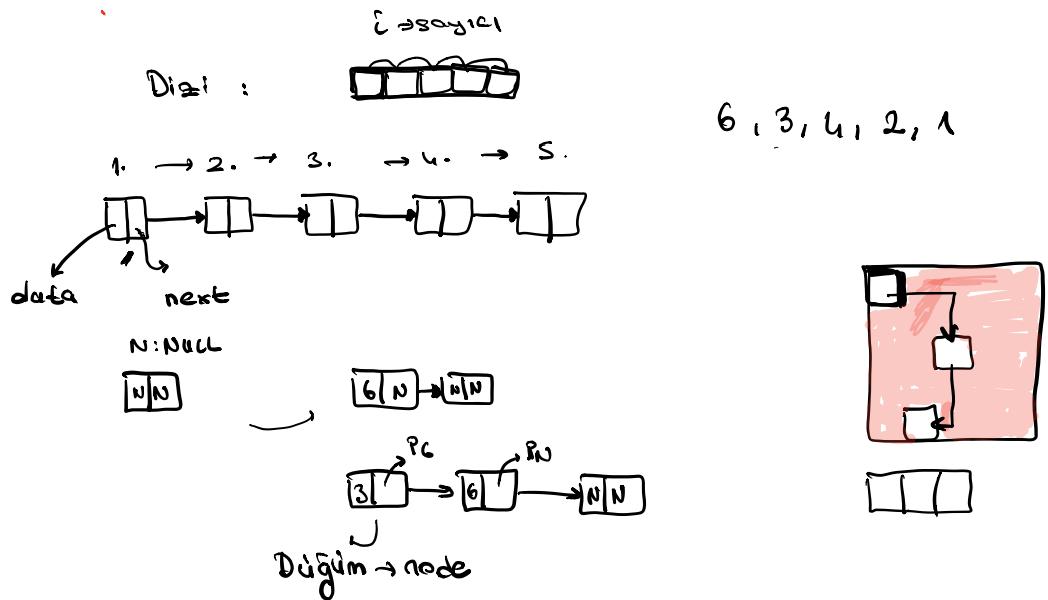
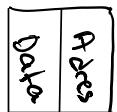
Bağlantılı (Linked) Liste: Sıralı bir biçimde pointer ile bağlı elementlerden oluşan liste veri yapısıdır.

Tekli bağlantılı liste her elemen (düğüm) bir veri türü ve bir sonrakî elemenin log'una sahip türdir.

Bağlantılı liste elementlarına düğüm (node) denir.

Bir düğümün iki alanı vardır →

Örnek:



Veri yapısı:

```
struct node {  
    float info;  
    struct node *next;  
};
```

Bellekten bir düğümlük yer alma:

NODEPTR : bizim tüm programımızda genel node

tipi degisletirmez.

typedef struct node *NODEPTR

NODEPTR getNode() → Bellekten yeri al!
{ NODEPTR p = (NODEPTR) malloc (sizeof (struct node));

return p;

}

Bellekten alinan yeri servest birde;

void freeNode(NODEPTR p)
{ free(p);

g

linked List icinde kullanilan Fonksiyonlar:

RNA

① Push()

② Pop()

③ Insert After()

④ Delete After()



Liste Basina Dugum Ekle:

→ main() icinde

NODEPTR list=NULL;

void push(NODEPTR *list, float k)

{

NODEPTR kutu=getNode(); /Bellekten dugumu a.

kutu->info=k; /Dugumun veri alanini doldur

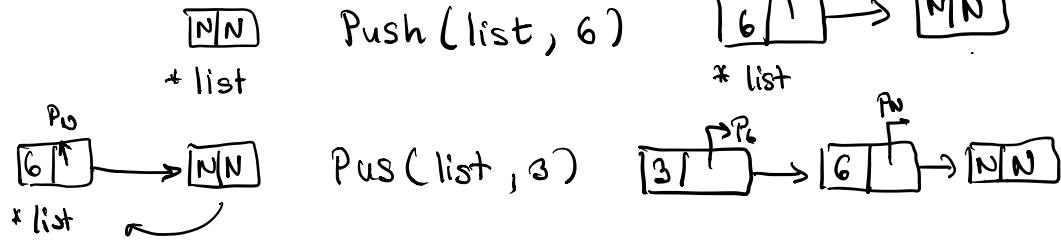
kutu->next=*list;

*list=p;

g

Oturul:

6 3 4 2 1



Linked List element Ekleme:

```

float pop (NODEPTR *list)
{
    if (*list != NULL)
    {
        NODEPTR p = *list; / Silinecek düğüm
        *list = (*list) ->next; / Liste başını güncelle.
        float k = p->info; / düğüm ian deyr uygula

        freeNode(p); / düğümü sil
        return k;
    }
    else
        printf ("Liste Boş");
    return -1; / Hata
}

```

ARAÇA ELEMENT EKLE :

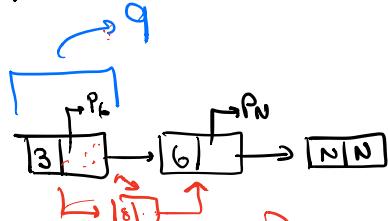
void insertAfter (NODEPTR q, float k)

{ if (q != NULL)

```

    {
        NODEPTR p = getNode();
        p->info = k;
        p->next = q->next;
        q->next = p;
    }
}

```

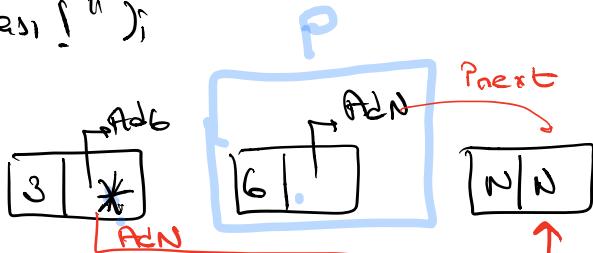


```

    }
    else
        printf ("Adres hatalı!");
}

```

Aradon düşüm Sil!



float deleteAfter (NODEPTR q)

```

{
    if (q != NULL && q->next != NULL)
    {
        NODEPTR p = q->next; // Sil
        float k = p->info;
        q->next = p->next;
        freeNode(p);
        return k;
    }
    else
        printf ("Adres hatalı!");
    return 0;
}

```

Ekrana yazdır:

```

void printList (NODEPTR list)
{
    while (list != NULL)
    {
        printf ("\t.%f ", list->info);
        list = list->next;
    }
}

```

3 İstekler sonunda list isaretçisi ^{yeni} deplâmerz
(adres göndermedik)

```
int main( )
{
    NODEPTR list = NULL;

    Push(&list, 6);
    push(&list, 8);
    printf("listenin son hali ");
    printList(list);

    Pop(&list);
    printf("listenin son hali ");
    printList(list);

    NODEPTR p = list;
    p->next;
    insertAfter(p, 11);
    printf("listenin son hali ");
    printList(list);

    deleteAfter(p);
    printList(list);
    return 0;
}
```