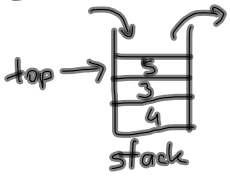
STACKS (Yığınlar) ADT

Bir stack LIFO (Last-in-First-out) prensibine göre nesnelerin eklenip, silindiği veri yapısıdır.
son giren ilk çıkar

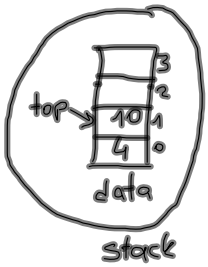


Gerçek hayatta stackler:
- şarjör
- yemekhanedeki tepsiler
- el feneri

1-) stacklerin Dizi Implemantasyonu

```
# define STACK_SIZE 4

typedef struct{
    int data[STACK_SIZE];
    int top;
} stack;
```

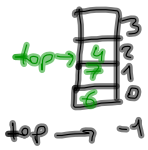


i) Push (Ekleme işlemi)

```
void push (stack *stk, int c){
    if (stk->top == STACK_SIZE-1)
        printf ("stack dolu");
    else {
        stk->top++;
        stk->data[stk->top] = c;
    }
}
```

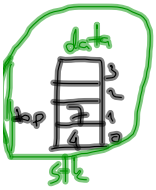
ii) reset işlemi

```
void reset (stack *stk){
    stk->top = -1;
}
```



iii) Pop (Çıkarma işlemi)

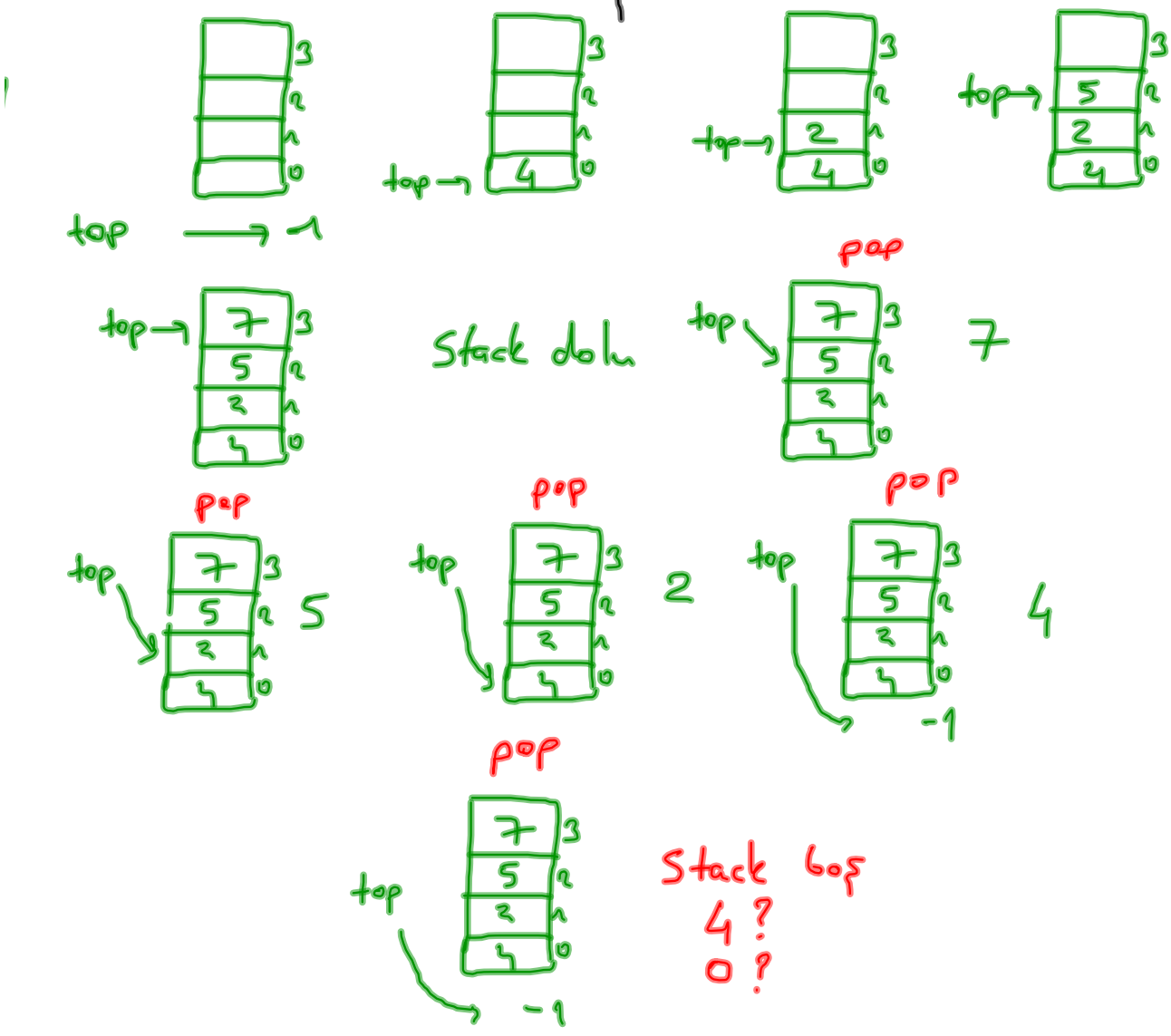
```
int pop (stack *stk){
    if (stk->top == -1)
        printf ("stack boş");
    else
        return (stk->data[stk->top--]);
}
```



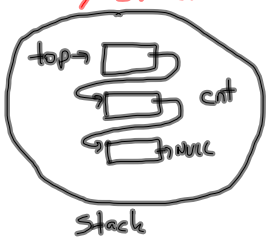

```
y=7;
x= y++;
```
x=y;
y=y+1;

```
y=7;
x= ++y;
```
y=y+1;
x=y;

```
main( ){
    stack n; int x;
    reset(&n);
    push(&n,4);
    push(&n,2);
    push(&n,5);
    push(&n,7);
    push(&n,11); ?
    x= pop(&n);
    printf("%d", x);
    x= pop(&n);
    printf("%d", x);
    x= pop(&n);
    printf("%d", x);
    x= pop(&n);
    printf("%d", x);
    x= pop(&n);
    printf("%d", x);
}
```



2.) Stacklerin Bağlı Liste Implementasyonu



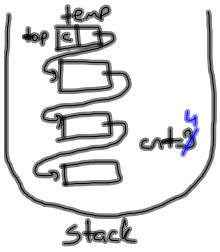

```
typedef struct{
   struct node *top;
   int cnt; //stacktek: elemen sayısı
}stack;
```

i)
```
typedef enum{false,true} boolean;

boolean isempty(stack *stk){
  return(stk->cnt==0);
}
```

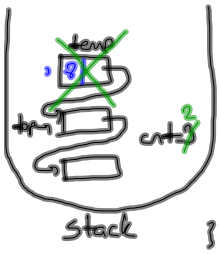
ii)
```
boolean isfull(stack *stk){
  return(stk->cnt==STACK_SIZE);
}
```

iii)
```
void push(stack *stk,int c){
```




```
  if(!isfull(&stk)){
     struct node *temp=new node();
     temp->data=c;
     temp->next=stk->top;
     stk->top=temp;
     stk->cnt++;
  }
}
```

iv)
```
int pop(stack *stk){
  if(!isempty(stk)){
     struct node *temp=stk->top;
     stk->top=stk->top->next;
     int x=temp->data;
     free(temp);
     stk->cnt--;
     return x;
  }
}
```



v)
```
int top(stack *stk){
  if(!isempty(stk)){
     return(stk->top->data);
  }
}
```

vi)
```
void initialize(stack *stk){
   stk->top=NULL;
   stk->cnt=0;
}
```

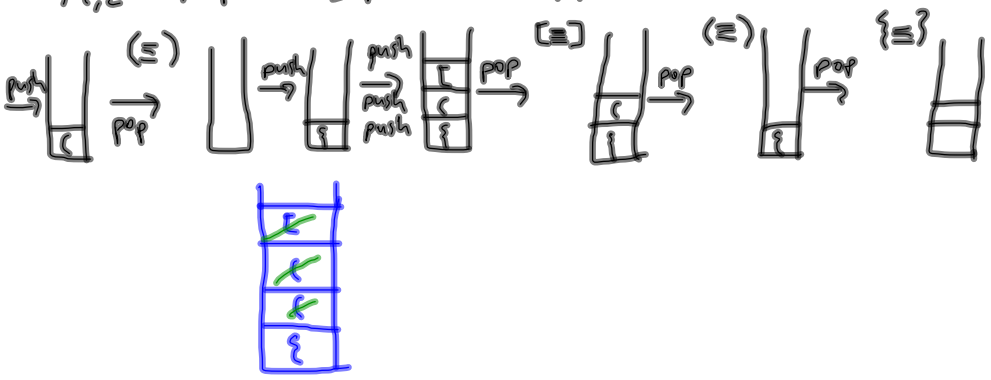
Stackler bilgisayarda nerede kullanılır

- Recursive fonksiyonlarda.
- Postfix → infix dönüşümlerinde
- (,{,[,],},) ayraçlarının C/C++ derleyicisinin kontrollerinde

fact(4)
↓
fact(3)
↓
⋮
fact(0)=1

Örnek:
```
while(x==7){
  printf("%d",a[i]);
}
```

(,{,[→ push yapın),},] → pop yapın



Infix, Prefix, Postfix

Operator: +, −, /, *, ^
Operand: A, B, C

A+B +AB
AB+

Infix	Prefix	Postfix
A+B−C	−+ABC	AB+C−
(A+B)*(C−D)	*+AB−CD	AB+CD−*
A−B/(C*D^E)	−A/B*C^DE	ABCDE^*/−

Örnek: 3 2 * 5 6 * + postfix ifadesini stack kullanarak hesaplayın.