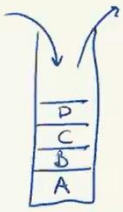
STACK (YIĞIN) (Abstract Data Type)



ABCD DCBA

First In Last Out (FILO)

Last In First Out (LIFO)

Applications

① Evaluation of Arithmetic Operations

A+(B+C-(D/E*F)*G)*H)

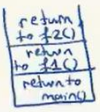
② Recursive Function Usage
Function Call

```
int main(){
   :
   f1()
   :
}
```

```
void f1(){
   :
   f2()
   :
}
```

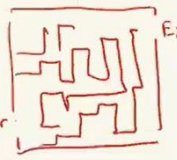
```
void f2(){
   :
   f3()
   :
}
```

```
void f3(){
   :
}
```



③ Reversing String

④ Backtracking



⑤ Parenthesis Matching

[a+(b*c)+{(d-e)}]

⑥ Conversion of Decimal to Other Number Systems

Stack Operations

Stack → Array
→ Linked List

- push()

- pop()

- peek()
  top()

→ get the value of the last element in the stack.
top

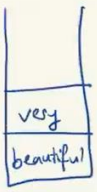
- isFull()
- isEmpty()

Stack Usage

①

The weather is very (2) beautiful. (1)

partly cloudy



Undo

The weather is very

Undo

The weather is very beautiful.

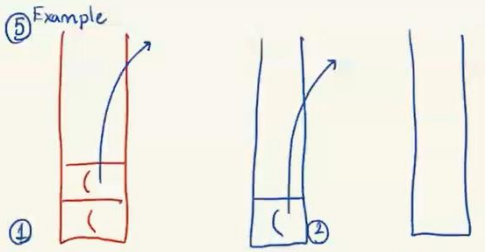
② Stack Usage in Compilers

① - (a + b (error
②) a + b (error
③ a * (b + c (error
④ a) * (b + c (error

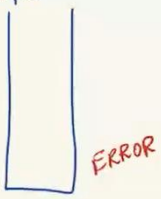
Syntax Error

⑤ ((a+b)*c) ✓
Example

⑤ Example

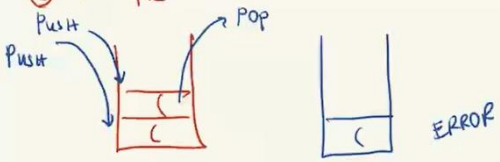


② Example



(→ Push
) → Pop

③ Example

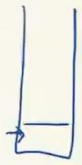


Stack Operations

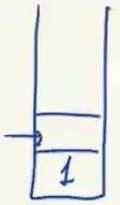
```
#define MAX 10
typedef struct {
        int item[MAX];
        int top;
} STACK;
int push (int X, STACK *s){
        if (isFull(s))
            return 0;
        else {
            s->item[s->top] = X;
            ++s->top;
            return 1;
        }
}
```

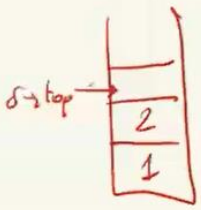
s->item[s->top++]=X

```
void initStack (STACK *s){
    s->top = 0;
}
int isEmpty (STACK *s) {
    if (s->top==0)
        return 1;
    else
        return 0;
}
int isFull (STACK *s) {
    if (s->top == MAX)
        return 1;
    else
        return 0;
}
```



```
int pop (STACK *s, int *X){
    if (isEmpty(s))
        return 0;
    else{
        --s->top;
        *X = s->item[s->top];
        return 1;
    }
}
```






```
int peek (STACK *s, int *X){
    int adr;
    if (isEmpty(s))
        return 0;
    else{
        adr = s->top-1;
        *X = s->item[ adr ];
        return 1;
    }
}
```

top

Arithmetic Expression Evaluation

(3+5)*2/3+7

a+b*c

- Infix
- Prefix
- Postfix

Infix Expression	Prefix Expression	Postfix Expression
A+B	+AB	AB+
A+B*C	+A*BC	ABC*+
A+B*C+D	++A*BCD	ABC*+D+

Precedence of Arithmetic Operations

(

*/

+-

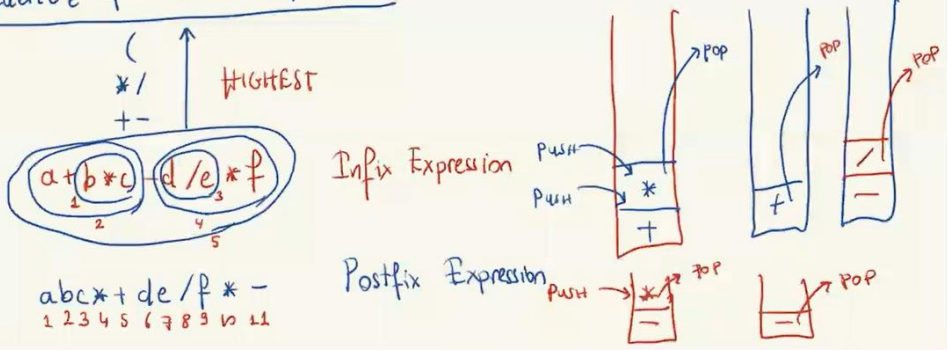
HIGHTEST

a+b*c-d/e*f

1 2 4 5

abc*+de/f*-

1 2 3 4 5 6 7 8 9 10 11

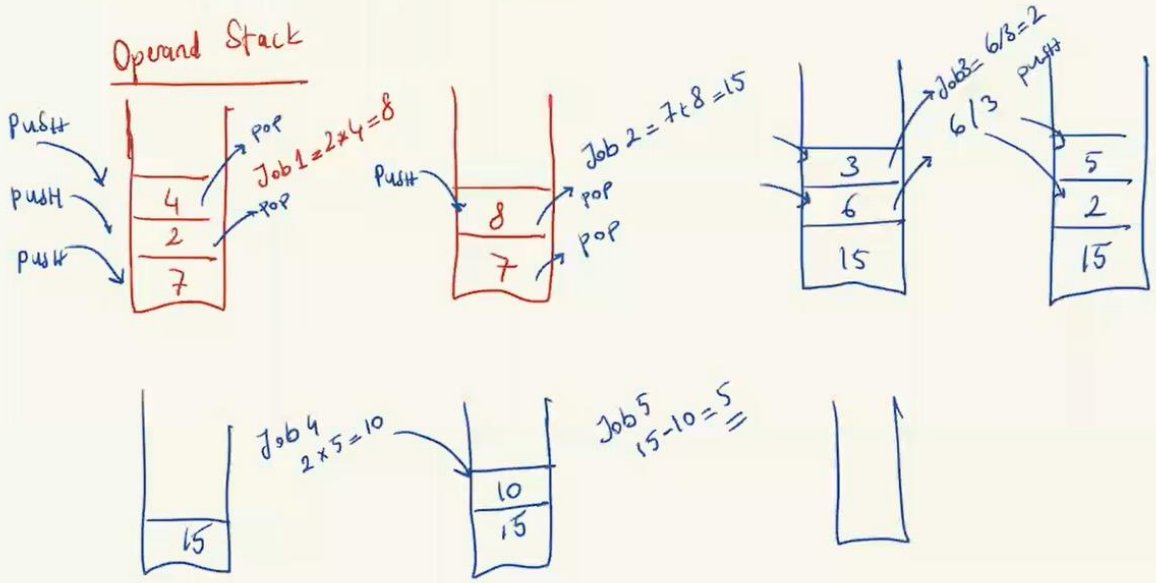


a=7 b=2 c=4 d=6 e=8 f=5

abc*+de/f*-

724*+68/5*- Postfix Expression

Operand Stack



Stack

3*(5+4/1)-2 Infix

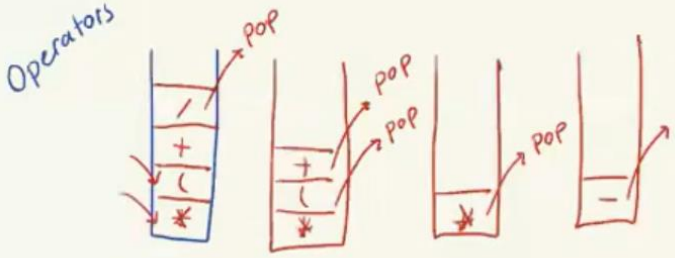
3541/+*2- Postfix

Precedence

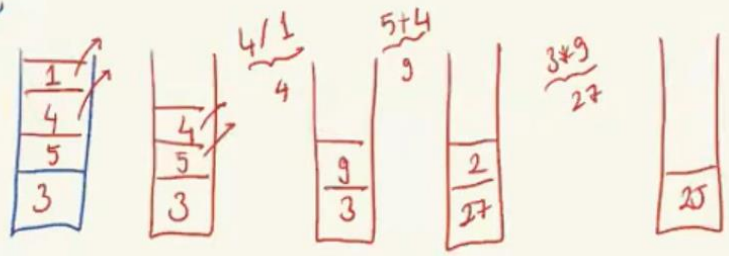
PUSH	POP
(	*/
*/	+-
+-	(

HIGHEST

Operators

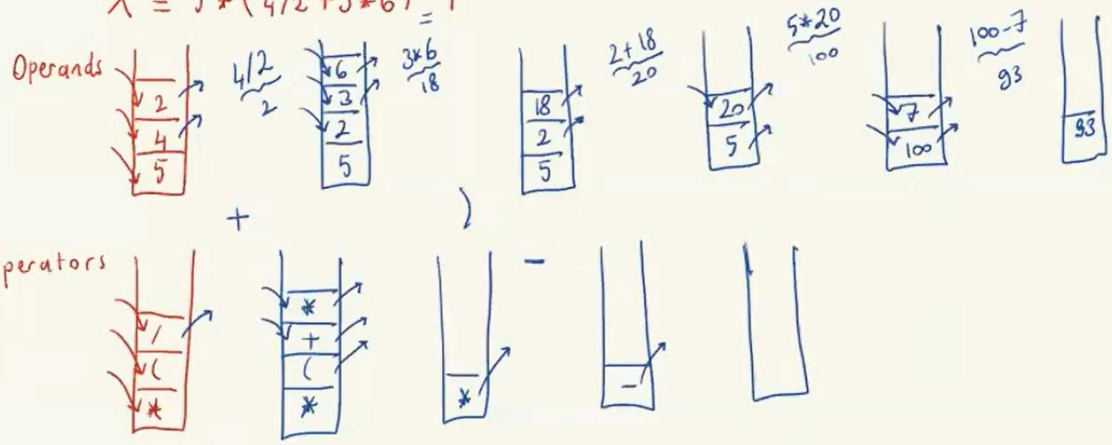


Operand

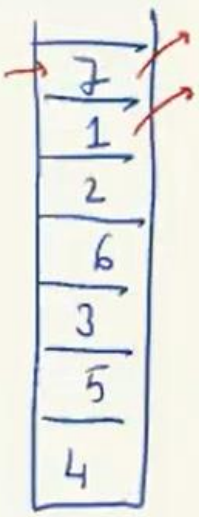


Arithmetic Operation on Compilers in Real-life

$X = 5 * (4/2 + 3 * 6) - 7$



* Finding the minimum element in a given stack



Stack 1

Stack 2

1