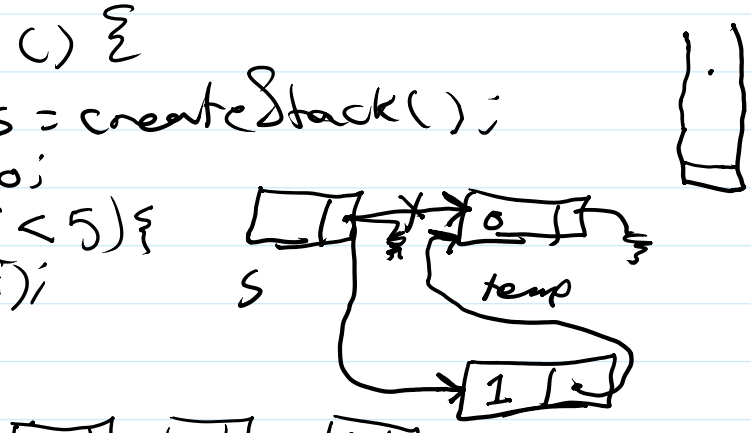
Monday, March 22, 2021 11:22 AM

Implementation of stack

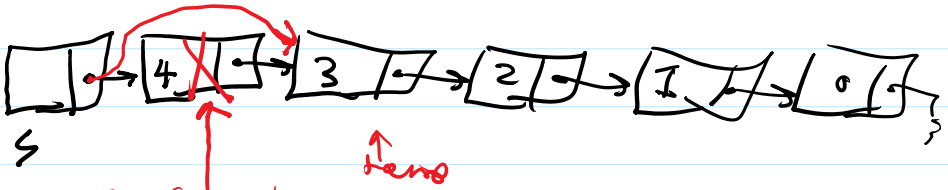
1- Pointer imp.
2- Array imp.

```
struct node;
typedef struct node* ptr;
typedef ptr stack;
struct Node {
    int Data;
    ptr next;
};

stack createStack() {
    stack s;
    s = (stack) malloc(sizeof(struct Node));
    if(s == NULL){
        printf("Error: out of memory");
        return -1;
    }
    s->next = NULL;
    return s;
}
void push(stack s, int x){
    ptr temp;
    temp = (ptr) malloc(sizeof(struct Node));
    if (temp != Null){
        temp->Data = x;
        temp->next = s->next;
        s->next = temp;
    }
    else
}
Void pop(stack s) {
    ptr temp;
```

```
  if (isEmpty(S)){
    printf("Error: do pop from empty stack");
  }else{
    temp = S->next;
    S->next = S->next->next;
    free(temp);
  }
}

int isEmpty(stack S){
  return (S->next == NULL);
}

int Top(stack S){
  if(!isEmpty(S))
    return S->next->Data;
  else
    return -1;
}
void main(){
  stack S = createStack();
  int j = 0;
  while(i<5){
    push(i);
    ++i;
  }

  pop(S);
  pop(S);
```



LIFO



S->next = S->next->next;

ptr P;

free(temp)

s->next = s->next->next;

```
void makeEmpty(stack s){
  while(!isEmpty(s))
    pop(s);
}
```

```
ptr p;
p = s;
while (p != Null){
  printf("%d", p->data);
  p = p->next;
}
```

Array Imp. of stack.

LIFO