Assignment Solutions | Stacks - 3 | Week 16

1. Baseball Game [Leetcode - 682]

Solution:

```
class Solution {
public:
    int calPoints(vector<string>& op) {

        stack<int>s;

        for(int i=0;i<op.size();i++){
            if(op[i].size() > 1 or (op[i].size() == 1 and op[i][0] >= '0'
            and op[i][0] <= '9'))s.push(stoi(op[i]));

            else if(op[i] == "C")s.pop();
            else if(op[i] == "D")s.push(2*s.top());
            else {
                int val1 = s.top();
                s.pop();
                int sum = val1 + s.top();

                s.push(val1);
                s.push(sum);
            }
        }
        int sum = 0;
        while(!s.empty()){
```

2. Remove Nodes from a Linked List [Leetcode - 2487]

Solution:

```
class Solution {
public:
    ListNode* removeNodes(ListNode* head) {
        stack<ListNode*>st;

        while(head){
            st.push(head);
            head = head->next;
        }

        ListNode *tail = st.top();
        st.pop();
        int mx = tail->val;

        while(!st.empty()){
            ListNode *top = st.top();
            st.pop();

            if(top->val >= mx){
                top->next = tail;
                tail = top;
                mx = top->val;
            }
```

3. Maximal Rectangle [Leetcode - 85]

Solution:

```
class Solution {

public:

int largestRectangleArea(vector& arr) {

int n = arr.size();

int nsi[n];

stack st;

nsi[n-1] = n;

st.push(n-1);
```

```
for(int i=n-2;i>=0;i--){
while(st.size()>0 && arr[st.top()]>=arr[i]) st.pop();
if(st.size()==0) nsi[i] = n;
else nsi[i] = st.top();
st.push(i);
}
int psi[n];
stack gt;
psi[0] = -1;
gt.push(0);
for(int i=1;i<n;i++){
while(gt.size()>0 && arr[gt.top()]>=arr[i]) gt.pop();
if(gt.size()==0) psi[i] = -1;
else psi[i] = gt.top();
gt.push(i);
}
int maxArea = 0;
for(int i=0;i<n;i++){
int height = arr[i];
int breadth = nsi[i] - psi[i] - 1;
int area = height * breadth;
maxArea = max(maxArea,area);
}
return maxArea;
}
int maximalRectangle(vector>& a) {
int n = a.size();
int m = a[0].size();
vectorrow(m , 0);
int maxArea = 0;
for(int i=0;i<n;i++){
for(int j=0;j<m;j++){
if(a[i][j] == '1')row[j] += 1;
```

```
else row[j] = 0;
}
maxArea = max(maxArea , largestRectangleArea(row));
}
return maxArea;
}
};
```