***Week – 4 (19.04.2021 – 25.04.2021)***

***RANDOM CODES***

1. ***Student Attendance Record I:***

class Solution {

public:

bool checkRecord(string s) {

return ((s.find("LLL") == string::npos) && (count(s.begin(), s.end(), 'A') < 2));

}

};

1. ZigZag Conversion:

class Solution {

public:

string convert(string s, int numRows) {

if(numRows == 1 || s.size() == 1) return s;

string res="";

int cycle = 2 \* numRows - 2, i, j;

for(i=0; i<numRows; i++)

{

for(j=0; j+i<s.size(); j = j+cycle)

{

res += s[j+i];

if(i!=0 && i!=numRows-1 && j+cycle-i<s.size())

res += s[j+cycle-i];

}

}

return res;

}

};

1. ***Valid Parentheses:***

class Solution {

public:

bool isValid(string s) {

stack <char> st;

int i;

for(i=0; i<s.size(); i++)

{

if(s[i] == '{' || s[i] == '[' || s[i] == '(') st.push(s[i]);

if(s[i] == '}' || s[i] == ']' || s[i] == ')')

{

if(st.size() == 0) return false;

else if(st.top() == '{' && s[i] == '}') st.pop();

else if(st.top() == '(' && s[i] == ')') st.pop();

else if(st.top() == '[' && s[i] == ']') st.pop();

else return false;

}

}

if(st.size() == 0)return true;

else return false;

}

};

1. ***Longest Common Prefix:***

class Solution {

public:

string longestCommonPrefix(vector<string>& strs) {

if(strs.size() == 0) return "";

if(strs.size() == 1) return strs[0];

string res="";

int i,j;

for(i=0;i<strs[0].size();i++)

{

char s = strs[0][i];

for(j=1; j<strs.size(); j++)

if(i == strs[j].length() || strs[j][i] != s) return res;

res = res + s;

}

return res;

}

};

1. ***Daily Temperatures:***

class Solution {

public:

vector<int> dailyTemperatures(vector<int>& T) {

vector<int> res;

stack<int> s;

int i;

for(i=T.size()-1; i>=0; i--)

{

while(!s.empty() && T[i] >= T[s.top()]) s.pop();

if(s.size() == 0) res.push\_back(0);

else if(T[s.top()]>T[i]) res.push\_back(s.top()-i);

s.push(i);

}

reverse(res.begin(), res.end());

return res;

}

};

1. ***Asteroid Collision:***

class Solution {

public:

vector<int> asteroidCollision(vector<int>& asteroids) {

int i;

vector<int> s;

for(i=0; i<asteroids.size(); i++)

{

if(asteroids[i]>0) s.push\_back(asteroids[i]);

else

{

while(!s.empty() && s.back()>=0 && s.back()<abs(asteroids[i])) s.pop\_back();

if(s.empty() || s.back()<0) s.push\_back(asteroids[i]);

else if(s.back()==abs(asteroids[i])) s.pop\_back();

}

}

return s;

}

};