DEQUE

1. DEQUE IN C++ STL

```
#include <iostream>
#include <cmath>
#include <bits/stdc++.h>
#include <climits>
#include <deque>
using namespace std;


int main()
{
        deque<int> dq = {10, 20, 30};

        dq.push_front(5);
        dq.push_back(50);

        for(auto x: dq)
                cout << x << " ";

        cout << dq.front() << " " << dq.back();


        return 0;
}


#include <iostream>
#include <cmath>
#include <bits/stdc++.h>
#include <climits>
```

```
#include <deque>
using namespace std;


int main()
{
	deque<int> dq = {10, 15, 30, 5, 12};

	auto it = dq.begin();

	it++;

	dq.insert(it, 20);

	dq.pop_front();
	dq.pop_back();

	cout<<dq.size();


	return 0;
}


#include <iostream>
#include <cmath>
#include <bits/stdc++.h>
#include <climits>
#include <deque>
using namespace std;


int main()
{
	deque<int> dq = {10, 20, 5, 30};
```

```
        auto it = dq.begin();

        it = dq.insert(it, 7);

        it = dq.insert(it, 2, 3);

        it = dq.erase(it + 1);

        cout << (*it) << endl;

        for(int i = 0; i < dq.size(); i++)
                cout<<dq[i]<<" ";


        return 0;
}
```

2. DESIGN A DATA STRUCTURE WITH MIN AND MAX OPERATIONS

```
#include <iostream>
#include <cmath>
#include <bits/stdc++.h>
#include <climits>
#include <deque>
using namespace std;

struct MyDS{

   deque<int> dq;
```

```
void insertMin(int x)
{

    dq.push_front(x);
}

void insertMax(int x)
{

    dq.push_back(x);
}

int getMin()
{

    return dq.front();
}

int getMax()
{

    return dq.back();
}

int extractMin()
{
    int x = dq.front();

    dq.pop_front();

    return x;
}

int extractMax()
{
```

```
        int x = dq.back();

        dq.pop_back();

        return x;
    }
};

int main()
{
        struct MyDS ds;

        ds.insertMin(10);
        ds.insertMax(15);
        ds.insertMin(5);

        int x= ds.extractMin();
        cout << x << endl;

        x= ds.extractMax();
        cout << x << endl;

        ds.insertMin(8);

        return 0;
}
```

3. MAXIMUM OF ALL SUBARRAYS OF SIZE K

```
#include <iostream>
#include <cmath>
#include <bits/stdc++.h>
#include <climits>
#include <deque>
```

```
using namespace std;


void printMax(int arr[], int n, int k)
{

    deque<int> dq;



    for (int i = 0; i < k; ++i) {

        while (!dq.empty() && arr[i] >= arr[dq.back()])
            dq.pop_back();


        dq.push_back(i);
    }


    for (int i=k; i < n; ++i) {

        cout << arr[dq.front()] << " ";


        while ((!dq.empty()) && dq.front() <= i - k)
            dq.pop_front();


        while ((!dq.empty()) && arr[i] >= arr[dq.back()])
            dq.pop_back();


        dq.push_back(i);
    }
```

```
    cout << arr[dq.front()];
}

int main()
{
        int arr[] = { 20, 40, 30, 10, 60}, n = 5;
    int k = 3;

    printMax(arr, n, k);

        return 0;
}
```

4.