s= Eeketr; k=2

Ketr

reeeeeeeeeeeeeke, k =2

```
// Brute force solution
/*



F (string input_string)
{
Set store_unique_substr;
for(i : 0 to n - 1)
        for(j : 0 to n - 1)
                String tmp = input_string.substr(i, j - i + 1);
                store_unique_substr.push(tmp);


for(string &s: store_unique_substr)
        if(isValid(s) == true)
        {
                cout<<s<<endl;
                break;
        }


*/
```

Input = r begin->eeeeeeeeeeeeeke<-high

```
String findLongestSubstring(int K, string s)
{
        Int size = s.size();
        Int end = 0, begin = 0;

        unordered_set<char> window;
        vector<int> frequency(26, 0);

        for(int low = 0, high = 0; high < size; high++)
        {
                // window = r e
                // frequency = r =  1, e = 2


                window.insert(s[high]);
                Frequency[s[high] - ‘a’]++;

                while(window.size() > k)
                {
                        if(--freq[s[low]] == 0)
                        {
                                window.erase(s[low]);
                        }

                        low++;
                }

                if(end - begin < high - low)
                {
                        End = high;
                        Begin = low;
                }
        }

        String ans = s.substr(begin, end - begin + 1);
        Return ans;

}
```