}

Solution #1

```
include <iostream>
using namespace std;

void printSorted(int arr[], int start, int end)
{
   if(start > end)
       return;

   printSorted(arr, start*2 + 1, end);



   cout << arr[start] << " ";

   printSorted(arr, start*2 + 2, end);

}



int minValue(struct node* node)
{
struct node* current = node;



while (current->left != NULL)
{
   current = current->left;
}
return(current->data);

}



int main()

{

   int arr[] = {4, 2, 5, 1, 3};

   int arr_size = sizeof(arr)/sizeof(int);

   printSorted(arr, 0, arr_size-1);

   getchar();




struct node* root = NULL;

root = insert(root, 4);

insert(root, 2);

insert(root, 5);

insert(root, 1);

insert(root, 3);


cout << "\n Minimum value in BST is " << minValue(root);

getchar();


return 0;


   return 0;

}
```

Solution #2

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

using namespace std;



bool isPalindromeRec(char str[], int a, int b)
{

    if (a == b)
    {
        return true;
    }

    if (str[a] != str[b])
    {
        return false;
    }

    if (a < b+1)
    {
        return isPalindromeRec(str, a+1, b-1);
    }

    return true;
}
```

```
bool isPalindrome(char str[])
{
  int n = strlen(str);

  if (n == 0)
  {
    return true;
  }

  return isPalindromeRec(str, 0, n-1);
}


int main()
{
   char str[] = "racecar";
   if (isPalindrome(str))
   {
     cout<<"Given input string is palindrome"<<endl;;
   }
   else
   {
      cout<<"Input string is not palindrome"<<endl;
   }
   return 0;
}
```

Solution #3

```
#include <iostream>

#include "unsortedtype.h"

#include "unsortedtype.cpp"


bool checkPallindrome()


{

    UnsortedType<char> p,q;

    char temp;

    for(int i=0 ; i < 5; i++){

        cin>>temp;

        p.InsertItem(temp);

    }

    for(int i=0;  i < p.LengthIs(); i++){

        p.GetNextItem(temp);

        q.InsertItem(temp);

    }

    p.ResetList();

    char m,n;

    for(int i=0; i < q.LengthIs(); i++){


        p.GetNextItem(m);

        q.GetNextItem(n);

        if(m!=n) return false;

    }

    return true;

}


int main(){
    if(checkPallindrome())


        cout<<"character sequence is pallindrome."<<endl;

    else


    cout<<"Not pallindrome"<<endl;

}
```