OOP find values and swap values / sorting in an array and list string

#include "stdafx.h"

#include <iostream>

using namespace std;

#include <list>

#include <algorithm>

int \_tmain(int argc, \_TCHAR\* argv[])

{

const int SIZE = 4;

int a[SIZE] = {2,6,4,8};

list<int>values,othervalues;

char name;

list<char>firstName;

values.push\_front(1);

values.push\_front(2);

values.push\_back(4);

values.push\_back(3);

cout<<"values contains: ";

printList(values);

values.sort();

cout << "\n Values after sorting contains:";

printList(values);

otherValues.insert(otherValues.begin(),a, a+SIZE);

cout <<"\n otherValues contains: ";

values.splice(value.end(),otherValues);

cout << "\n after splice values contains: ";

printList(values);

values.sort();

cout <<"\n values contains";

printList(values);

otherValues.insert(otherValues.begin(),a, a + SIZE);

otherValues.sort();

cout <<"\n OTher Values contains: ";

printList(otherValues);

values.merge(otherValues);

values.swap(otherValues);

cout << "\n Values contains";

printList(values);

values.remove(4);

cout<<"\n After remove(4)values contains:";

printList(values);

cout << "\n";

return 0;

}

template <class T>

void printList(const std::list<T> &listRef)

{

if(listRef.empty())

{

cout <<"list is empty";

}

else

{

for(auto it - listRef.begin();it != listRef.end();it++)

{

cout << " " << \*it;

}

}

}

#include "stdafx.h"

#include "students.h"

students::students(void)

{

}

students::~students(void)

{

}

void removeSpaces(char \*str)

{

// To keep track of non-space character count

int count = 0;

// Traverse the given string. If current character

// is not space, then place it at index 'count++'

for (int i = 0; str[i]; i++)

if (str[i] != ' ')

{

str[count++] = str[i]; // here count is

} // incremented

str[count] = 0;

}

#pragma once

class students

{

public:

students(void);

void removeSpaces();

~students(void);

private:

};