Project 4

by

Paola Socorro

CSCI110

Professor Horia Pop

Online files location: <https://github.com/ZenRumi/Project4>

/\*

Program: sort.cpp

Author: Paola Socorro

Project: Project 4

Date: May 27, 2014

\*/

#include <iostream>

#include <stdlib.h> /\* srand, rand \*/

#include <time.h> /\* time \*/

/\* Sources:

http://www.cplusplus.com/reference/cstdlib/rand/

\*/

using namespace std;

void bubbleSort(int[],int);

void selectionSort(int[],int);

int main()

{

srand (time(0));

int const S = 25;

int bubbleNums[S];

int selNums[S];

cout<<"Unsorted Numbers"<<endl;

for(int i=0; i< S;i++)

{

bubbleNums[i] = rand()%100+1;

selNums[i]= bubbleNums[i];

cout<<bubbleNums[i]<<" ";

//cout<<selNums[i]<<" ";

}

cout<<endl;

bubbleSort(bubbleNums,S);

selectionSort(selNums,S);

cout<<endl;

cout<<"\n";

cout<<"Bubble Sort Algorithm"<<endl;

for(int i=0; i< S;i++)

{

cout<<bubbleNums[i]<<" ";

//cout<<selNums[i]<<" ";

}

cout<<endl;

cout<<"\n";

cout<<"Selection Sort Algorithm"<<endl;

for(int i=0; i< S;i++)

{

//cout<<bubbleNums[i]<<" ";

cout<<selNums[i]<<" ";

}

cout<<endl;

cout<<"\n";

return 0;

}

void bubbleSort(int barr[], int n)

{

bool pass = true;

int temp;

int count=0;

while (pass)

{

pass=false;

count++;

for(int i=0; i<n-count;i++)

{

if(barr[i]>barr[i+1]){

temp=barr[i];

barr[i]=barr[i+1];

barr[i+1]=temp;

pass=true;

}}}}

void selectionSort(int sarr[],int n)

{

int temp=0;

for(int i=0; i<n;i++)

{

for(int j=0; j<n;j++)

{

if(sarr[i]<sarr[j])

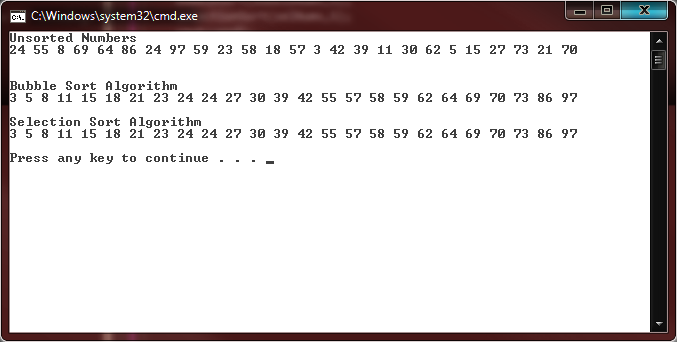
{

temp=sarr[i];

sarr[i]=sarr[j];

sarr[j]=temp;

}}}}



/\*

Program: convert.cpp

Author: Paola Socorro

Project: Project 4

Date: May 29, 2014

\*/

//I wasn't sure if you meant this or a windows form with menus and buttons.

//I figured it had to mean this since windows forms hasn't been taught in class, though i now it from a previous c++ class.

#include <iostream>

using namespace std;

float milestoK(float);

float FtoC(float);

void main()

{

int select=-1;

int sel=-1;

float data=0, result=0;

//1: Miles to Kilometers\_\_\_or\_\_\_2: Fahrenheit to Celsius -> "

cout<<"Welcome to the Conversion program."<<endl;

while (select !=0 && sel!=0)

{

cout<<"Enter 0: to Exit\_\_\_1: for OPTIONS ";

cin>>select;

if(select ==1)

{

cout<<"\n";

cout<<"1: MILES"<<endl;

cout<<"2: DEGF"<<endl;

cout<<"0: EXIT ";

cin>>sel;

cout<<"\n";

if(sel==1){

cout<<"Enter the MILES to convert to Kilometers ";

cin>>data;

result=milestoK(data);

cout<<data<<" Miles is "<<result<<" Kilometers"<<endl;

cout<<"\n";

}

if(sel==2){

cout<<"Enter the Fahrenheits to convert to Celsius ";

cin>>data;

result=FtoC(data);

cout<<data<<" Fahrenheit is "<<result<<" Celsius"<<endl;

cout<<"\n";

}

//else

//cout<<"Please Enter 0,1 or 2 as your selection"<<endl;

}

else

cout<<"Please Enter 0,1 or 2 as your selection"<<endl;

}

}

float milestoK(float m)

{

int k=0;

k=m/0.62137;

return k;

}

float FtoC(float f)

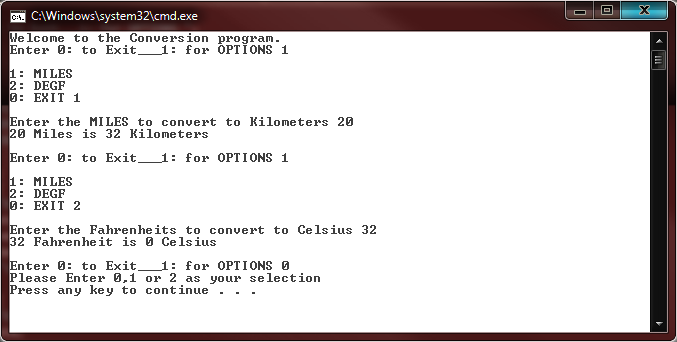
{

int c=0;

c= 5\*(f-32)/9;

return c;

}



/\*

Program: wordPro.cpp

Author: Paola Socorro

Project: Project 4

Date: May 29, 2014

\*/

#include <iostream>

#include <string>

using namespace std;

string capsFix(string);

void main()

{

bool end=false;

string data,fixed;

cout<<"Enter a string to fix"<<endl;

while(end==false)

{

getline(cin,data);

fixed=capsFix(data);

cout<<fixed<<endl;

cout<<'\n';

cout<<"Enter another string"<<endl;

cout<<'\n';

}

}

string capsFix(string str)

{

for(int i=1;i<str.length();i++)

{

if(str[i]==' ')

str[i]==' ';

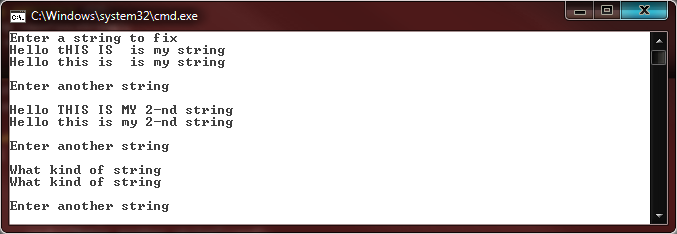
else

str[i]=tolower(str[i]);

}

return str;

}



/\*

Program: filemerge.cpp

Author: Paola Socorro

Project: Project 4

Date: May 28, 2014

\*/

#include <iostream>

#include <string>

#include <fstream>

#include <ctype.h>

/\* Sources:

http://www.cplusplus.com/

\*/

using namespace std;

void mergeData(ifstream&,ifstream&,ifstream&,ofstream&);

string removeSpace(string);

//int countWords(string); //tried with a string ended up using ifstream instead.

void countWords(ifstream&);

void wordSize(string);

int wordCount;

string longWord;

string shortWord;

int main()

{

ifstream data;

ifstream data2;

ifstream data3;

ofstream outdata;

ifstream endFile;

string line,file1,file2,file3,file4;

cout<<"Enter names of files to merge. DO NOT add .txt extension."<<endl;

cout<<"Enter each name in a new line"<<endl;

cin>>file1;

cin>>file2;

cin>>file3;

data.open(file1 +".txt");

if(data.fail()){return 0;}

data2.open(file2 +".txt");

if(data2.fail()){return 0;}

data3.open(file3 +".txt");

if(data3.fail()){return 0;}

outdata.open("output.txt");

if(outdata.fail()){return 0;}

mergeData(data,data2,data3,outdata); //Merging files together into a third.

data.close();

data2.close();

data3.close();

outdata.close();

endFile.open("output.txt");

if(endFile.fail()){return 0;}

countWords(endFile);

cout<<"Word Count is: "<<wordCount<<endl;

cout<<"Longest Word: "<<longWord<<endl;

cout<<"Shortest Word: "<<shortWord<<endl;

endFile.close();

}

void mergeData(ifstream&data,ifstream&data2,ifstream&data3,ofstream&outdata)

{

string line1,line2,line3,merge;

if (data.is\_open() || data2.is\_open() || data3.is\_open() )

{

getline(data,line1);

//cout<<line1<<endl;

getline(data2,line2);

//cout<<line2<<endl;

getline(data3,line3);

//cout<<line2<<endl;

merge=line1+" "+line2+" "+line3;

merge=removeSpace(merge);

outdata<<merge;

}

else cout << "Unable to open file";

}

string removeSpace(string data) //working inside mergeData function.

{

for(unsigned int i=0; i<data.length();i++)

{ for(unsigned int j=0;j<data.length();j++)

{

if(data.at(j)==' ' && data.at(j+1)==' ')

data.erase(j+1,1);}}

return data;}

void countWords(ifstream& data)

{ string word;

while(!data.eof()){

data>>word;

wordCount++;

wordSize(word);

}}

void wordSize(string data)

{

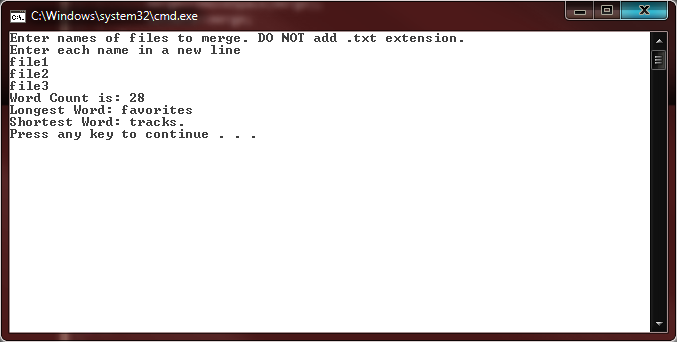
if(data.size()<longWord.size())

shortWord=data;

if(data.size()>shortWord.size())

longWord=data;

}



/\*

Program: binsearch.cpp

Author: Paola Socorro

Project: Project 4

Date: May 27, 2014

\*/

#include <iostream>

#include <stdlib.h>

#include <time.h>

/\* Sources:

key equal to 8

b) key equal to 81

c) key equal to 82

\*/

using namespace std;

int binSearch(int[],int,int,int);//array, lowerbound, upperbound, key.

void bubbleSort(int[],int); //bubble sort from question 1.

int main()

{

int const S =27;

int numbers [S] = {12,30,33,81,99,85,28,24,25,24,26,27,9,41,32,35,23,39,43,41,46,56,62,8,20,22,18};

int pos;

int key;

bubbleSort(numbers,S);

for(int i=0; i< S;i++)

{

cout<<numbers[i]<<" ";

//cout<<selNums[i]<<" ";

}

cout<<"\n";

cout<<"Enter a number to find ";

cin>>key;

pos=binSearch(numbers,0,S,key);

if(pos==-1)

cout<<"Number not found"<<endl;

else{

cout<<"\n";

cout<<"Found "<<key<<" at position "<<pos<<endl;

}

}

int binSearch(int arr[],int lower,int upper,int key){

if(lower<=upper){

int pos=(lower+upper)/2;

if(key==arr[pos])

return pos;

else if (key<arr[pos])

return binSearch(arr,lower,pos-1,key);

else

return binSearch(arr,pos+1,upper,key);

}

return -1;

}

void bubbleSort(int barr[], int n)

{

bool pass = true;

int temp;

int count=0;

while (pass)

{ pass=false;

count++;

for(int i=0; i<n-count;i++)

{

if(barr[i]>barr[i+1]){

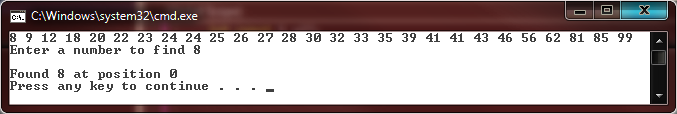
temp=barr[i];

barr[i]=barr[i+1];

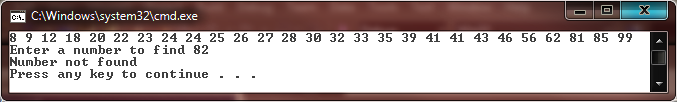
barr[i+1]=temp;

pass=true;

}}}}







COMMENTARY

by Paola S.

1. Number one was not particularly difficult. Had to go read up a bit on stdlib because I keep forgetting where to find rand. Applying bubble sort was pretty straight forward with one for loop and an if statement inside. Selection sort took me a bit to get right. At first only the first five numbers were sorted while it skipped the rest of the numbers. Later it didn't seem to skip them and made an attempt at sorting but it still wasn't completely sorted. I figured it was because one of the for loops was stopping too soon, therefore leaving the list fairly unsorted still. Inside the if statement I had also made a mistake, when it came to changing the numbers and using a temporary variable. Turned out I was putting the wrong value in there. After that it came together rather nicely.

2. This one confused me slightly because I have taken previous C++ and we went much further than the material covered in this class. We went over windows forms , which I have also used in a C# class. Originally I thought that's what you meant, and actually starting coding a windows forms. Not hard at all really, but later realized that probably couldn't be since it wasn't covered in class. I ended up scrapping that code entirely and rewriting a console type menu of sorts. For once I wrote all the code in about 15 minutes, without having to google anything and without getting an error. (Okay, i google how to convert from miles to Km.) I'm still sure a windows form would have been so much nicer.

3. Here I used a string as my input. Created a function that takes this string and converts all letters in the string to lowercase, except the first letter in the string. The strings were entered one at a time into the console and read one at a time. I imagine that when using a file to check for this error, I would have to add an extra layer of code. This might be another function that checks for the beginning of each sentence by looking for a newline or a dot. Then the capsFix function i created would be used inside of it.

4. This was a fun one to write and figure out. Used StackExchange and google quite a bit but the code is all my own. Hence its messy, and there was probably a better way to write that. While writing c (word count) I came across a problem I'm still not sure how to fix. It counts the words in the file properly, except if there is a space at the end of the last word, it counts n words+1. So if its 20 words and a space, it will say 21 words. I had to leave that alone and concentrate on the rest, but I'm still not sure how to change that. In my mind I'm thinking there should be a way to check for the end of file, and then ask -is the spot before endoffile a space? if so do not count towards word count.

I also found out you can pass ifstream and ofstream data into a function. That was pretty nice and it made the process that much easier. As always I was looking for the most complicated way to solve a problem. My logic is always that if its difficult, it's probably smart and right. It's a bad habit of mine.

For d, I created a function that compares the size of each word to each other. This function is inside the countWords function.

5. For this one I used part of the code from problem 1, to sort the array given in 1b. I hardcoded the array in, but it would have just as easy to generate a random array of 27 like the one in 1b. I used the bubble sort to sort the array before implement the binary search. Here I chose to go with a recursive function which I understood. So far I can understand recursive functions of they are easy like this one, and generally clear.

All code is my own. I hope it's not too messy and crude. :D

Chances are I might not see you next semester Mr. Pop. I'm in the process of joining the U.S Navy and I might ship out before the next semester. If I don't see you, know that I have greatly enjoyed your class. Specially the binary and hex conversions at the very beginning. That part alone will be extremely useful in my future career in the Navy.