# Lab 10

Problem: Create recursive solutions to: creating strlen and strcpy, creating pow() function, finding the sum of a linked list, finding a target in a linked list.

## Source Files:

(strlen, strcpy, and pow test bed)

#include <iostream>

using namespace std;

int strlen(char\* string);

void strcpy(char\* destination, char\* source);

double power(double base,int exponent);

int main()

{

char a[20]={"Here we have it!"};

cout<<"The count of characters in string A is: ";

int i;

i=strlen(a);

cout<<i<<" characters."<<endl;

cout<<"Copying the string to a new string...\n\n";

char b[20];

strcpy(b,a);

cout<<"Here's the copy!: "<<b<<endl;

cout<<"What number would you like to raise to a power? ";

double number;

cin>>number;

cout<<"What power would you like to raise this number to? ";

int exponent;

cin>>exponent;

double result=power(number,exponent);

cout<<"Here's the result: "<<result<<endl<<endl;

return 0;

}

int strlen(char\* string)

{

int n=0;

if(string[0]!=NULL)

return 1+strlen(string+1);

else

return n;

}

void strcpy(char\* destination,char\* source)

{

if(source[0]=='\0')

destination[0]=NULL;

else

{

destination[0]=source[0];

strcpy(destination+1,source+1);

}

}

double power(double base,int exponent)

{

exponent--;

return base\*pow(base,exponent--);

}

(linked list test bed)

main.cpp:

#include "list.h"

int main()

{

list l1;

cout<<"Filling list one with 0-10!";

for(int i=0;i<11;i++)

l1.push\_back(i);

cout<<"\nAdding up all of the nodes!";

int sum;

sum=l1.add\_ints(l1.getfront());

cout<<"\nThe sum of all of the nodes is "<<sum<<endl;

cout<<"What value would you like to find, recursively? ";

int target;

cin>>target;

node\* location;

location=l1.find(l1.getfront(),target);

if(location!=NULL)

cout<<"The target is at node with value "<<location->getdata()<<endl<<endl;

else

cout<<"The target does not exist in this list.\n\n";

return 0;

}

list.cpp(only displaying the functions pertinent to the main):

int list::add\_ints(node \* p)

{

if(p==NULL)

return 0;

return p->getdata()+add\_ints(p->getnext());

}

node\* list::find(node \* p,int target)

{

if(p==NULL)

return NULL;

if(p->getdata()==target)

return p;

else

{

if(p->getnext()!=NULL)

return find(p->getnext(),target);

else

return NULL;

}

}

## Sample Runs:

The count of characters in string A is: 14 characters.

Copying the string to a new string...

Here's the copy!: Banana Montana

What number would you like to raise to a power? 5.5

What power would you like to raise this number to? 4

Here's the result: 915.063

Press any key to continue . . .

(with NULL node pointers):

Filling list one with 0-10!

Adding up all of the nodes!

The sum of all of the nodes is 55

What value would you like to find, recursively? 2

The target does not exist in this list.

(where find cannot find the target):

Filling list one with 0-10!

Adding up all of the nodes!

The sum of all of the nodes is 55

What value would you like to find, recursively? 11

The target does not exist in this list.

(where find can find the target):

Filling list one with 0-10!

Adding up all of the nodes!

The sum of all of the nodes is 55

What value would you like to find, recursively? 4

The target is at node with value 4