Talen Phillips

EE 107-01

Homework 8

Due 22 Apr 13

1) What does the following program print?

#include <stdio.h>

int main(void) {

char s[]=”Hsjodi”, \*p;

for (p=s; \*p; p++)

--\*p;

puts(s);

return 0;

}

**Answer: the program prints "Grinch".**

I could just copy-paste the program and run it, but I'll provide comments. It's a little easier to see what's happening (and provide comments) if I rewrite it:

#include <stdio.h>

int main(void){

char s[]="Hsjodi";// Initializing the array. (s[0]='G'+1, etc...)

char \*p; // Initializing pointer.

p=s; // References pointer to first element of array.

while (\*p){ // Stops loop when it sees the null character (\*p)--; // Subtracts 1 from the value at address: \*p.

p++; // Points \*p to the next address.

}

puts(s); // Prints contents of the s array to screen

return 0;

}

2) Write a program named *sum.c* that adds up its command-line arguments, which are assumed to be integers. For example, running the program by typing

./sum.exe 8 24 62

Should produce the following output:

Total: 94

/\* Talen Phillips

\* EE107-01

\* Due: 22APR2013

\* "Argument adder"

\*/

#include <stdio.h>

/\* declaring argument count and array as discussed in class. \*/

int main (int argc, char \*argv[]){

int x=0, n; /\* x is the total, and n is the loop counter. \*/

/\* There is always one argument before the user entered ones. \*/

if (argc<2){

printf("No arguments entered.");

}

else {

/\* stop adding when n reaches the total number of argument. \*/

for (n=1;n<=argc;++n)

x+=atoi(argv[n]);

/\* atoi turns each row of the array into an integer. \*/

printf("%d",x); /\* display the result.\*/

}

return x; /\* also return the result. \*/

}

Note: conio.h and getch() were used to hold the windows open for a screenshot, but they were not part of the final code.

