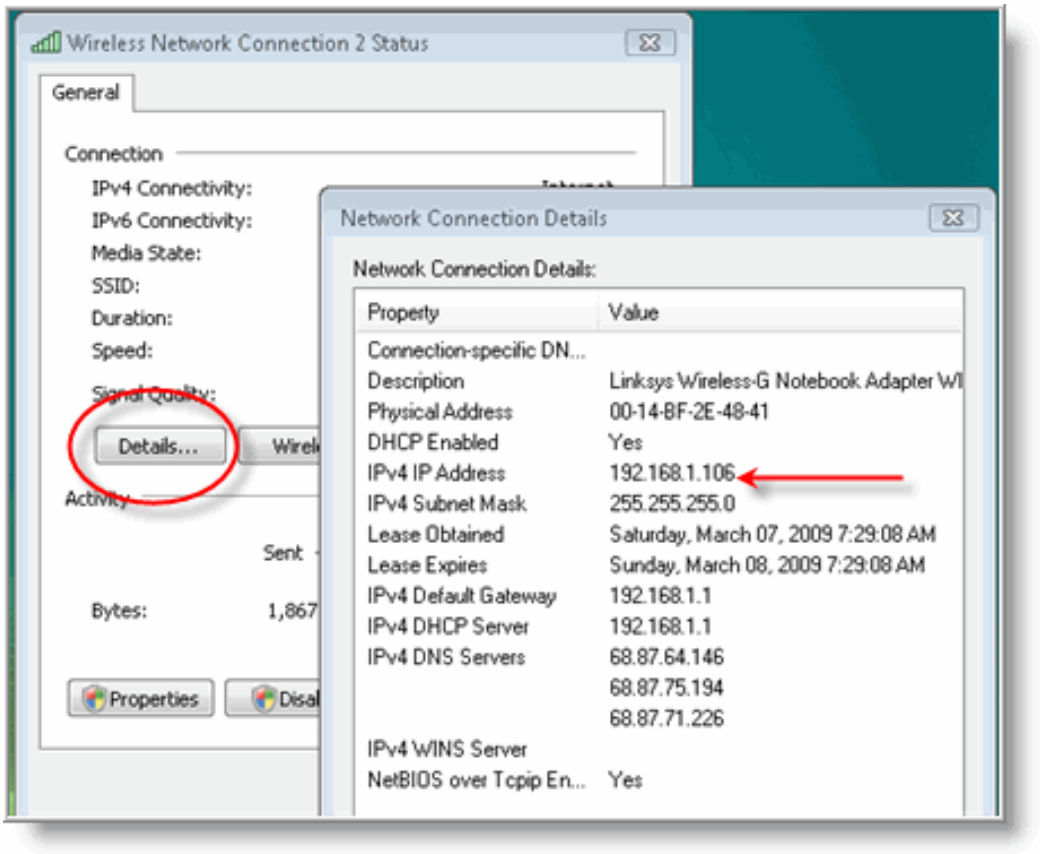
R-j-ascii-program-translator.epub



DOWNLOAD: https://byltly.com/2ivjst

Download

Q: How to use a structure as a function pointer parameter? In the following code, main() calls function add() and function number() and expects the returned values to be added. The function add() calls its body func and return the sum. But the value returned is 4... My questions are: Why is this? I had expected that the value returned would be the sum of the two numbers given. I tried the same code with ints and it returned the correct answer. #include int add(int num1, int num2){ return num1+num2; } int number(int *num1, int *num2){ *num1 = 1; *num2 = 1; return *num1 + *num2; int main() { int num1 = 0; int num2 = 0; add(num1, num2); printf("The sum is : %d ",number(&num1,&num2)); return 0; A: The & is needed to dereference a pointer. In other words, &num1 is not a pointer to a int, it's a pointer to a pointer to an int. The & in C is similar to the keyword this in C#. In the code above, the &num1 is not really the address of the int, it's the address of a pointer to the int. You're making main a function pointer, that's not necessary and is unnecessary. I would do it like this, and that should work just fine: #include int add(int num1, int num2) { return num1 + num2; int number(int *num1, int *num2) { ", number(& 82157476af