实验0

练习1

- 修改
  - V0=3
  - V1=3
  - V2=1
  - V3=3

练习2

1. How do you pass command line arguments to a program when using gdb?
   A：Add them after "run" or use "set args" command. Use "show args" command can see the arguments.
2. How do you set a breakpoint which only occurs when a set of conditions is true (e.g. when certain variables are a certain value)?
   A：Use "break" to add breakpoint. Add "if" and the conditions after the "break" command.
3. How do you execute the next line of C code in the program after stopping at a breakpoint?
   A：Use "next" command.
4. If the next line of code is a function call, you'll execute the whole function call at once if you use your answer to #3. How do you tell GDB that you want to debug the code inside the function instead?
   A：Use "step" command.
5. How do you resume the program after stopping at a breakpoint?
   A：Use "continue" command.
6. How can you see the value of a variable (or even an expression like 1+2) in gdb?
   A：Use the print command with a variable name as the argument.
7. How do you configure gdb so it prints the value of a variable after every step?
   A：Use "display" command.
8. How do you print a list of all variables and their values in the current function?
   A：Use "info locals" command
9. How do you exit out of gdb?
   A：Use "quit" command.

练习3

```
int ll_equal(const node* a, const node* b)
{
    while(a != NULL && b != NULL)
    {
        if(a->val != b->val)
        {
            return 0;
        }
        a = a->next;
        b = b->next;
    }
    return a == b;
}
```

练习4

- 前者是运行wc.c编译后的wc文件；后者是用wc命令统计wc.c中的字节数、字数、行数。

```
void wc(FILE *ofile, FILE *infile, char *inname)
{
    char buffer[1000];
    char c;
    int bufferlen;
    int isLastBlank = 0; //上个字符是否是空格(1表示是  0表示不是)
    int charCount = 0;
    int wordCount = 0;
    int lineCount = 0;
    int i;

    if(!infile)
    {
        while(gets(buffer) != NULL)
        {
            bufferlen = strlen(buffer);
            for(i = 0;i < bufferlen;i++)
            {
                c = buffer[i];
                if(c == ' ' || c == '\t')
                {
                    if(isLastBlank == 0)
                    {
                        wordCount++;
                    }
                    isLastBlank = 1;
                }
                else if(c != '\n' && c != '\r')
                {
                    charCount++;
                    isLastBlank = 0;
                }
            }
            if(isLastBlank == 0)
            {
                wordCount++;
            }
            isLastBlank = 1;
            lineCount++;
        }
    }
    else
    {
        while(fgets(buffer,1000,infile) != NULL)
        {
            bufferlen = strlen(buffer);
            for(i = 0;i < bufferlen;i++)
            {
                c = buffer[i];
                if(c == ' ' || c == '\t')
                {
                    if(isLastBlank == 0)
                    {
                        wordCount++;
                    }
                    isLastBlank = 1;
                }
                else if(c != '\n' && c != '\r')
                {
                    charCount++;
                    isLastBlank = 0;
                }
            }
            if(isLastBlank == 0)
            {
                wordCount++;
            }
            isLastBlank = 1;
            lineCount++;
        }
    }

    if(!ofile)
    {
        printf("行数：%d\n",lineCount);
        printf("字符数：%d\n",charCount);
        printf("单词数：%d\n",wordCount);
    }
    else
    {
        fprintf(ofile, "行数：%d\n",lineCount);
        fprintf(ofile, "字符数：%d\n",charCount);
        fprintf(ofile, "单词数：%d\n",wordCount);
    }

    return;
}
```