

```

Ts-MacBook-Pro-2:problem1 tmoore$ cat countWords.c
#include<stdio.h>
#include<ctype.h>
#include<string.h>

struct key
{
    char *word;
    int count;
}keytab[] =({"auto",0}, {"break",0}, {"case",0}, {"char",0}, {"const",0}, {"continue",0}, {"default",0}, {"unsigned",0}, {"while",0});

#define MAXWORD 100
#define NKEYS (sizeof keytab / sizeof(struct key))

int getword(char *, int);
int binsearch(char *,struct key*, int);

//count c keywords

int main()
{
    int n;
    char word[MAXWORD];

    while(getword(word,MAXWORD)!=EOF)
    {
        if(isalpha(word[0]))
        {
            if((n=binsearch(word,keytab,NKEYS))>=0) // NKEYS not defined
                keytab[n].count++;
        }

        for(n = 0;n<NKEYS; n++)
        {
            if(keytab[n].count>0)
                printf("%4d %s\n",keytab[n].count,keytab[n].word);
        }
    }

    return 0;
}

int getword(char *word, int lim)
{

```

```
problem1 — bash — 137x48

int getword(char *word, int lim)
{
    int c,etch(void);
    void ngetch(int);
    char *w = word;

    while(isspace(c=etch()))
        ;
    if(c!=EOF)
        *w++=c;
    if(!isalpha(c))
    {
        *w = '\0';
        return c;
    }

    for(;--lim >0; w++)
        if (!isalnum(*w=etch()))
        {
            ngetch(*w);
            break;
        }
    *w = '\0';
    return word[0];
}

int binsearch(char *word, struct key tab[],int n)
{
    int cond;
    int low,high,mid;

    low = 0;
    high = n-1;
    while(low<=high)
    {
        mid = (low+high)/2;
        if((cond=strcmp(word,tab[mid].word))<0)
            high = mid-1;
        else if (cond>0)
            low = mid+1;
        else
            return mid;
    }
    return -1;
}

Ts-MacBook-Pro-2:problem1 twoore$ ./s.out
while
1 while
while
2 while
while
3 while
auto
1 auto
3 while
auto
2 auto
3 while
```

The difference between the two is that the second example returns the pointer and manipulates the array with a pointer and the first example manipulates the array with a variable n .