Ryan Reynolds
2693018
CIS 340

Homework 3

Question 1:

Output:

```
ryan@ryan-VirtualBox: ~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1
 3 has type ‘int’ [-Wformat=]
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1$ gcc -o
 Q1.out Q1.c
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1$ ./Q1.o
ut input.txt
key to search for:ing

Key:ing found in word:ring
1 found so far Key(s):ing

Key:ing found in word:string
2 found so far Key(s):ing

Key:ing found in word:ring
3 found so far Key(s):ing

Key:ing found in word:ding
4 found so far Key(s):ing

Key:ing found in word:srtroing
5 found so far Key(s):ing

Key:ing found in word:ring
6 found so far Key(s):ing

Key:ing found in word:string
7 found so far Key(s):ing

Key:ing found in word:tring
8 found so far Key(s):ing

Key:ing found in word:ing
9 found so far Key(s):ing
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1$
```

Input File:

```
input.txt (~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1) - gedit
Open                                                              Save
Q1.c                                                  input.txt
the ring is a
string ring
ding srtroing

sadsa
cas'c


ring string
tring ing

asds
```

Source:

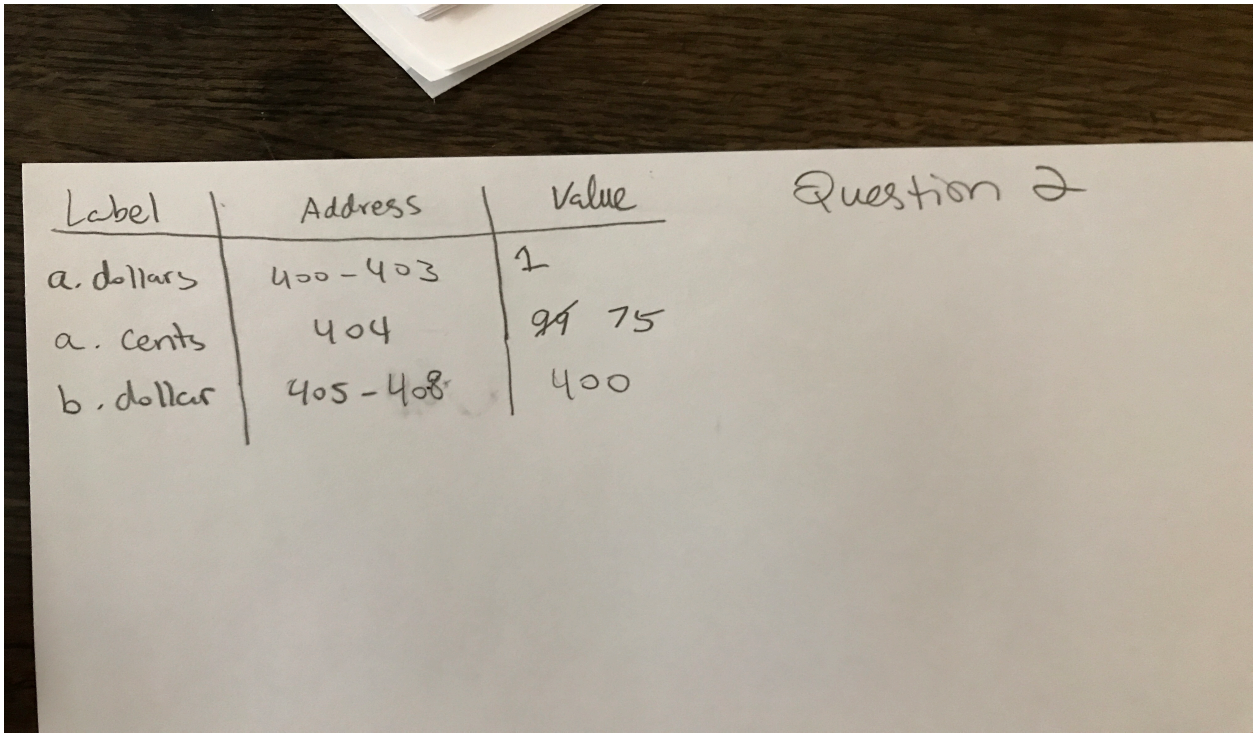
Q1.c (~/Desktop/CIS 340/Homeworks/Reynolds_HW3/Question 1) - gedit

Open Save

Q1.c input.txt

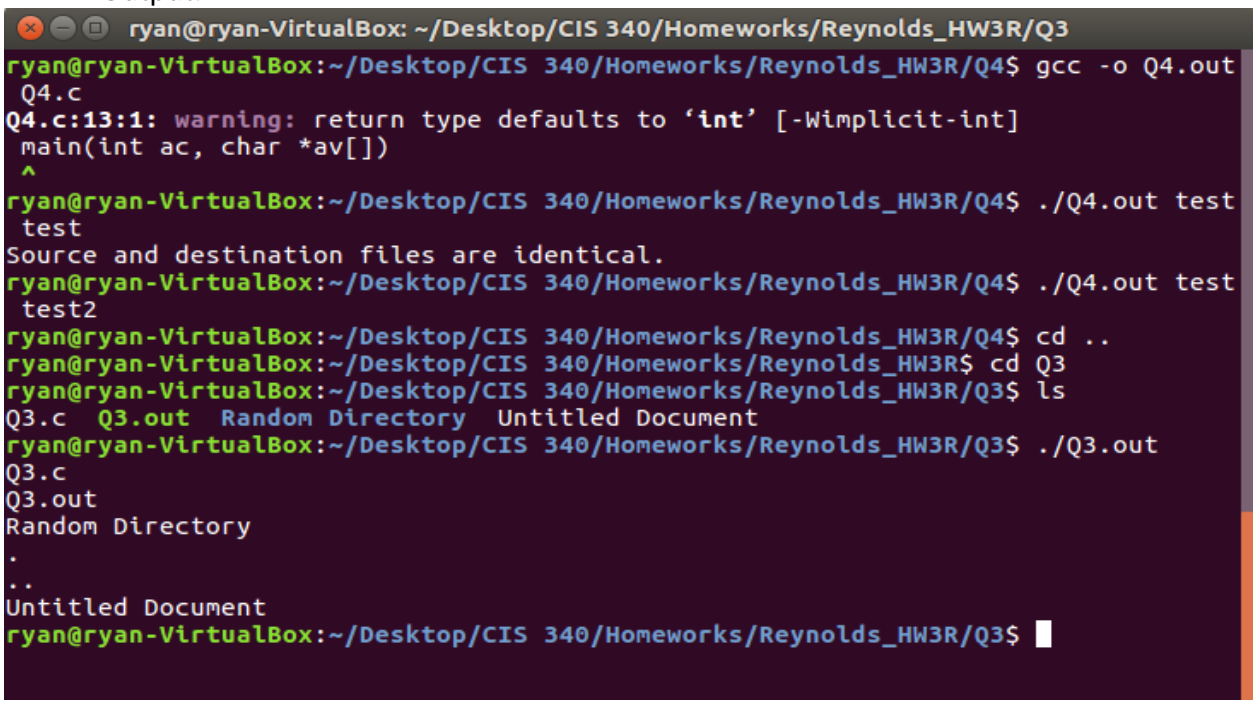
```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
int main (int argc, char *argv[])
{
FILE *fpt;
int j, count=0;
char word[80], key[80];

if(argc !=2)
{
     printf("key[filename]\n");
     exit(0);
}
if((fpt=fopen(argv[1],"r")) == NULL)
{
     printf("unable to open %s for reading\n",argv[1]);
     exit(0);
}
printf("key to search for:");
scanf("%s",key);

while (fscanf (fpt,"%s",word) == 1)
{
          int i, j=0, k;
            for(i=0; word[i]; i++)
            {
              if(word[i] == key[j])
              {
                for(k=i, j=0; word[k] && key[j]; j++, k++)
                  if(word[k]!=key[j])
                      break;
                 if(!key[j]){
                  count++;
                  printf("\nKey:%s found in word:%s",key, word);
                  printf("\n%i found so far Key(s):%s\n", count, key);

                  }
              }
            }

}
}
```

Question 2:

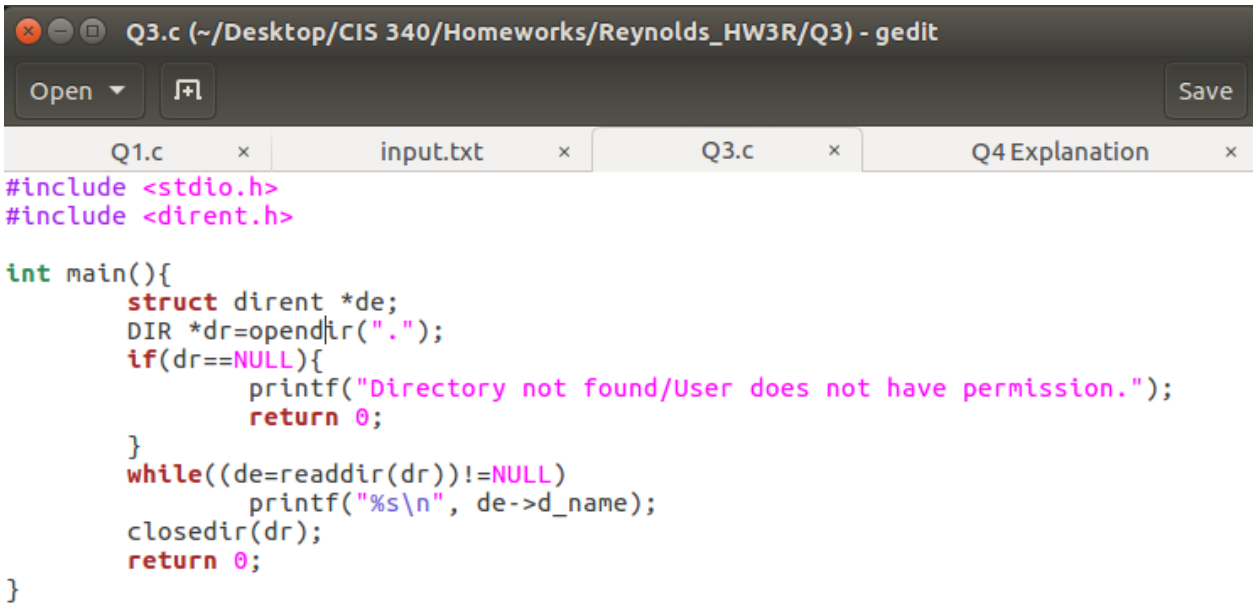


Question 3:

Output:

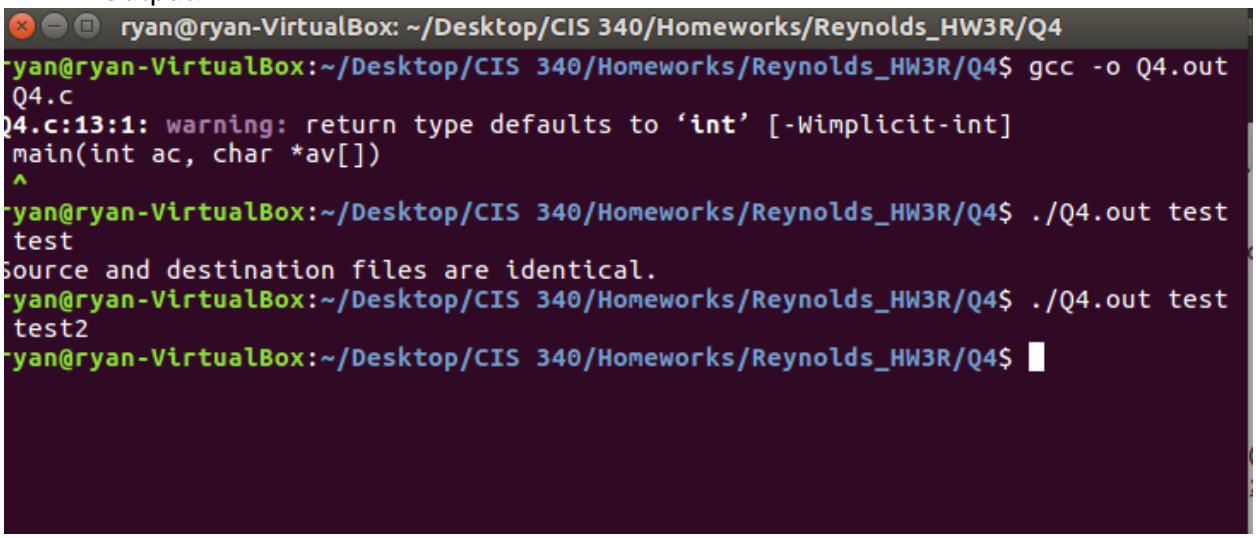


Source:



Question 4:

Output:



Source:

Q4.c (~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q4) - gedit

Open Save

Q1.c × input.txt × Q3.c × Q4 Explanation × Q4.c × test2 × test ×

```
#include        <stdio.h>
#include        <unistd.h>
#include        <fcntl.h>
#include        <stdlib.h>
#include        <string.h>

#define BUFFERSIZE      4096
#define COPYMODE        0644

void oops(char *, char *);

main(int ac, char *av[])
{
        int     in_fd, out_fd, n_chars;
        char    buf[BUFFERSIZE], buf1[BUFFERSIZE], buf2[BUFFERSIZE];

        strcpy(buf1,av[2]);
        strcpy(buf2,av[1]);

        if(strcmp(av[1],av[2])!=0)
        {

                if (strcmp(av[1],buf2)==0){
                        strcat(buf2, "(1)");


                }
                                                /* check args   */
                if ( ac != 3 ){
                        fprintf( stderr, "usage: %s source destination\n", *av);
                        exit(1);
                }
                                                        /* open files   */

                if ( (in_fd=open(av[1], O_RDONLY)) == -1 )
                        oops("Cannot open ", av[1]);

                if ( (out_fd=creat( buf2, COPYMODE)) == -1 )
                        oops( "Cannot creat", buf2);
```

```
                                                        /* copy files    */
                while ( (n_chars = read(in_fd , buf, BUFFERSIZE)) > 0 )
                        if ( write( out_fd, buf, n_chars ) != n_chars )
                                oops("Write error to ", buf2);
                if ( n_chars == -1 )
                                oops("Read error from ", av[1]);

                                                        /* close files   */

                if ( close(in_fd) == -1 || close(out_fd) == -1 )
                        oops("Error closing files","");

        }
        else{printf("Source and destination files are identical.\n");}
}
void oops(char *s1, char *s2){
                fprintf(stderr,"Error: %s ", s1);
                perror(s2);
                exit(1);
        }
```

Question 5:

Output:

```
ryan@ryan-VirtualBox: ~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q5
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q5$ gcc -o Q5.out Q5.c
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q5$ ./Q5.out
2         0         ~         ~~           reboot   reboot   Mar 21 17:45 (4.13.0-37-generic)
1         53        ~         ~~           runlevel runlevel Mar 21 17:45 (4.13.0-37-generic)
6         1232      tty1      tty1LOGIN    LOGIN    LOGIN    Mar 21 17:45 ()
7         1363      tty7      :0           ryan     ryan     Mar 21 17:45 (:0)
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q5$
```

Source:

```
Q5.c (~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q5) - gedit
Open                                                                    Save
#include        <stdlib.h>
#include        <stdio.h>
#include        <unistd.h>
#include        <utmp.h>
#include        <fcntl.h>
#include        <time.h>


#define SHOWHOST /* include remote machine on output */
void showtime(long);
void show_info(struct utmp *);

int main() {
  struct utmp current_record; /* read info into here */
  int utmpfd;  /* read from this descriptor */
  int reclen = sizeof(current_record);

  if ((utmpfd = open(UTMP_FILE, O_RDONLY)) == -1) {
    perror( UTMP_FILE ); /* UTMP_FILE is in utmp.h */
    exit(1);
  }

  while (read(utmpfd, &current_record, reclen) == reclen) {
    show_info(&current_record);
  }
  close(utmpfd);
  return 0; /* went ok */
}
```

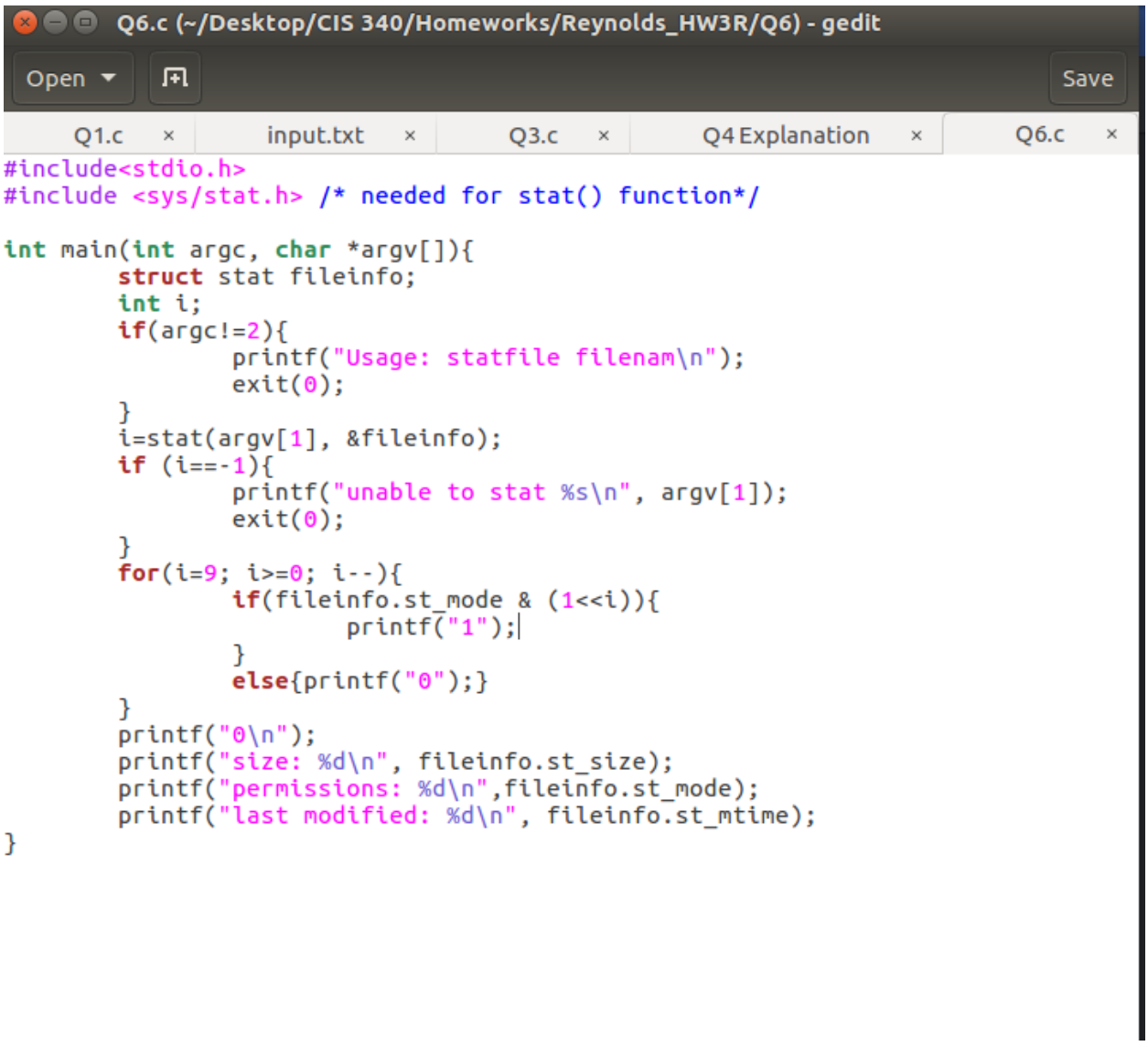
```c
/*!
 * show info()
 * displays contents of the utmp struct in human readable form
 * *note* these sizes should not be hardwired
 */
void show_info(struct utmp *utbufp) {
   printf("%-8i", utbufp->ut_type); /* Type of Log in*/
  printf(" "); /* a space */
  printf("%-8i", utbufp->ut_pid); /* Process ID*/
  printf(" "); /* a space */
  printf("%-8.8s", utbufp->ut_line); /* Devicename*/
  printf(" "); /* a space */
  printf("%-12.12s", utbufp->ut_id); /* Inittab ID */
  printf(" "); /* a space */
  printf("%-8.8s", utbufp->ut_user); /* USERNAME */
  printf(" "); /* a space */
  printf("%-8.8s", utbufp->ut_name); /* the logname */
  printf(" "); /* a space */
  showtime(utbufp->ut_time);
  printf(" "); /* a space */
#ifdef SHOWHOST
  printf("(%s)", utbufp->ut_host); /* the host */
#endif
  printf("\n"); /* newline */
}

void showtime( long timeval )
/*
 *       displays time in a format fit for human consumption
 *       uses ctime to build a string then picks parts out of it
 *       Note: %12.12s prints a string 12 chars wide and LIMITS
 *       it to 12chars.
 */
{
        char    *cp;                    /* to hold address of time      */

        cp = ctime(&timeval);           /* convert time to string       */
                                        /* string looks like            */
                                        /* Mon Feb  4 00:46:40 EST 1991 */
                                        /* 0123456789012345.            */
        printf("%12.12s", cp+4 );       /* pick 12 chars from pos 4     */
}
```

Question 6:

Output:

```
ryan@ryan-VirtualBox: ~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q6
   printf("size: %d\n", fileinfo.st_size);
          ^
Q6.c:24:10: warning: format ‘%d’ expects argument of type ‘int’, but argument 2
has type ‘__time_t {aka long int}’ [-Wformat=]
   printf("last modified: %d\n", fileinfo.st_mtime);
          ^
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q6$ gcc -o Q6.out
 Q6.c
Q6.c: In function ‘main’:
Q6.c:9:3: warning: implicit declaration of function ‘exit’ [-Wimplicit-function-
declaration]
   exit(0);
   ^
Q6.c:9:3: warning: incompatible implicit declaration of built-in function ‘exit’
Q6.c:9:3: note: include ‘<stdlib.h>’ or provide a declaration of ‘exit’
Q6.c:14:3: warning: incompatible implicit declaration of built-in function ‘exit
’
   exit(0);
   ^
Q6.c:14:3: note: include ‘<stdlib.h>’ or provide a declaration of ‘exit’
Q6.c:23:9: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 h
as type ‘__off_t {aka long int}’ [-Wformat=]
  printf("size: %d\n", fileinfo.st_size);
         ^
Q6.c:25:9: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 h
as type ‘__time_t {aka long int}’ [-Wformat=]
  printf("last modified: %d\n", fileinfo.st_mtime);
         ^
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q6$ ./Q6.out Q6.c
01101101000
size: 560
permissions: 33204
last modified: 1521732137
ryan@ryan-VirtualBox:~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q6$
```

Source:


Q6.c (~/Desktop/CIS 340/Homeworks/Reynolds_HW3R/Q6) - gedit
Open
Save
Q1.c
input.txt
Q3.c
Q4 Explanation
Q6.c
#include<stdio.h>
#include <sys/stat.h> /* needed for stat() function*/
int main(int argc, char *argv[]){
struct stat fileinfo;
int i;
if(argc!=2){
printf("Usage: statfile filenam\n");
exit(0);
}
i=stat(argv[1], &fileinfo);
if (i==-1){
printf("unable to stat %s\n", argv[1]);
exit(0);
}
for(i=9; i>=0; i--){
if(fileinfo.st_mode & (1<<i)){
printf("1");
}
else{printf("0");}
}
printf("0\n");
printf("size: %d\n", fileinfo.st_size);
printf("permissions: %d\n",fileinfo.st_mode);
printf("last modified: %d\n", fileinfo.st_mtime);
}