Homework #3
Due Mo. 10/29

Please prepare a report including the source code, captures of output and send it along with related source files as a zip file to my blackboard. The zip file should be named by your family name. Please print the report and bring the hard copy by the due date to the class. The selective grading will be used for homeworks of this course.

1. Write a program that reads a text file and prints out any words that begin with a user-given string. The filename should be given at the command line as an argument. The program should prompt the user for the search string. The program should then read the file one word at a time and print out the word if its first N bytes match the search string, where N is the length of the search string.

2. Write out the memory map for the following code, providing all values at the end of execution.

```
#include <stdio.h>

typedef struct amount {
                      int dollars;
                      char cents;
                      } money;

main()
{
money a,*b;

a.dollars=1;
a.cents=99;
b=&a;
b->cents=75;
}
```

3. Write a Program to list all files and sub-directories in a current directory.

4. What does the standard *cp* do if you try to copy a file onto itself? For example: *cp file1 file1* . What do you think is the correct action? Modify *cp1.c* to handle that situation.

5. The *who1* program lists every entry in the *utmp* file. That was not our intention but it provided a handy tool for examining the contents of *utmp*. Make *who1* more useful by adding code to it that prints out all the other fields in the struct. The *ut_type* field is particularly useful.

6. The following code prints the permission as number. Modify the code to print 10 lowest significant binary bits (e.g., "0110100100") for permission.

```
#include <stdio.h>
#include <sys/stat.h>  /* needed for stat() function */

int main(int argc, char *argv[])
{
struct stat     fileinfo;  /* returned info about file */
int             i;

if (argc != 2)
  {
  printf("Usage:  statfile filename\n");
  exit(0);
  i=stat(argv[1],&fileinfo);
  if (i == -1)
    {
    printf("Unable to stat %s\n",argv[1]);
    exit(0);
    }
  printf("size: %d\n",fileinfo.st_size);
  printf("permissions: %d\n",fileinfo.st_mode);
  printf("last modified: %d\n",fileinfo.st_mtime);
  }
```