```
#include <iostream>
#include <cstdlib>
#include <cerrno>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>
#include <pwd.h>
#include <grp.h>

using namespace std;
/**
 *main function where everything is done
 *@param argc number of arguments
 *@param argv[] the array of arguments.
 */
int main(int argc, char *argv[])
{

  if(argc == 1)
    {
      string filename;
      cin >> filename;
      struct stat sb;
      if (stat(filename.c_str(), &sb) == -1)
      {
        perror("stat");
        cout << "\n";
        return EXIT_FAILURE;
      }
      cout << "File: '" << filename <<"'" << endl;
      cout << "Size:  " << (long long)sb.st_size << "\t\t";
      cout << "Blocks: " << (long long)sb.st_blocks << "\t\t";
      cout << "IO Block: " << (long long)sb.st_blksize << "\t";
      switch (sb.st_mode & S_IFMT)
      {
      case S_IFBLK: cout <<"block device\n"; break;
      case S_IFCHR: cout <<"character device\n"; break;
      case S_IFDIR: cout <<"directory\n"; break;
      case S_IFIFO: cout <<"FIFO/pipe\n"; break;
      case S_IFLNK: cout <<"symlink\n"; break;
      case S_IFREG: cout <<"regular file\n"; break;
      case S_IFSOCK: cout <<"socket\n"; break;
      default: cout << "unknown\n" ; break;
      }
      stringstream ss;
      ss << hex << sb.st_dev;
      string hex = ss.str();
```

```
      cout << "Device: " << sb.st_dev << "/" << hex << "d\t";
      cout << "Inode: " << (long) sb.st_ino << "\t\t";
      cout << "Links: " << (long) sb.st_nlink << "\n";
      int octal = (long)sb.st_mode & 07777;
      cout << "Access: " << "(" << std::oct  << octal << "/" ;

      if(sb.st_mode & S_IFDIR)
      cout << "d";
      else
      cout << "-";

      cout << (((sb.st_mode & S_IRUSR) != 0) ? "r" : "-")
         << (((sb.st_mode & S_IWUSR) != 0) ? "w" : "-")
         << (((sb.st_mode & S_IXUSR) != 0) ? "x" : "-")
         << (((sb.st_mode & S_IRGRP) != 0) ? "r" : "-")
         << (((sb.st_mode & S_IWGRP) != 0) ? "w" : "-")
         << (((sb.st_mode & S_IXGRP) != 0) ? "x" : "-")
         << (((sb.st_mode & S_IROTH) != 0) ? "r" : "-")
         << (((sb.st_mode & S_IWOTH) != 0) ? "w" : "-")
         << (((sb.st_mode & S_IXOTH) != 0) ? "x" : "-") << ")";

      cout << "Uid: (" << (long)sb.st_uid << "/     " << getpwuid(sb.st_uid)-
>pw_name <<")\t";
      cout << "Gid: (" << (long)sb.st_gid << "/     " << getgrgid(sb.st_gid)-
>gr_name <<")\n";

      cout << "Access: " << ctime(&sb.st_atime);
      cout << "Modify: " <<ctime(&sb.st_mtime);
      cout << "Change: " <<ctime(&sb.st_ctime) << endl;

    }
  else
    {
      for(int i = 1; i < argc; i++)
      {
        if(strcmp(argv[i],"-") == 0 || argc == 1)
          {
            string filename;
            cin >> filename;
            struct stat sb;
              if (stat(filename.c_str(), &sb) == -1)
                {
                  perror("stat");
              cout << "\n";
              return EXIT_FAILURE;
            }
            cout << "File: '" << filename <<"'" << endl;
              cout << "Size:  " << (long long)sb.st_size << "\t\t";
              cout << "Blocks: " << (long long)sb.st_blocks << "\t\t";
              cout << "IO Block: " << (long long)sb.st_blksize << "\t";
              switch (sb.st_mode & S_IFMT)
                {
                case S_IFBLK: cout <<"block device\n"; break;
                case S_IFCHR: cout <<"character device\n"; break;
```

```
              case S_IFDIR: cout <<"directory\n"; break;
              case S_IFIFO: cout <<"FIFO/pipe\n"; break;
              case S_IFLNK: cout <<"symlink\n"; break;
              case S_IFREG: cout <<"regular file\n"; break;
              case S_IFSOCK: cout <<"socket\n"; break;
              default: cout << "unknown\n" ; break;
              }

            stringstream ss;
            ss << hex << sb.st_dev;
            string hex = ss.str();

            cout << "Device: " << sb.st_dev << "/" << hex << "d\t";
            cout << "Inode: " << (long) sb.st_ino << "\t\t";
            cout << "Links: " << (long) sb.st_nlink << "\n";
            int octal = (long)sb.st_mode & 07777;
            cout << "Access: " << "(" << std::oct  << octal << "/" ;

            if(sb.st_mode & S_IFDIR)
              cout << "d";
            else
              cout << "-";

          cout << (((sb.st_mode & S_IRUSR) != 0) ? "r" : "-")
                 << (((sb.st_mode & S_IWUSR) != 0) ? "w" : "-")
                 << (((sb.st_mode & S_IXUSR) != 0) ? "x" : "-")
                 << (((sb.st_mode & S_IRGRP) != 0) ? "r" : "-")
                 << (((sb.st_mode & S_IWGRP) != 0) ? "w" : "-")
                 << (((sb.st_mode & S_IXGRP) != 0) ? "x" : "-")
                 << (((sb.st_mode & S_IROTH) != 0) ? "r" : "-")
                 << (((sb.st_mode & S_IWOTH) != 0) ? "w" : "-")
                 << (((sb.st_mode & S_IXOTH) != 0) ? "x" : "-") << ")";

            cout << "Uid: (" << (long)sb.st_uid << "/     " <<
getpwuid(sb.st_uid)->pw_name <<")\t";
            cout << "Gid: (" << (long)sb.st_gid << "/     " <<
getgrgid(sb.st_gid)->gr_name <<")\n";

            cout << "Access: " << ctime(&sb.st_atime);
            cout << "Modify: " <<ctime(&sb.st_mtime);
            cout << "Change: " <<ctime(&sb.st_ctime) << endl;
          }
      else
        {
          char* filename = argv[i];
          struct stat sb;
          if (stat(filename, &sb) == -1)
          {
            perror("stat");
            cout << "\n";
          }

          /*
          struct stat statbuf;
```

```
struct passwd *pwd;
pwd = getpwuid(geteuid());
if ((pwd = getpwuid(statbuf.st_uid)) != NULL)
printf(" %-8.8s", pwd->pw_name);
*/

/*
struct group *grp;

if ((grp = getgrgid(statbuf.st_gid)) != NULL)
printf(" %-8.8s", grp->gr_name);
else
printf(" %-8d", statbuf.st_gid);
*/


cout << "File: '" <<filename<<"'" << endl;
cout << "Size:  " << (long long)sb.st_size << "\t\t";
cout << "Blocks: " << (long long)sb.st_blocks << "\t\t";
cout << "IO Block: " << (long long)sb.st_blksize << "\t";
switch (sb.st_mode & S_IFMT)
{
case S_IFBLK: cout <<"block device\n"; break;
case S_IFCHR: cout <<"character device\n"; break;
case S_IFDIR: cout <<"directory\n"; break;
case S_IFIFO: cout <<"FIFO/pipe\n"; break;
case S_IFLNK: cout <<"symlink\n"; break;
case S_IFREG: cout <<"regular file\n"; break;
case S_IFSOCK: cout <<"socket\n"; break;
default: cout << "unknown\n" ; break;
}

stringstream ss;
ss << hex << sb.st_dev;
string hex = ss.str();

cout << "Device: " << sb.st_dev << "/" << hex << "d\t";
cout << "Inode: " << (long) sb.st_ino << "\t\t";
cout << "Links: " << (long) sb.st_nlink << "\n";
int octal = (long)sb.st_mode & 07777;
cout << "Access: " << "(" << std::oct  << octal << "/" ;

if(sb.st_mode & S_IFDIR)
cout << "d";
else
cout << "-";

cout << (((sb.st_mode & S_IRUSR) != 0) ? "r" : "-")
   << (((sb.st_mode & S_IWUSR) != 0) ? "w" : "-")
   << (((sb.st_mode & S_IXUSR) != 0) ? "x" : "-")
   << (((sb.st_mode & S_IRGRP) != 0) ? "r" : "-")
   << (((sb.st_mode & S_IWGRP) != 0) ? "w" : "-")
   << (((sb.st_mode & S_IXGRP) != 0) ? "x" : "-")
   << (((sb.st_mode & S_IROTH) != 0) ? "r" : "-")
```

```
              << (((sb.st_mode & S_IWOTH) != 0) ? "w" : "-")
              << (((sb.st_mode & S_IXOTH) != 0) ? "x" : "-") << ")";

            cout << "Uid: (" << (long)sb.st_uid << "/     " <<
getpwuid(sb.st_uid)->pw_name <<")\t";
            cout << "Gid: (" << (long)sb.st_gid << "/     " <<
getgrgid(sb.st_gid)->gr_name <<")\n";

            cout << "Access: " << ctime(&sb.st_atime);
            cout << "Modify: " <<ctime(&sb.st_mtime);
            cout << "Change: " <<ctime(&sb.st_ctime) << endl;
          }
      }
    }
  return EXIT_SUCCESS;
}
```