```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

char* str_prefix(char *prefix, char *str);

int main(int argc, char *argv[])
{
  FILE *scorefile;
  int ruid, euid;
  char *name, *ssn;
  char *matching_pattern, *line, *match_point;
  char *score, *pscore;

  if (argc != 3) {
    printf("Usage: getscore name SSN\n");
    exit(1);
  }

  name = argv[1];
  ssn = argv[2];

  time_t current_time = time(NULL);

  ruid = getuid ();
  euid = geteuid ();
  // This is to make sure the logging command will have
  // sufficient privilege.
  if (setreuid(euid, euid)){
    perror("setreuid");
  }

  scorefile = fopen("score.txt", "r");
  if (scorefile == NULL){
    printf ("failed to open score file\n");
    exit(-1);
  }


  if ((matching_pattern = (char *)malloc(strlen(name)+17)) == NULL){
    printf("Failed to allocate memory.\n");
    exit(-1);
  }
```

```
if ((score = (char *)malloc(10)) == NULL){
  printf("Failed to allocate memory.\n");
  exit(-1);
}

if ((line = (char *)malloc(128)) == NULL){
  printf("Failed to allocate memory.\n");
  exit(-1);
}

strcpy(matching_pattern, name);
strcat(matching_pattern, ":");
strcat(matching_pattern, ssn);

while (fgets(line, 128, scorefile)!=NULL){
  if (match_point=str_prefix(matching_pattern, line)){
    if (*match_point++==':'){
        pscore = score;
        while (*match_point!=':'){
          *pscore++=*match_point++;
        }
        *pscore=0;
        printf("Your score is %s\n", score);
        free(matching_pattern);
        free(line);
        return 0;
    }
  }
}

char command[256];
printf("Invalid user name or SSN.\n");
sprintf(command, "echo \"%s: Invalid user name or SSN: %s,%s\"|cat >> error.log",
          ctime(&current_time), argv[1], argv[2]);
if (system(command)){
  perror("Logging");
}

free(matching_pattern);
free(score);
free(line);
```

```
  return -1;
}

char* str_prefix(char *prefix, char *str){
  while (*prefix && *str){
    if (*prefix != *str)
      return NULL;
    prefix++;
    str++;
  }
  return *prefix==0?str:NULL;
}
```