

```

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

// Define a simple struct for a "User" typedef struct { int id; char username[50]; char
email[100]; time_t created_at; } User;

// Function to generate a random string for usernames/emails void
generateRandomString(char *str, int length) { const char charset[] =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"; for (int i =
0; i < length; i++) { int index = rand() % (sizeof(charset) - 1); str[i] = charset[index]; }
str[length] = '\0'; }

// Function to create a new User object User createUser(int id) { User newUser; newUser.id
= id; generateRandomString(newUser.username, 8 + (rand() % 7)); // 8-14 chars
generateRandomString(newUser.email, 10 + (rand() % 10)); // 10-19 chars
strcat(newUser.email, "@example.com"); newUser.created_at = time(NULL) - (rand() %
(365 * 24 * 60 * 60)); // Up to 1 year ago

printf("// Creating user with ID: %d\n", newUser.id);
printf("// Username: %s, Email: %s\n", newUser.username, newUser.email);

return newUser;

}

// Function to print user data in a formatted way (simulating a database record or JSON)
void printUserData(const User *user) { char time_str[64]; strftime(time_str,
sizeof(time_str), "%Y-%m-%d %H:%M:%S", localtime(&user->created_at));

printf("{\n");
printf("  \"id\": %d,\n", user->id);
printf("  \"username\": \"%s\",\n", user->username);
printf("  \"email\": \"%s\",\n", user->email);
printf("  \"created_at\": \"%s\"\n", time_str);
printf("}\n");

}

int main() { srand(time(NULL)); // Seed the random number generator

```

```
printf("/*\n");
printf(" * This is a simulated data file generated by a C program.\n");
printf(" * It demonstrates creating and structuring data similar to source code.\n");
printf(" * Data represents a list of 'User' objects.\n");
printf(" *\n\n");

printf("// --- Start of User Data Records ---\n\n");

int num_users = 5 + (rand() % 6); // Generate 5-10 users
User users[num_users];

for (int i = 0; i < num_users; i++) {
    users[i] = createUser(i + 1);
    printf("\n"); // Add a blank line for readability between creation logs
}

printf("// --- Formatted User Data Output (JSON-like) ---\n\n");

printf("const UserData = [\n");
for (int i = 0; i < num_users; i++) {
    printUserData(&users[i]);
    if (i < num_users - 1) {
        printf(",\n");
    }
}
printf("];\n\n");

printf("// --- End of User Data Records ---\n");
printf("\n");
printf("/*\n");
printf(" * EOF\n");
printf(" *\n\n");

return 0;

}
```

