

تابع های نوشته شده::

تابع ::getTeamsProfile

این تابع؛ لیست تیم ها را از پروفایل ذخیره شده روی کامپیوتر میگیرد. به پوینتر به ابتدای آرایه ی تیم هاست. ورودی اون آدرس (پوینتر) به یه آرایه ی 16 تایی از یه struct از نوع team_t هست.
مثال::

```
team_t teams[16];
```

```
getTeamsProfile(teams);
```

تابع ::getTeamsDatabase دقیقاً مشابه بالایی. تنها با این تفاوت که لیست تیم ها را از دیتابیس پیشفرض میگیرد

تابع ::getPlayersProfile

این تابع هم لیست بازیکنان لیگ را (تمامی بازیکنان از تمامی تیم ها) از پروفایل کاربر میگیرد. تعداد بازیکنانی است که از پروفایل گرفته است. ورودی آن یه آدرس به پوینتری است که میخواهیم بازیکنان را در آن ذخیره کنیم (آدرس ابتدای آرایه)
مثال::

```
int count = 0;
```

```
player_t *players;
```

```
count = getPlayersProfile(&players);
```

در این کد players الان یه آرایه ی count تایی از مشخصات بازیکنان ما است.
تابع ::getPlayersDatabase دقیقاً مشابه به تابع قبلی. با این تفاوت که اطلاعات را از دیتابیس میگیرد.

تابع ::getGames

لیست بازی های انجام شده و انجام نشده را از پروفایل کاربر میگیرد (تمامی بازی هایی که بین تیم ها باید انجام شود). خروجی آن تعداد بازی هاست و ورودی آن آدرس آرایه ای است که میخواهیم لیست بازی ها را در آن ذخیره کنیم.
مثال::

```
int count = 0;
```

```
game_t *games;
```

```
count = getGames(&games);
```

تابع ::getNatayej

لیست نتایج بازیهای برگزار شده را از پروفایل کاربر بر میگرداند. مانند تابع قبلی خروجی آن تعداد بازی های انجام شده و ورودی آن آدرس آرایه ای است که میخواهیم نتایج را در آن ذخیره کنیم.
مثال::

```
int count = 0;
```

```
natayej_t *natayej;
```

```
count = getNatayej(&natayej);
```

تابع ::saveNatayejProfile

پس از انجام شدن یک بازی (بازی ها) در صورتی که بخواهیم آن بازی/بازیها را در پروفایل کاربر ذخیره کنیم (تنها ذخیره ی نتایج و نه دیگر اطلاعات) میتوانیم این تابع را فراخوانی کنیم. خروجی این تابع در صورت موفقیت ذخیره سازی نتایج 1 و در صورت عدم موفقیت 0 خواهد بود. ورودی این تابع آرایه ی نتایج بازیها و تعداد آنهاست.

مثال::

فرض کنیم نتایج ما در یک آرایه ی

```
natayej[100];
```

ذخیره شده است. برای ذخیره کردن این نتایج در دیتابیس تابع

```
saveNatayejProfile(natayej, 100);
```

را فراخوانی میکنیم.

توجه کنید که این تابع؛ تمامی نتایج قبلی را پاک کرده و یک آرایه از نتایج را جایگزین آن میکند. برای ذخیره ی تنها یک/چند نتیجه ی خاص در پروفایل از این تابع استفاده نکنید.

تابع ::saveGamesProfile

پس از ساخته شدن و معلوم شدن بازیهایی که باید انجام شود؛ از این تابع استفاده کنید. این تابع لیست بازیها را در پروفایل کاربر ذخیره میکند. خروجی آن 1 در صورت موفقیت و 0 در صورت عدم موفقیت است.

مثال:

فرض کنید بازیهای ما در یک آرایه ی

```
games[300]
```

ذخیره شده است. برای ذخیره کردن این بازیها در پروفایل؛ تابع زیر را فراخوانی میکنیم

```
saveGamesProfile(games, 300);
```

تابع ::saveTeamsProfile

از این تابع پس از به روزرسانی شدن مشخصات تیم ها (مانند موقعی که کاربر تیم خودش را اضافه میکند یا موقعی که امتیاز؛ تعداد گل ها و... تیم ها تغییر میکند) استفاده میشود. این تابع مشخصات تیم ها را در پروفایل کاربر ذخیره میکند. خروجی این تابع 0 در صورت عدم موفقیت در ذخیره کردن اطلاعات و 1 در صورت موفقیت در ذخیره کردن اطلاعات است.

مثال::

فرض کنید مشخصات تیم ها در یک آرایه ی 16 تایی به اسم teams ذخیره شده است. برای ذخیره کردن این اطلاعات بر روی پروفایل کاربر از تابع به شکل زیر استفاده می شود

```
saveTeamsProfile(teams,16);
```

تابع ::savePlayersProfile

از این تابع برای ذخیره کردن اطلاعات بازیکنان (تمامی بازیکنان) بر روی پروفایل استفاده میشود. از این تابع پس از به روز شدن اطلاعات بازیکنان پس از انجام یک بازی (تغییر در روحیه و بازیکنان) استفاده شود. برای مثال برای ذخیره ی لیست بازیکنان با این تابع میتوان به شکل زیر عمل کرد::

فرض کنید تمام بازیکنان در یک آرایه ی 300 تایی players ذخیره شده اند. داریم

```
savePlayersProfile(players, 300);
```

تابع ::clearPage

ورودی و خروجی این تابع void است. از این تابع برای پاک کردن صفحه استفاده کنید

تابع ::welcomeFootball

این تابع خوشگله ((از این تابع فقط موقع شروع شدن برنامه استفاده کنید ((فقط هم توی لینوکس کار میکنه (((ویندوزیش را حال نداشتم بنویسم ((

تابع ::random

این تابع هم قرار بود لیلی بنویسه؛ دیدم چهار خط بیشتر نیست و ساده است، خودم نوشتمش ((این تابع یه بازی میگیره و یه عدد رندوم توی اون بازه تولید میکنه. این عدد میتونه جزو نقاط شروع و پایان دامنه هم باشه

مثال::

```
int a = random(2,4);
```

تابع ::checkProfile

بررسی میکنه که آیا پروفایلی روی کامپیوتر ذخیره شده یا نه. در صورت ذخیره شدن عدد 1 و در صورت عدم وجود پروفایلی بر روی کامپیوتر عدد 0 را برمیگردونه

تابع ::clearProfile

این تابع هم تمام اطلاعات پروفایل کاربر را پاک میکنه براش ((

تابع ::saveProfile

تمام اطلاعات پروفایل را save میکنه. تمام اطلاعات اعم از بازیکنان؛ تیم ها؛ نتایج؛ بازی ها و... پروفایل را به طور کامل ذخیره میکنه. خروجیش 1 در صورت ذخیره ی موفقیت آمیز و 0 در صورت عدم موفقیت است

مثال::

```
saveProfile(teams, team_size, players, player_size, games, game_size, natayej,  
natayej_size);
```

البته احتمالاً این تابع در ادامه تغییر خواهد کرد و ممکنه چیزهای بیشتری را هم ذخیره کنه...

ساختار داده ها:

فعلاً ما داده‌های زیر و استراکچرهای زیر را داریم::

```
struct team {  
    int id;  
    char name[100];  
    int isPlayer;  
    int money;  
    int score;  
    int zade;  
    int khorde;  
    int count;  
};
```

```
struct game {  
    int id;  
    int team1id;  
    int team2id;  
    int raftorbargasht;  
    int week;  
};
```

```
struct natayej{  
    int gameid;  
    int team1goal;  
    int team2goal;  
};
```

```
struct goal{  
    int gameid;  
    int playerid;  
};
```

```
struct player{  
    int id;  
    int teamid;  
    char name[50];
```

```
char lastname[50];  
int goal;  
int position;  
int skill;  
int amadegi;  
int khastegi;  
int rouhiye;  
int khoshunat;  
};
```

که البته هر کدومشون به صورت زیر typedef شده‌اند

```
typedef struct team team_t;  
typedef struct game game_t;  
typedef struct natayej natayej_t;  
typedef struct goal goal_t;  
typedef struct player player_t;
```

خب دیگه
این‌ام از یه سری توضیحات
باشد که رستگار شوید ((