



Tic Tac Toe

توضیح پروژه

در بخش دوم پروژه‌ی Tic Tac Toe قرار است که ساختار برنامه با استفاده از توابع و آرایه‌ها، منظم شود.

در مورد ساختار برنامه شما با توجه به عملکرد هر قسمت از برنامه، تابع‌های مختلف را جدا کنید و به هر تابع وظیفه‌ی خاصی را محول کنید. برای مثال یک تابع می‌تواند مسئول چاپ جدول بازی باشد و تابع دیگری مسئول چک کردن مکان‌های درست باشد. بنابراین در تابع `main` باید روال کلی برنامه مشخص باشد و هر وظیفه‌ی خاص مانند مثال‌هایی که ذکر شد، در توابع پیاده‌سازی شوند. استفاده و طراحی درست توابع در برنامه در این بخش از پروژه اهمیت ویژه‌ای دارد. بنابراین به نحوه‌ی طراحی ساختار برنامه توجه کنید.

در مورد نحوه‌ی نگه‌داری اطلاعات در این پروژه، شما با توجه به میزان اطلاعاتی که در هر قسمت ذخیره می‌کنید، از ساختار داده‌ای به نام آرایه استفاده می‌کنید که تعداد اطلاعات درون هر آرایه، با توجه به تعریفی که برای آن در نظر می‌گیرید متفاوت است. استفاده درست از این ساختار نیز اهمیت بسیار زیادی دارد. بنابراین به استفاده درست از آرایه‌ها در پروژه خود توجه کنید. برای مثال، یک بدنه‌ی `main` می‌تواند بصورت زیر نوشته شود:

```

int main()
{
    char cells[9];
    int turn = 1;
    int check_step;

    initial_game(cells);

    while(check_endgame(turn, cells) < 0)
    {
        check_step = play_game(turn, cells);
        if (check_step > 0)
        {
            if (turn == 1)
                turn = 2;
            else
                turn = 1;
        }
        print_board(cells);
    }

    return 0;
}

```

قطعه کد شماره ۱

لازم به ذکر است که تعریف توابع استفاده شده در قطعه کد شماره ۱، به صورت زیر است:

```

void print_board(char cell[]);
int play_game(int turn, char cells[]);
void initial_game(char cells[]);
int check_endgame(int turn, char cells[]);

```

معرفی توابع استفاده شده در قطعه کد شماره ۱

در این قطعه کد، با استفاده از چهار تابع و یک آرایه (که در آن خانه های مربوط به جدول نگه داشته شده اند)، بدنه ی main نوشته شده است. شما می توانید با استفاده از این طراحی (و یا هر طراحی دیگری که انجام می دهید) از نوشته شدن قطعه کد های تکراری جلوگیری کنید.

در این پروژه شما موظف هستید که با استفاده از نوشتن توابع (با توجه به قطعه کد شماره ۱ و یا هر ساختار دیگری که خودتان طراحی می کنید)، برنامه ای که در بخش قبل نوشته اید را بازنویسی کنید.

نحوه‌ی نمره‌دهی :

عنوان	نمره
نام‌گذاری مناسب و اصولی متغیرها	۱۰
استفاده‌ی مناسب و درست از آرایه	۲۰
استفاده‌ی مناسب و درست از تابع	۳۰
کامنت‌گذاری مناسب در هرجایی که نیاز به مستندسازی دارد	۱۰
عملکرد صحیح برنامه	۱۵
پاسخ‌دهی مناسب هنگام تحویل و تسلط بر کد	۱۵

نحوه‌ی تحویل پروژه :

شما باید یک فایل c. در قسمت در نظر گرفته شده با عنوان CA1[Upload here] در صفحه‌ی درس در سایت CECM آپلود نمایید. مهلت آپلود تا ساعت ۲۳:۵۵ روز چهارشنبه ۲۸ آبان است. تأخیر در سه روز نخست به ازای هر روز ۱۰ درصد و در دو روز دوم به ازای هر روز ۱۵ درصد جریمه خواهد داشت. پس از این پنج روز، به هیچ وجه نمره‌ای در نظر گرفته نخواهد شد. همچنین توجه کنید که با توجه به تاخیر در نظر گرفته شده به هیچ وجه امکان تمدید پروژه‌ها وجود ندارد.

فایل‌های خود را در یک فایل زیپ با فرمت “zip” و با نام CA1P2_[SID].zip قرار دهید که SID همان شماره‌ی دانشجویی شماست. برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۰۰۰ باشد باید نام فایل خود را CA1_810198000.zip قرار دهید.

پروژه ها برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی میشوند و انجام آنها به صورت انفرادی خواهد بود. همچنین، در صورت شباهت میان دو پروژه (که به وسیله ی نرم افزارهای مربوطه چک میشود) برای هر دو نفر نمره ی صفر در نظر گرفته خواهد شد .

در صورت وجود هرگونه سوال میتوانید پرسشهای خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید یا به icsp98.ca@gmail.com ایمیل بزنید.

موفق و سربلند باشید