

پروژه برنامه‌نویسی: بازی چند-وزیر

در این پروژه می‌خواهیم یک بازی چند-وزیر ([n-Queens](#)) را ساخته و حل نماییم. در بازی چند-وزیر، صفحه بازی یک ماتریس $n \times n$ است که n وزیر در آن چیده شده است. بازیکن باید موقعیت وزیرها را طوری تغییر دهد که هیچ دو وزیری نتوانند با حرکات مجاز وزیر، یکدیگر را بزنند. حرکات مجاز وزیر، قطری، سطری و ستونی هستند. به عبارت دیگر هیچ دو وزیری نباید در یک قطر، سطر یا ستون قرار داشته باشند.

این پروژه شامل دو مرحله است:

مرحله اول:

در این مرحله، شما باید محیط بازی را نوشته و امکان بازی را برای بازیکن فراهم کنید. در لحظه اول، صفحه بازی خالی است. بازیکن دو امکان دارد: گذاشتن وزیر در یک خانه یا حذف یک وزیر از یک خانه. بعد از هر حرکت، باید وزیرهایی که یکدیگر را می‌زنند و جهت آن توسط برنامه مشخص شوند و به اطلاع کاربر برسند. بازی زمانی خاتمه می‌یابد که یک جواب پیدا شود یا کاربر بخواهد بازی را خاتمه دهد.

مرحله دوم:

نوشتن یک بازیکن که بتواند بازی را حل نماید. برای این کار باید دو مفهوم را بشناسید: جستجوی پسگرد ([backtracking](#)) و پشته ([stack](#)). برای یافتن جواب باید از جستجوی پسگرد استفاده نمایید و برای پیاده‌سازی جستجوی پسگرد از ساختار پشته استفاده کنید.

نکات مهم:

۱- مقدار n توسط کاربر تعیین می‌شود و می‌تاند بین ۴ تا ۸ باشد.

۲- برنامه شما باید حتماً حاوی توابع و متغیرهای زیر باشد:

array stack	پشته
array board	صفحه بازی
int top	بالای پشته
int collisions (array, length)	بررسی و مشخص کردن وزیرهایی که یکدیگر را می‌زنند و بازگرداندن تعداد آنها
void solve (array, length)	حل بازی با استفاده از پسگرد
void next (array, array, lengths)	مشخص کردن حالت بعدی در الگوریتم پسگرد
push, pop	توابع درج و حذف پشته

- ۳- در تابع solve کافیت یک راه حل پیدا شود.
- ۴- هنگام فراخوانی تابع solve زمان اجرای آن را محاسبه کرده و نمایش دهید.
- ۵- برنامه باید به کاربر امکان انتخاب بدهد: می‌خواهد خودش مسئله را حل کند (بازی کند) یا می‌خواهد سیستم مسئله را حل کند (فراخوانی solve).
- ۶- زمانی را که کاربر صرف بازی می‌کند، محاسبه و نمایش دهید.
- ۷- همه خطاهای ممکن در اثر انتخاب‌های نامناسب کاربر (مثل انتخاب اندیس ناموجود در آرایه) را پیش‌بینی کرده و در صورت بروز، پیغام مناسب را برای آن چاپ نمایید. سعی کنید پیغام‌هایی که برای کاربر چاپ می‌کنید پیغام‌هایی جذاب و ترغیب‌کننده باشند.
- ۸- تمامی اعضای گروه باید بر برنامه نوشته‌شده تسلط داشته و قادر به پاسخ‌گویی به سؤالاتی که در زمان ارائه مطرح می‌شود، باشند.

عوامل موثر در نمره پروژه:

- ✓ به درستی اجرا شدن برنامه. (وتویی: سایر نمرات در صورت برقراری این شرط، به پروژه مربوطه تعلق خواهد گرفت)
- ✓ رعایت نام‌گذاری‌های تعیین شده. (وتویی: سایر نمرات در صورت برقراری این شرط، به پروژه مربوطه تعلق خواهد گرفت)
- ✓ خوانایی کد: استفاده از comment، نام‌گذاری مناسب متغیرها، رعایت indent و فاصله‌گذاری‌های مناسب. (۱۰ نمره)
- ✓ عملکرد درست برنامه (۳۰ نمره)
- ✓ استفاده به‌جا از پیغام‌ها در راهنمایی کاربر. (۱۰ نمره)
- ✓ پیش‌بینی مناسب خطاهای ممکن در زمان اجرا و چاپ پیغام‌های مناسب. (۱۰ نمره)
- ✓ محاسبه زمان تابع solve. (۱۰ نمره)
- ✓ محاسبه زمان کاربر. (۱۰ نمره)
- ✓ نمایش مناسب صفحه بازی. (۱۰ نمره)
- ✓ سرعت اجرای مناسب. (۱۰ نمره)

نمره اضافه:

- ✓ استفاده از ساختمان‌ها (struct) در پیاده‌سازی
- ✓ گرافیکی نوشتن برنامه
- ✓ تابع solve خیلی سریع