

1. برنامه‌ای بنویسید که از کاربر یک عدد دریافت کند و مقدار  $n$ -مین عدد فیبوناچی را محاسبه کند.

2. برنامه‌ای بنویسید که عددی از کاربر بگیرد و بررسی کند که آیا این عدد یک عدد کامل است یا خیر. عدد کامل عددی است که مجموع مقسوم‌علیه‌های آن، به جز خودش، برابر خود عدد باشد.

3. برنامه‌ای بنویسید که یک عدد صحیح از کاربر بگیرد و بررسی کند که آیا این عدد از سمت چپ و راست یکسان است (پالیندروم) یا خیر. محدودیت: از هیچ تابع آماده استفاده نکنید.

4. برنامه‌ای بنویسید که یک عدد از کاربر بگیرد و آن را به صورت وارونه چاپ کند. محدودیت: از هیچ آرایه یا تابع آماده‌ای استفاده نکنید.

5. برنامه‌ای بنویسید که یک عدد مخفی ثابت (عدد جادویی) بین 1 تا 100 داشته باشد. کاربر باید عدد را حدس بزند. اگر عدد وارد شده کاربر کمتر از عدد مخفی بود، پیام "عدد وارد شده کوچک‌تر است" نمایش دهد. اگر بیشتر بود، پیام "عدد وارد شده بزرگ‌تر است" نمایش دهد. برنامه باید تعداد دفعات حدس را نیز شمارش کرده و در انتها نمایش دهد.

6. برنامه‌ای بنویسید که یک عدد تصادفی (یا ثابت) سه‌رقمی به عنوان **عدد مخفی** داشته باشد (مثلاً 347).

- کاربر باید با وارد کردن یک عدد سه‌رقمی، آن را حدس بزند.
- برنامه باید به کاربر بگوید که:
  1. چه تعداد از ارقام وارد شده **در مکان درست** (هم مکان و هم مقدار) قرار دارند.
  2. چه تعداد از ارقام وارد شده **فقط در مقدار درست** هستند ولی در مکان اشتباه قرار دارند.

### محدودیت‌ها:

1. فقط از حلقه‌ها و شرط‌ها برای پیاده‌سازی استفاده شود.
  2. برنامه باید تا زمانی که کاربر عدد صحیح را حدس نزنند، ادامه پیدا کند.
  3. در پایان، تعداد تلاش‌های کاربر نیز نمایش داده شود.
7. برنامه‌ای بنویسید که یک بمب مجازی با زمان شروع مشخص (مثلاً 10 ثانیه) داشته باشد.

- کاربر باید در هر ثانیه (با وارد کردن یک عدد خاص، مثلاً 1) تایمر را متوقف کند.
- اگر کاربر نتواند عدد درست را در زمان مقرر وارد کند، بمب منفجر می‌شود.
- اگر کاربر موفق شود تایمر را به صفر برساند، پیام موفقیت نمایش داده می‌شود.

### مراحل بازی:

1. تایمر از عددی مشخص (مثلاً 10) شروع می‌کند و هر بار که کاربر عدد صحیح را وارد می‌کند، تایمر یک واحد کم می‌شود.
2. اگر کاربر عدد اشتباهی وارد کند یا از محدوده زمانی فراتر رود، بمب منفجر می‌شود.

### محدودیت‌ها:

1. فقط از حلقه و شرط برای پیاده‌سازی استفاده شود.
2. برنامه باید شمارش زمان و کنترل ورودی کاربر را در حلقه بررسی کند.
3. برنامه تا پایان زمان تایمر یا وقوع انفجار اجرا شود.