

```

# =====
# پروژه 2 :سیستم مدیریت تسکها (To-Do List)
# =====

# =====# لیست اصلی تسکها=====
# یک لیست خالی برای ذخیره تسکها
# هر تسک یک دیکشنری است با دو کلید:
# - "title": (عنوان تسک) رشته
# - "done": (انجام نشده = False ، انجام شده = True) وضعیت انجام
tasks = []

# =====# تابع نمایش همه تسکها=====
def show_tasks():
    # اگر لیست خالی بود
    # تعداد عناصر داخل لیست را برمیگرداند
    if len(tasks) == 0:
        print("\n🚫 لیست تسکها خالی است")
        # return یعنی از این تابع خارج شو و دستورات بعدی اجرا نشوند
        return

    # اگر لیست خالی ،نبود تسکها را نمایش بده
    print("\n" + "=" * 50)
    print("لیست تسکهای شما:")
    print("-" * 50)

    # یک شمارنده خودکار به هر عنصر لیست اضافه میکند
    # enumerate(tasks, 1): (شمارنده از 1 شروع میشود) نه از 0
    # i = ...), 3, 2, 1 (شمارنده)
    # task = هر دیکشنری داخل لیست
    for i, task in enumerate(tasks, 1):
        # عملگر سه تایی (if کوتاه)
        # اگر task["done"] برابر True بود → status = "✅"
        # در غیر این صورت → status = "⌚"
        status = "✅" if task["done"] else "⌚"
        # چاپ ،شماره وضعیت و عنوان تسک
        print(f"{i}. {status} {task['title']}")
    print("=" * 50)

# =====# تابع اضافه کردن تسک جدید=====
def add_task():
    # دریافت عنوان تسک از کاربر
    title = input("\nعنوان تسک: ")

    # ساخت یک دیکشنری جدید برای تسک
    # done = False یعنی تسک جدید انجام نشده است
    new_task = {

```

```

        "title": title,
        "done": False
    }

    # append: عنصر جدید را به انتهای لیست اضافه می‌کند
    tasks.append(new_task)
    print(f"✅ تسک '{title}' اضافه شد.")

===== #تابع علامت زدن تسک به عنوان انجام شده=====
def mark_done():
    #ابتدا لیست فعلی را به کاربر نشان بده تا شماره تسک را ببیند
    show_tasks()

    #اگر لیست خالی، بود از تابع خارج شو
    if len(tasks) == 0:
        return

    #بلوک try برای مدیریت خطا) اگر کاربر به جای عدد حرف زد)
    try:
        #دریافت شماره تسک از کاربر
        num = int(input("شماره تسکی که انجام داده اید: "))

        #بررسی می‌کنیم شماره وارد شده بین 1 تا تعداد تسک‌ها باشد
        if 1 <= num <= len(tasks):
            #توجه: اندیس لیست از 0 شروع می‌شود
            #اگر کاربر شماره 1 را وارد، کند باید به اندیس 0 دسترسی پیدا کنیم
            #به همین دلیل از num-1 استفاده می‌شود
            tasks[num - 1]["done"] = True
            print(f"🎉 تسک '{tasks[num - 1]['title']}' انجام شد")
        else:
            print("شماره نامعتبر.")
    except ValueError:
        #اگر کاربر به جای عدد حرف وارد کرد
        print("لطفاً عدد وارد کنید.")

===== #تابع حذف تسک=====
def delete_task():
    #ابتدا لیست فعلی را نمایش بده
    show_tasks()

    if len(tasks) == 0:
        return

    try:
        num = int(input("شماره تسک برای حذف: "))
        if 1 <= num <= len(tasks):

```

```

#متد pop: عنصر مورد نظر را از لیست حذف می‌کند و آن را برمی‌گرداند
# num-1: ایندکس عنصر مورد نظر
removed = tasks.pop(num - 1)
print(f"🗑️ تسک '{removed['title']}' حذف شد.")
else:
    print("شماره نامعتبر.")
except ValueError:
    print("لطفاً عدد وارد کنید.")

===== #منوی اصلی برنامه=====
print("📋 به سیستم مدیریت تسکها خوش آمدید")

#حلقه بی‌نهایت) تا زمانی که کاربر گزینه خروج را انتخاب کند ادامه دارد)
while True:

    #نمایش منو
    print("\n" + "=" * 40)
    print("منوی اصلی:")
    print("1. نمایش تسکها")
    print("2. اضافه کردن تسک جدید")
    print("3. علامت زدن تسک انجام شده")
    print("4. حذف تسک")
    print("5. خروج")

    #دریافت انتخاب کاربر
    choice = input("انتخاب شما: ")

    #ساختار if-elif برای هدایت کاربر به تابع مناسب
    if choice == "1":
        show_tasks()
    elif choice == "2":
        add_task()
    elif choice == "3":
        mark_done()
    elif choice == "4":
        delete_task()
    elif choice == "5":
        print("خروج از برنامه. فعلاً")
        break # break: while خارج شدن از حلقه
    else:
        #اگر کاربر عددی خارج از محدوده 1 تا 5 وارد کرد
        print("ورودی نامعتبر! لطفاً عدد 1 تا 5 را وارد کنید")

```