

```

# =====
# پروژه 5 : ماشین حساب میانگین وزنی) معدل)
# =====

# ===== متغیرهای سراسری =====
# لیست برای ذخیره اطلاعات دروس
# هر درس یک دیکشنری با چهار کلید دارد:

# - "name": نام درس
# - "score": (نمره درس) 0 تا 20
# - "unit": تعداد واحد
# - "grade": (وضعیت توصیفی) محاسبه می‌شود
courses = []

# جمع کل واحدهای گرفته شده
total_units = 0

# جمع (نمره × واحد) برای هر درس
total_weighted_score = 0

# تابع نمایش راهنما
def show_calculator():
    print("\n" + "=" * 40)
    print("📊 ماشین حساب معدل")
    print("=" * 40)
    print("دستورالعمل:")
    print("- برای هر درس، نمره و تعداد واحد را وارد کنید.")
    print("- را وارد کنید 'end' برای پایان، دادن کلمه")
    print("- نمرات باید بین 0 تا 20 باشند")
    print("- واحدها باید عدد مثبت باشند")

# تابع تبدیل نمره به وضعیت توصیفی
def get_grade_point(score):
    # این تابع بر اساس نمره یک رشته توصیفی برمی‌گرداند
    if score >= 17:
        return "عالی"
    elif score >= 14:
        return "خوب"
    elif score >= 12:
        return "قابل قبول"
    elif score >= 10:
        return "مردود مشروط"
    else:
        return "مردود مطلق"

# نمایش راهنما در ابتدای برنامه
show_calculator()

```

```

=====حلقه اصلی دریافت دروس=====
while True:
    print("\n" + "-" * 30)

    #دریافت نام درس از کاربر
    course_name = input("برای پایان 'end' یا نام درس: ")

    #بررسی شرط خروج
    # lower(): ورودی را به حروف کوچک تبدیل می‌کند تا مقایسه آسان شود
    if course_name.lower() == "end":
        break # خارج شدن از حلقه

    #بلوک try برای مدیریت خطا (اگر کاربر به جای عدد حرف زد)
    try:
        #دریافت نمره و تبدیل به float (عدد اعشاری)
        score = float(input("نمره درس: "))

        #اعتبارسنجی نمره: باید بین 0 تا 20 باشد
        if score < 0 or score > 20:
            print("⚠️.نمره باید بین 0 تا 20 باشد. این درس ثبت نشد")
            continue # به ابتدای حلقه برگرد (این درس را نادیده بگیر)

        #دریافت تعداد واحد و تبدیل به int (عدد صحیح)
        unit = int(input("تعداد واحد: "))

        #اعتبارسنجی واحد: باید مثبت باشد
        if unit <= 0:
            print("⚠️.واحد باید مثبت باشد. این درس ثبت نشد")
            continue

        #ساخت دیکشنری درس و اضافه کردن به لیست
        courses.append({
            "name": course_name,
            "score": score,
            "unit": unit,
            "grade": get_grade_point(score) # فراخوانی تابع برای تعیین وضعیت
        })

        #به روزرسانی متغیرهای محاسباتی
        total_units += unit # جمع واحدها
        total_weighted_score += score * unit # جمع (نمره × واحد)

        print(f"✅ واحد ثبت شد {unit} و {score} با نمره '{course_name}' درس")

    except ValueError:

```

```

print("⚠️ خطا: لطفاً برای نمره و ، واحد عدد وارد کنید")

##### محاسبه و نمایش کارنامه #####
print("\n" + "=" * 60)

# بررسی اینکه آیا حداقل یک درس ثبت شده است
if len(courses) == 0:
    print("هیچ درسی ثبت نشد")
else:
    # محاسبه معدل
    # فرمول: جمع نمره‌های وزنی (تقسیم بر) جمع واحدها
    gpa = total_weighted_score / total_units

    # نمایش کارنامه
    print("📋 کارنامه شما:")
    print("=" * 60)

    # عنوان ستون‌ها
    # < 25 یعنی چپ‌چین با عرض 25 کاراکتر
    print(f"{'25':<25} {'8':<8} {'6':<6} {'15':<15}"
          "وضعیت: {course['name']}<25 {course['score']<8 {course['unit']<6 {course['gpa']<15}")
    print("-" * 60)

    # حلقه برای نمایش هر درس در یک خط
    for course in courses:
        print(f"{'25':<25} {'8':<8} {'6':<6} {'15':<15}"
              "وضعیت: {course['name']}<25 {course['score']<8 {course['unit']<6 {course['gpa']<15}")
        print("-" * 60)

    print(f"جمع واحدها: {total_units}")
    print(f"معدل شما: {gpa:.2f}") # یعنی 2 رقم اعشار .2f

    # تعیین رتبه کلی بر اساس معدل
    if gpa >= 17:
        rank = "عالی (مشمول بورسیه تحصیلی) 🏆"
    elif gpa >= 15:
        rank = "خوب 🥇"
    elif gpa >= 12:
        rank = "قابل قبول 🥈"
    else:
        rank = "نیاز به تلاش بیشتر 🥉"

    print(f"رتبه کلی: {rank}")
    print("=" * 60)

    # پیام تشویقی اضافی
    if gpa < 12:
        print("\n🙏 گران نباشید! با تمرین بیشتر می‌توانید معدل خود را بالا ببرید")

```

```
elif gpa >= 17:  
    print("\n🎓 تبریک! عملکرد شما عالی است")
```