لم يكن الكود يدعم (Millimeter): تم إجراء عدة تغييرات على الكود بهدف تحسين وظيفته وجودته.1. إضافة وحدة الميليمتر وحدة الميليمتر. بما أن الميليمتر هو جزء من النظام المتري (1 ميليمتر = 0. فقد كان من الضروري إضافة هذه الوحدة إلى unit_to_cm إلى القاموس "millimeter": 0.1 القاموس ليتمكن المستخدمون من تحويل القيم منها وإليها.● التغيير: تم إضافة الذي يحتوي على الوحدات الأخرى. بذلك أصبح الكود يدعم الميليمتر كإحدى الوحدات القابلة للتحويل.2. تحديث القاموس لتحسين مرونة التحويلات:● السبب: كان الكود في البداية يعتمد على العديد من العمليات الحسابية المتكررة لتحويل الوحدات إلى السنتيمتر ثم إلى الوحدة المطلوبة. يمكن الآن إجراء هذه العمليات بسرعة ومرونة أكبر.● التغيير: تم تنظيم الوحدات في بحيث يتم تحديد معامل التحويل إلى السنتيمتر لكل وحدة، مما يسهل إضافة أو تعديل الوحدات (unit_to_cm) قاموس المستقبلية.3. تحسين التحقق من صحة المدخلات: كانت المدخلات غير الصحيحة (مثل وحدات غير مدعومة) قد تتسبب في ظهور أخطاء برمجية قد تربك المستخدم.● التغيير: تم إضافة تحقق من صحة الوحدات المدخلة عبر التأكد من وجود الوحدات وهذا يضمن أن المستخدم يحصل على إشعار واضح في .(ValueError) في القاموس. يتم رفع استثناء مع رسالة خطأ واضحة حال كان إدخالهم غير صحيح.● السبب: كانت واجهة المستخدم بحاجة إلى تحسين في ما يخص عرض الوحدات المتاحة وكيفية التفاعل مع البرنامج.● التغيير: تم تحديث الرسائل التعليمية للمستخدم لتشمل الوحدة الجديدة "ميليمتر"، وتم توجيه المستخدم بشكل أكثر وضوحًا بشأن كيفية إدخال الوحدات.5. التحقق من القيم المدخلة من المستخدم:● السبب: خلال اختبار البرنامج، تبين أن البرنامج كان يقبل القيم السلبية أو القيم غير الرقمية. وقد يتسبب ذلك في حدوث أخطاء.● التغيير: تمت إضافة حلقة تحقق للتأكد من أن القيمة المدخلة تكون إيجابية وأنها قيمة عددية صحيحة، مع رسالة تحذير إذا كانت القيم المدخلة غير صحيحة.6. تحسين التفاعل مع الأخطاء:● السبب: في حالة إدخال بيانات خاطئة أو حدوث أخطاء منطقية، كان من المفيد للمستخدم أن يحصل على تفاصيل واضحة حول المشكلة.● التغيير: تم تحسين التعامل مع الأخطاء عبر رفع استثناءات محددة، مع رسائل توضح السبب (مثل "الوحدة المدخلة غير صحيحة"). هذا يساعد في توجيه المستخدم بشكل أفضل في حال حدوث خطأ.7. تبسيط الشيفرة وتقليل التكرار:● السبب: كان الكود يحتوي على العديد من العمليات الحسابية المتكررة لكل وحدة. هذا لتخزين قيم التحويل، مما أدى إلى unit_to_cm جعل الشيفرة أقل مرونة وأصعب في التعديل.● التغيير: تم استخدام القاموس تقليل التكرار وزيادة وضوح الكود وسهولة صيانته. يمكن الآن إضافة أو تعديل الوحدات ببساطة من خلال القاموس دون الحاجة لتكرار الحسابات.1. زيادة المرونة: أصبح من السهل إضافة المزيد من الوحدات أو تعديل الوحدات الموجودة.2. تحسين الأداء: تم تحسين الكود ليكون أكثر كفاءة باستخدام القاموس بدلاً من العمليات الحسابية المتكررة.3. تحسين تجربة المستخدم: تم إضافة رسائل أكثر وضوحًا وإشعارات في حال وجود خطأ، مما يساعد المستخدم على فهم ما يحدث بشكل أفضل.4. تحسين التحقق من المدخلات: أصبح البرنامج أكثر قوة ضد المدخلات غير الصحيحة، مما يقلل من حدوث الأخطاء.5. الوضوح والصيانة: أصبح الكود أكثر وضوحًا وأسهل في الصيانة بسبب استخدام القاموس بدلًا من كتابة الحسابات لكل وحدة.● إضافة لتحسين التفاعل.● دعم وحدات Tkinter باستخدام مكتبات مثل (GUI) واجهة رسومية: يمكن إضافة واجهة مستخدم رسومية لدعم المزيد من الاحتياجات. تلبية متطلبات التحويل بين الوحدات "mile" أو "yard" أكثر: يمكن إضافة المزيد من الوحدات مثل المتنوعة:● الغرض: كان من المطلوب أن يكون البرنامج قادرًا على تحويل قيم وحدات القياس بين عدة وحدات معيارية.● التلبية: قدم البرنامج دعمًا للتحويل بين الوحدات الأكثر استخدامًا مثل الكيلومتر، تم إضافة الميليمتر إلى الكود في مرحلة لاحقة لتوسيع دعم التحويلات وتلبية متطلبات المستخدمين الذين يحتاجون إلى هذه الوحدة الدقيقة.● الملاءمة: يساعد هذا البرنامج المستخدمين في إجراء التحويلات بين الوحدات بشكل سريع وبدون الحاجة إلى القيام بالحسابات يدويًا، مما يوفر الوقت ويضمن الدقة. التعامل مع المدخلات غير الصحيحة:● الغرض: من المتوقع أن يكون المستخدم قد يواجه صعوبة في إدخال قيم غير صحيحة أو قد يستخدم وحدات غير مدعومة.● التلبية: تم إدخال تحقق شامل للمدخلات باستخدام رسائل خطأ واضحة. إذا كانت القيمة المدخلة غير رقمية أو سالبة، سيطلب من المستخدم إدخال قيمة صحيحة. كذلك إذا كانت الوحدة المدخلة غير صحيحة (مثل إدخال وحدة غير موجودة)، سيتم إعلام المستخدم بذلك عبر استثناءات واضحة.● الملاءمة: هذا يحسن تجربة المستخدم، ويجعل البرنامج سهل الاستخدام حتى للمستخدمين المبتدئين أو غير الملمين بالمفاهيم الرياضية. وضوح الرسائل والإرشادات للمستخدم:● الغرض: كان من المهم أن يقدم البرنامج واجهة مستخدم مفهومة وبسيطة لتوجيه المستخدم خلال العملية.● التلبية: تم إضافة رسائل توجيهية للمستخدم، مثل طباعة الوحدات المتاحة للتحويل، وطباعة رسالة توضح النتائج بعد

إجراء التحويل. كما تم تحسين الرسائل عند حدوث خطأ بحيث تكون واضحة ومباشرة.● الملاءمة: هذه الميزة تضمن أن المستخدم يعرف بالضبط كيفية إدخال البيانات بشكل صحيح وأيضًا يحصل على إخراج واضح للنتيجة.● الغرض: من المهم أن يكون البرنامج سهل الاستخدام بحيث يمكن للمستخدم إدخال القيم والوحدات بطريقة بسيطة.● التلبية: البرنامج يعتمد على واجهة التي تعد سهلة الاستخدام للمستخدمين الذين يعرفون كيفية العمل مع المدخلات النصية. يمكن للمستخدم (CLI) سطر الأوامر ،(GUI) إدخال القيمة ووحدتي القياس بسهولة،● الملاءمة: بالرغم من أنه يمكن تحسين واجهة المستخدم بإضافة واجهة رسومية فإن البرنامج بسيط وفعّال في تلبية الاحتياجات الأساسية للمستخدمين الذين يتعاملون مع سطر الأوامر. توفير سرعة ومرونة في العمليات الحسابية:● الغرض: من المتوقع أن يطلب المستخدم إجراء عمليات تحويل دقيقة وسريعة.● التلبية: تم استخدام لتحسين الكفاءة في عملية التحويل، حيث يتم تحويل جميع الوحدات إلى السنتيمتر أولاً، ثم إتمام التحويل unit_to_cm القاموس إلى الوحدة المطلوبة. هذه الطريقة تجعل التحويلات سريعة للغاية وأكثر مرونة.● الملاءمة: يضمن هذا أن العمليات الحسابية تتم بسرعة ودقة دون الحاجة لإعادة كتابة نفس العمليات الحسابية مرارًا وتكرارًا في الكود، ما يجعل البرنامج أكثر كفاءة في المعالجة. قابلية التوسع والتطوير في المستقبل:● الغرض: من الممكن أن يتطور استخدام البرنامج ليشمل وحدات جديدة في المستقبل، أو قد يتطلب إضافة ميزات أخرى مثل القدرة على إجراء التحويلات بين وحدات أخرى.● التلبية: تم تصميم الكود ليكون مرنًا. عند إضافة المزيد من الوحدات، مما يجعل الكود جاهزًا للتطور unit_to_cm إضافة وحدات جديدة، يمكن ببساطة توسيع القاموس مع احتياجات المستخدم.● الملاءمة: يوفر هذا تصميمًا مرنًا وقابلًا للتطوير بحيث يمكن للبرنامج التكيف مع متطلبات المستخدم المستقبلية.● الملاءمة الإجمالية للغرض المطلوب:● سهولة الاستخدام: البرنامج يفي بالغرض الأساسي لتحويل وحدات القياس بشكل بسيط وسهل.● الفعالية: يحقق دقة التحويل بين الوحدات بشكل سريع وفعال باستخدام القاموس في حسابات التحويل.● المرونة والتطوير: تم تصميم البرنامج بحيث يمكن توسيعه في المستقبل بسهولة لإضافة وحدات جديدة أو وظائف أخرى.● التفاعل مع المستخدم: التحقق من المدخلات وإعطاء رسائل واضحة يسهل على المستخدم فهم كيفية استخدام البرنامج بشكل صحيح.● البرنامج يلبي متطلبات المستخدم بفعالية عن طريق تقديم أداة تحويل وحدات قياس مرنة وسهلة الاستخدام. يتميز بالتحقق الدقيق من المدخلات، ويوفر تجربة مستخدم ممتازة مع إمكانية توسيع البرنامج في المستقبل. تلبية متطلبات التحويل بين الوحدات المتنوعة:● الغرض: كان من المطلوب أن يكون البرنامج قادرًا على تحويل قيم وحدات القياس بين عدة وحدات معيارية.● التلبية: قدم البرنامج دعمًا للتحويل بين الوحدات الأكثر استخدامًا مثل الكيلومتر، تم إضافة الميليمتر إلى الكود في مرحلة لاحقة لتوسيع دعم التحويلات وتلبية متطلبات المستخدمين الذين يحتاجون إلى هذه الوحدة الدقيقة.● الملاءمة: يساعد هذا البرنامج المستخدمين في إجراء التحويلات بين الوحدات بشكل سريع وبدون الحاجة إلى القيام بالحسابات يدويًا، مما يوفر الوقت ويضمن الدقة. التعامل مع المدخلات غير الصحيحة:● الغرض: من المتوقع أن يكون المستخدم قد يواجه صعوبة في إدخال قيم غير صحيحة أو قد يستخدم وحدات غير مدعومة.● التلبية: تم إدخال تحقق شامل للمدخلات باستخدام رسائل خطأ واضحة. إذا كانت القيمة المدخلة غير رقمية أو سالبة، سيطلب من المستخدم إدخال قيمة صحيحة. كذلك إذا كانت الوحدة المدخلة غير صحيحة (مثل إدخال وحدة غير موجودة)، سيتم إعلام المستخدم بذلك عبر استثناءات واضحة.● الملاءمة: هذا يحسن تجربة المستخدم، ويجعل البرنامج سهل الاستخدام حتى للمستخدمين المبتدئين أو غير الملمين بالمفاهيم الرياضية. وضوح الرسائل والإرشادات للمستخدم:● الغرض: كان من المهم أن يقدم البرنامج واجهة مستخدم مفهومة وبسيطة لتوجيه المستخدم خلال العملية.● التلبية: تم إضافة رسائل توجيهية للمستخدم، مثل طباعة الوحدات المتاحة للتحويل، وطباعة رسالة توضح النتائج بعد إجراء التحويل. كما تم تحسين الرسائل عند حدوث خطأ بحيث تكون واضحة ومباشرة.● الملاءمة: هذه الميزة تضمن أن المستخدم يعرف بالضبط كيفية إدخال البيانات بشكل صحيح وأيضًا يحصل على إخراج واضح للنتيجة.● الغرض: من المهم أن يكون البرنامج سهل الاستخدام بحيث يمكن للمستخدم إدخال القيم والوحدات بطريقة بسيطة.● التلبية: البرنامج يعتمد على واجهة التي تعد سهلة الاستخدام للمستخدمين الذين يعرفون كيفية العمل مع المدخلات النصية. يمكن للمستخدم (CLI) على واجهة سطر الأوامر للمستخدم إدخال القيمة ووحدتي القياس بسهولة،● الملاءمة: بالرغم من أنه يمكن تحسين واجهة المستخدم بإضافة واجهة فإن البرنامج بسيط وفعّال في تلبية الاحتياجات الأساسية للمستخدمين الذين يتعاملون مع سطر الأوامر. توفير ،(GUI) رسومية سرعة ومرونة في العمليات الحسابية:● الغرض: من المتوقع أن يطلب المستخدم إجراء عمليات تحويل دقيقة وسريعة.● التلبية: لتحسين الكفاءة في عملية التحويل، حيث يتم تحويل جميع الوحدات إلى السنتيمتر أولاً، ثم unit_to_cm تم استخدام القاموس

إتمام التحويل إلى الوحدة المطلوبة. هذه الطريقة تجعل التحويلات سريعة للغاية وأكثر مرونة.● الملاءمة: يضمن هذا أن العمليات الحسابية تتم بسرعة ودقة دون الحاجة لإعادة كتابة نفس العمليات الحسابية مرارًا وتكرارًا في الكود، ما يجعل البرنامج أكثر كفاءة في المعالجة. قابلية التوسع والتطوير في المستقبل:● الغرض: من الممكن أن يتطور استخدام البرنامج ليشمل وحدات جديدة في المستقبل، أو قد يتطلب إضافة ميزات أخرى مثل القدرة على إجراء التحويلات بين وحدات أخرى.● التلبية: تم تصميم الكود لإضافة المزيد من الوحدات، مما يجعل unit_to_cm ليكون مرنًا. عند إضافة وحدات جديدة، يمكن ببساطة توسيع القاموس الكود جاهزًا للتطور مع احتياجات المستخدم.● الملاءمة: يوفر هذا تصميمًا مرنًا وقابلًا للتطوير بحيث يمكن للبرنامج التكيف مع متطلبات المستخدم المستقبلية.● الملاءمة الإجمالية للغرض المطلوب:● سهولة الاستخدام: البرنامج يفي بالغرض الأساسي لتحويل وحدات القياس بشكل بسيط وسهل.● الفعالية: يحقق دقة التحويل بين الوحدات بشكل سريع وفعال باستخدام القاموس في حسابات التحويل.● المرونة والتطوير: تم تصميم البرنامج بحيث يمكن توسيعه في المستقبل بسهولة لإضافة وحدات جديدة أو وظائف أخرى.● التفاعل مع المستخدم: التحقق من المدخلات وإعطاء رسائل واضحة يسهل على المستخدم فهم كيفية استخدام البرنامج بشكل صحيح.● البرنامج يلبي متطلبات المستخدم بفعالية عن طريق تقديم أداة تحويل وحدات قياس مرنة وسهلة الاستخدام. يتميز بالتحقق الدقيق من المدخلات، ويوفر تجربة مستخدم ممتازة مع إمكانية توسيع البرنامج في المستقبل.