

Es sind die vorgegebenen Datentypen zu beachten (int Werte werden in Variable vom Typ int eingelesen und double Werte in Variable vom Typ double).

Keine Zeichenketten oder Strings verwenden, wenn Zahlen verlangt sind.

Wenn ein Ergebnis berechnet werden soll, dann reicht die Ausgabe alleine (außer wenn nur diese explizit verlangt ist) nicht aus. Der Wert muss in einer Variable zur Verfügung stehen, sodass er prinzipiell für weitere Berechnungen verwendbar wäre.

Die Idee der rekursiven Aufteilung des zu lösenden Problems ist bereits in der Angabe beschrieben. Sie müssen diese nur mehr in C++ umsetzen!

Schreiben Sie eine rekursive Funktion `int anz_gerade(int wert)`, die die Anzahl der geraden Ziffern in der als Parameter übergebenen ganzen Zahl (diese muss ungleich 0 sein) retourniert. Verwenden Sie in der Funktion keine Schleifen.

z.B.:

```
cout << anz_gerade(123); //gibt 1 aus
cout << anz_gerade(278); //gibt 2 aus
```

Rekursive Formulierung des Problems: Die Anzahl der geraden Ziffern in einer beliebigen Zahl ergibt sich aus der Anzahl der geraden Ziffern in der Zahl ohne die letzte Ziffer plus 1, falls die letzte Ziffer gerade ist bzw. plus 0, falls die letzte Ziffer ungerade ist.

Die Rekursion endet bei der Zahl 0, die 0 gerade Ziffern liefert (genau genommen ist 0 natürlich eine gerade Ziffer, die 0, mit der die Rekursion endet, kommt aber in der ursprünglichen Zahl nicht vor. Wir haben 0 als möglichen Parameterwert deshalb explizit ausgeschlossen).

z.B.: $\text{anz_gerade}(278) = \text{anz_gerade}(27) + 1 = \text{anz_gerade}(2) + 1 = \text{anz_gerade}(0) + 1 + 1 = 0 + 1 + 1 = 2$

Zusatzaufgabe (etwa 15 bis 30 Minuten extra): Lösen Sie das Problem iterativ.