

Aufgabe 1:**(5 Punkte)**

Gegeben sei das folgende Produkt

$$p_n = \prod_{k=1}^n \left(1 + \frac{1}{k^2}\right).$$

Schreiben Sie ein vollständiges C++-Programm zur Berechnung dieses Produkts. Der Benutzer soll die obere Grenze n eingeben können und das Programm soll dann den Wert von p_n ermitteln und ausgeben.

Aufgabe 2:**(5 Punkte)**

Schreiben Sie eine C++-Funktion `reverse`, die ein Feld von `double`-Variablen umkehrt. Die Funktion soll als Argument die Länge des Feldes n und das Feld selbst erhalten. Die gültigen Feldindizes bewegen sich zwischen 0 und $n - 1$.

Alle benötigten Variablen sollen lokal definiert oder Argumente der Funktion sein – verwenden Sie keine globalen Variablen.

Aufgabe 3:**(10 Punkte)**

Eine Klasse soll einen Tachometer simulieren, d.h. zur Verarbeitung von Wertepaaren (Strecke, Zeit) dienen. Strecken seien dabei in Metern gemessen, Zeiten in Sekunden. Die Klasse soll in folgendem Programm zum Einsatz kommen:

```
#include <iostream>
using namespace std ;

// Klasse tacho zu ergaenzen

int main()
{
    tacho trip (0, 0) ;
    trip += tacho ( 2000, 413.5) ; // Tram Uni->Zoo, 2 km, 413.5 sec
    trip += tacho ( 547.5, 800.3) ; // zu Fuss im Zoo, 547.5 m, 800.3 sec
    trip += tacho ( 2000, 415) ;    // Tram zur Uni, 2 km, 415 sec

    cout << "Gesamtstrecke: " << trip.get_dist() << endl ;
    cout << "Durchschnitts-Geschwindigkeit: " << avg_speed(trip) << endl ;
}
```

Folgende Methoden werden verwendet: `+=` Addition eines weiteren Wegabschnitts, `get_dist()` Ausgabe der Gesamtdistanz. Die freie Funktion `avg_speed` soll die Durchschnittsgeschwindigkeit berechnen.

Schreiben Sie die benötigte Klasse `tacho` und alle benötigten Konstruktoren, Mitgliedsfunktionen und freien Funktionen, die in dem gegebenen Hauptprogramm aufgerufen werden.

Die Strecken- und die Zeitvariable innerhalb der Klasse sollen von außen unzugängliche `double`-Variablen sein.

Aufgabe 4:**(8 Punkte)**

Setzen Sie an der markierten Stelle in folgendem Programm die letzte Ziffer Ihrer Matrikelnummer ein. Schreiben Sie dann *vollständig* auf, was von dem Programm ausgegeben wird.

```
#include <iostream>
using namespace std ;

int f(int z) {
    z += 1 ;
    cout << "UP1: " << z << endl ;
    return z*2 ;
}

int f(int* q) {
    (*q) += 2 ;
    cout << "UP2: " << *q << endl ;
    return 13 + (*q)/1000 ;
}

int f(int r, int s) {
    if (s==0) return 0 ;
    else return r + f(r-1,s-1) ;
}

int main() {
    int i= _ ; // <- setzen Sie hier bitte die letzte Ziffer
               // Ihrer Matrikelnummer ein.

    int j=f(i) ;
    cout << "HP1: " << i << " " << j << endl ;

    int* p = &i ;
    cout << "HP2: " << *p << endl ;

    j=f(p) ;
    cout << "HP3: " << i << " " << j << endl ;

    cout << "HP4: " << f(6,3) << endl ;
}
```

Aufgabe 5:**(12 Punkte)**

Eine n -stellige positive Zahl ($1 \leq n \leq 9$) heie *ziffern-schn*, wenn alle Ziffern von 1 bis n genau einmal vorkommen. Schreiben Sie ein vollstndiges C++-Programm, das Folgendes leistet:

- Eine long-Zahl wird von der Tastatur eingelesen. Sie brauchen nicht zu berprfen, ob die eingegebene Zahl positiv ist
- Ausgabe einer Liste, wie oft jede der Ziffern 0 bis 9 vorkommt.
- Ausgabe, ob die Zahl *ziffern-schn* ist oder nicht.