

```
#include <stdio.h>
#include <limits.h>
#include "my_math.h"
```

```
void pascall(int n);
int main()
{
```

```
    printf("%lu\n", my_fak(65)); // Maximaler Wert n, für den die Fakultät noch berechnet werden kann ist 65
    printf("%lu\n", my_binomial(2, 2));
```

```
    pascall(10);
```

```
    return 0;
```

```
}
```

```
void pascall(int n)
```

```
{
```

```
    int i;
```

```
    int k;
```

```
    unsigned long a[10];
```

```
    for(i = 0; i <= n; ++i){
        a[i] = 0;
```

```
    }
```

```
    printf("\nPascall Dreieck: \n\n");
```

```
    for(i = 0; i <= n; ++i){
        for(k = 0; k <= i; ++k){
            printf("%lu\t", my_binomial(i, k));
            a[i] += my_binomial(i, k);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    printf("\n");
```

```
    printf("\n");
```

```
    for(i = 0; i <= n; ++i){
        printf("Summe der Zeile %i: %lu\n", i, a[i]);
```

```
    }
```

```
}
```

```
#include "my_math.h"
```

```
unsigned long my_fak(unsigned long a)
```

```
{
```

```
    if(a <= 1){
```

```
        return 1;
```

```
    }else{
```

```
        return (a * my_fak(a - 1));
```

```
    }
```

```
}
```

```
unsigned long my_binomial(unsigned long n, unsigned long k)
```

```
{
```

```
    if(n >= 1 && k >= 0 && n >= k){
```

```
        return my_fak(n) / (my_fak(n - k) * my_fak(k));
```

```
    }else{
```

```
        return ERROR;
```

```
    }
```

```
}
```

```
#ifndef MY_MATH_H_INCLUDED
#define MY_MATH_H_INCLUDED

#define ERROR 0;
unsigned long my_fak(unsigned long a);
unsigned long my_binomial(unsigned long n, unsigned long k);

#endif
```

```

#include <stdio.h>
#include "strecken.h"

int main()
{
    int i;
    char eingabe;
    int size = TEILSTRECKEN;
    double strecke[TEILSTRECKEN] = {1.5, 2.5, 5, 100};

    // Ausgabe der Teilstrecken
    for(i = 0; i < TEILSTRECKEN; ++i){
        printf("%.2lf\n", strecke[i]);
    }
    printf("\n");

    do{

        printf("Sind die Teilstrecken in Seemeilen oder Kilometer eingegeben?\n");
        printf("'S' für Seemeilen und 'K' für Kilometer: ");
        scanf("%c", &eingabe);
        new_line();

        if(eingabe != 'K' && eingabe != 'S'){
            printf("Bitte tätigen Sie eine gültige Eingabe\n\n");
            flush();
        }

    }while (eingabe != 'K' && eingabe != 'S');

    // Prograblauf bei Seemeilen
    if(eingabe == 'S'){

        printf("Die Gesamtlänge der Teilstrecken beträgt: %.3lf Seemeilen\n", gesamtleange(strecke, size));
        printf("Die durchschnittliche Länge der Teilstrecken beträgt: %.3lf Seemeilen\n", durchschnitt_laenge(strecke, size));

        flush();
        new_line();
        printf("Für eine Ausgabe in km bitte 'K' eingeben: ");
        scanf("%c", &eingabe);

        if(eingabe == 'K'){
            new_line();
            printf("Die Gesamtlänge der Teilstrecken beträgt %.3lf Kilometer\n", seamile_to_km(gesamtleange(strecke, size)));
            printf("Die durchschnittliche Länge der Teilstrecken beträgt: %.3lf Kilometer\n", seamile_to_km(durchschnitt_laenge(strecke, size)));
        }

        flush();
        new_line();
        printf("Möchten Sie noch weitere Teilstrecken hinzufügen? ['Y' / 'N']: ");
        scanf("%c", &eingabe);

        if(eingabe == 'Y'){

            if(einlesen(strecke)){

                printf("Die Eingabe war erfolgreich.\n");

                new_line();
                // Ausgabe der Teilstrecken
                for(i = 0; i < TEILSTRECKEN; ++i){
                    printf("%.2lf\n", strecke[i]);
                }

                new_line();

                printf("Die neue Gesamtlänge der Teilstrecken beträgt: %.3lf Seemeilen\n", gesamtleange(strecke, size));
                printf("Die neue durchschnittliche Länge der Teilstrecken beträgt: %.3lf Seemeilen\n", durchschnitt_laenge(strecke, size));
            }else{
                printf("Die Eingabe war nicht erfolgreich.\n");
            }
        }
    }
}

```

```
}
```

```
}
```

```
// Programmablauf bei Kilometern
```

```
}else if(eingabe == 'K'){
```

```
    printf("Die Gesamtlänge der Teilstrecken beträgt: %.3lf Kilometer\n", gesamtleange(strecke, size));
```

```
    printf("Die durchschnittliche Länge der Teilstrecken beträgt: %.3lf Kilometer\n", durchschnitt_laenge(strecke, size));
```

```
    flush();
```

```
    new_line();
```

```
    printf("Für eine Ausgabe in Seemeilen bitte 'S' eingeben: ");
```

```
    scanf("%c", &eingabe);
```

```
    if(eingabe == 'S'){
```

```
        new_line();
```

```
        printf("Die Gesamtlänge der Teilstrecken beträgt %.3lf Seemeilen\n", km_to_seamile(gesamtleange(strecke, size)));
```

```
        printf("Die durchschnittliche Länge der Teilstrecken beträgt: %.3lf Seemeilen\n", km_to_seamile(durchschnitt_laenge(strecke, size)));
```

```
    }
```

```
    flush();
```

```
    new_line();
```

```
    printf("Möchten Sie noch weitere Teilstrecken hinzufügen? ['Y' / 'N']: ");
```

```
    scanf("%c", &eingabe);
```

```
    if(eingabe == 'Y'){
```

```
        if(einlesen(strecke)){
```

```
            printf("Die Eingabe war erfolgreich.\n");
```

```
            new_line();
```

```
            // Ausgabe der Teilstrecken
```

```
            for(i = 0; i < TEILSTRECKEN; ++i){
```

```
                printf("%.2lf\n", strecke[i]);
```

```
            }
```

```
            new_line();
```

```
            printf("Die neue Gesamtlänge der Teilstrecken beträgt: %.3lf Kilometer\n", gesamtleange(strecke, size));
```

```
            printf("Die neue durchschnittliche Länge der Teilstrecken beträgt: %.3lf Kilometer\n", durchschnitt_laenge(strecke, size));
```

```
        }else{
```

```
            printf("Die Eingabe war nicht erfolgreich.\n");
```

```
        }
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

```

#include <stdio.h>
#include "strecken.h"

#define ERROR 0
#define SUCCESS 1
#define TEILSTRECKEN 5

#define seamile_to_km(a) (a * 1.852)
#define km_to_seamile(a) (double)(a / 1.852)

#define new_line() printf("\n")

void flush()
{
    while(getchar() != '\n'){

    }
}

double gesamtleange(double a[], int size)
{
    int i;
    double strecke = 0;

    for(i = 0; i < size; ++i){
        strecke += a[i];
    }

    return strecke;
}

double durchschnitt_laenge(double a[], int size)
{
    return (gesamtleange(a, size) / size);
}

int einlesen(double a[])
{
    int i = 0;
    int status;

    while(a[i] != 0){
        ++i;
    }

    status = scanf("%lf", &a[i]);

    if(!status){
        flush();
    }
}

```

```
if(i >= TEILSTRECKEN){  
    return ERROR;  
}  
  
return SUCCESS;  
  
}
```

```
#ifndef STRECKEN_H_INCLUDED
#define STRECKEN_H_INCLUDED

#define ERROR 0
#define SUCCESS 1
#define TEILSTRECKEN 5

#define seamile_to_km(a) (a * 1.852)
#define km_to_seamile(a) (double)(a / 1.852)
#define new_line() printf("\n")

void flush();
double gesamtleange(double a[], int size);
double durchschnitt_laenge(double a[], int size);
int einlesen(double a[]);

#endif
```