assignments-1.md

1)

2)

a)

```c
int main(void)
{
        return 0;
}
```

b)

```c
#include <stdio.h>

int main(void)
{
        printf("Hallo\n");
        return 0;
}

/* Ausgabe: Hallo */
```

c)

```c
#include <stdio.h>

int main(void)
{
        double d = 5.50;

        printf("%.2f\n", d);
        return 0;
}

/* Ausgabe: 5.50 */
```

d)

```c
#include <stdio.h>

int main(void)
{
        char d = 'a';

        printf("%c\n", d);
        return 0;
}

/* Ausgabe: a */
```

3)

a)

```c
#include <stdio.h>

int main(void)
{
        printf("line\n\n");

        return 0;
}
```

b)

```c
#include <stdio.h>

int main(void)
{
        printf("%%\\\\n\n");

        return 0;
}
```

c)

```c
#include <stdio.h>

int main(int argc, char * argv[])
{
        printf("Anzahl der Parameter:%d\nProgrammname:%s\n", argc - 1, argv[0]);

        return 0;
}
```

d)

```c
#include <stdio.h>

int main(int argc, char * argv[])
{
        printf("Vierfaches der anzahl der Parameter:%d\n", (argc - 1) * 4);

        return 0;
}
```

e)

```c
#include <stdio.h>
#include <math.h>

int main(int argc, char * argv[])
{
        double result = sqrt((double)(argc - 1) * 2);

        printf("Quadratwurzel der verdoppelten Anzahl der Parameter:%.2f\n", result);

        return 0;
}
```

f)

```c
#include <stdio.h>
```

```c
int main(int argc, char * argv[])
{
        printf("Programmname: %s\nArgument 1:   %s\nArgument 2:   %s\n",
               argv[0],
               argv[1],
               argv[2]
               );

        return 0;
}
```

g)

```c
#include <stdio.h>

int main(void)
{
        int input;

        printf("Bitte geben Sie eine Zahl ein: ");

        scanf("%d", &input);

        printf("Sie haben %d eingegeben\n", input);

        return 0;
}
```

4)

a)

```c
double calc_arithmetic_mean(int a, int b)
```

b)

```c
#include <stdio.h>
#include <math.h>
/* wird das wirklich benötigt? */

/* define PI -  only needed incase I misunderstood circumference*/
#define M_PI acos(-1.0)

double calc_circumference_square(double length, double width)
{
        return 2 * (length + width);

        /*
           // circumference of smallest and largest possible circle:
           double smaller, larger;

           if (length < width) {
                smaller = length;
                larger = width;
           } else {
                smaller = width;
                larger = length;
           }

           // smallest
           return M_PI * smaller;

           // larger
           return M_PI * larger;
         */
}
```

```c
int main(void)
{
        double length, width;

        scanf("%lf %lf", &length, &width);
        printf("Der Umfang eines Rechtecks mit Laenge %f und Breite %f ist %f\n",
               length,
               width,
               calc_circumference_square(length, width));
        return 0;
}
```

c)

```c
#include <stdio.h>

int main(void)
{
        char input;

        printf("Please enter any character: ");
        scanf("%c", &input);
        printf("Previous character: %c\n",
               input - 1);

        return 0;
}
```

d)

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
        char input;

        printf("Please enter any character: ");

        scanf("%c", &input);

        printf("Abstand zu 'A': %d\n", abs('A' - input));

        return 0;
}
```

e)

```c
#include <stdio.h>
#include <math.h>

double milliseconds_to_hours(double ms)
{
        return ms / 1000 / 60 / 60;
}

int main(void)
{
        int ms;

        printf("Please enter a duration in milliseconds: ");
        scanf("%d", &ms);
        printf("%d milliseconds ammount to %.2f hours.\n",
               ms,
               milliseconds_to_hours(ms));
```

```
        return 0;
}
```