

Contents

1	probe-mab: C-basic	1
2	AUFGABE: caesar.c	1
3	AUFGABE: fastpow.c	2

1 probe-mab: C-basic

2 AUFGABE: caesar.c

- gegeben:

```
//mycat.c   Eingabe nach Ausgabe
#include <stdio.h>
int main(){
    int ch;
    while ( (ch= fgetc(stdin)) != EOF){
        fputc(ch, stdout);
    }

    return 0;
}
```

- Erstelle das Programm caesar.c
 - Es wird das gesamte Alphabet um eine bestimmte Anzahl von Buchstaben **verschoben** und
 - dadurch jeder Buchstabe des Klartextes einzeln verändert.
 - Der "Key" besteht also aus einer Zahl, um die der ASCII-Wert des Plain-Zeichens erhöht wird.
- Vorgaben:
 - Es sollen nur Buchstaben verschlüsselt werden.
 - Verwende im Programm die Variable: int key=5;
 - Verwende zum Testen die Dateiumlenkung: ./caesar.exe < caesar.c

- Zusatz:
 - Können Sie die Datei: werwolf-caesarkodiert.txt knacken?
- Beispiel: Cäsar verschlüsseln
 - Plaintext: "HALLO"
 - Key: 2
 - Ciphertext: JCNNQ
- Beachte:
 - Wenn man zB. zum Zeichen 'Z' kommt muss man beim Zeichen 'A' weiter zählen. (analog z -> a)
 - Wenn das jeweilige CIPHERzeichen > 'Z' (analog für 'z')
 - dann CiPHERzeichen = CIPHERzeichen - 26
- Beispiel:

```
ch=ch+key;
if (ch>'Z')
    ch= ch-26;
```

bzw. f r Kleinbuchstaben

```
ch=ch+key;
if (ch>'z')
    ch= ch-26;
```

- Anmerkung: Entschlüsselt wird mit $\text{key}=26-\text{key}$;
- Beispiel:
 - Hello, world! (key=4)
 - Lipps, asvph! (entschlüsselt wird mit 22) (vgl: 26 - 4)

3 AUFGABE: fastpow.c

- x hoch y kann man wie folgt sehr schnell berechnen:
- wenn y ungerade:

- `ergebnis = ergebnis * x;`
 - `y = y - 1;`
- wenn `y` gerade:
 - `x = x * x;`
 - `y = y / 2;`
- Beispiel: `x=3` und `y=16`
 - `ergebnis = 3 hoch 16`
 - `ergebnis = (3 hoch 2) hoch 8`
- Beispiel: `x=3` und `y=17`
 - `ergebnis = 3 hoch 17`
 - `ergebnis = 3 hoch 16 * 3`
 - `ergebnis * 3 = 3 hoch 16`
- Erstelle das Programm `fastpow.c`
 - lies `x` und `y` ein
 - gib das Ergebnis von `x hoch y` aus