

1) Rekursive Funktion (20 Punkte): Schreibe eine Funktion `long pow(int a, unsigned int b)`, die die Potenz der ganzen Zahl `a` mit der natürlichen Zahl `b` **rekursiv** berechnet und zurückliefert!

2) Funktionen Überladen (20 Punkte): Zeigen sie mit einer selbstgewählten einfachen Funktion die verschiedenen Möglichkeiten wie man Funktionen überladen kann (+5 Punkte wenn sie auch Standardwerte bei parameter verwenden)!

3) Modularisierung (20 Punkte): Eine Funktion `int sum(unsigned int a)` soll mittels H-Datei (inkl. Guards), separater CPP Datei und `main.cpp` (für Testaufrufe) umgesetzt werden!

Bsp: `cout << sum(4); //4+3+2+1=10`

4) Algorithmik (20 Punkte): Schreiben sie eine Funktion `print_pattern`, welche die Breite und Höhe eines zu zeichnenden Rechteckes angibt! Definieren sie sinnvolle maximal Werte (Terminalgröße ist normalerweise 80 Zeichen bei 20 Zeile) und gehen sie auch auf unmöglich Eingaben mit `throw logic_error` ein und behandeln diese auch!

Bsp: `print_pattern(4,5)`

```
*****
*   *
*   *
*   *
*   *
*****
```

5) Stringverarbeitung (20 Punkte): Schreiben sie eine Funktion `inverseString` welche die Großbuchstaben eines übergebenen Strings in Kleinbuchstaben umwandelt und die Kleinbuchstaben in Großbuchstaben – alle anderen Zeichen bleiben unverändert!

HINWEIS: `'A'=65` `'Z'=90` `'a'=97` `'z'=122`

Bsp: `cout << inverseString("AlexW");` `// aLEXw`