

! 2.1 - spot recursion



PT Klausurvorbereitung

2025-01-29 15:30 THU

Zeit: 60min, Gebraucht: 52min
Reelle Klausurzeit: ???

1

```
void arraySmoothing(int arr[], int arrLen, float* out) {
    for(int i=1; i<arrLen-1; i++) {
        float avg = (arr[i-1] + arr[i] + arr[i+1]) / 3.0;
        out[i-1] = avg;
    }
}
```

2.1

```
...
int c=0;
if (n <= 0) return 0;
countDivBy3Rec(?)  

if (a[0] % 3 == 0) c++;
return c + countDivBy3Rec(a+1, n-1);
}
```

2.2

...{

int sum = 0;

~~while (n > 10)~~ do {

int last-digit = n % 10;

sum += last-digit;

n = n / 10;

} while (n != 0);

return sum;

3.1

> int sum=0;

> for (...; i < n; ...)

> return sum;

3.2

> *out = 0;

> if (... && ...)

4 - initialize 2D-array

```
4] int **result = NULL;  
    for (int r=0; r<rows; r++) {  
        int *m = if malloc(cols * sizeof(int));  
        if (m = if malloc(cols * sizeof(int)) == NULL):  
            // handle malloc error  
        }  
        result[r] = m;  
    }  
    return result;  
  
- - - - -  
free: ... {  
    for (int r=0; r<rows; r++) {  
        free(m[r]);  
    }  
}
```

! 5: Schleife

* gesetzt, dass stark ergänzt
Probleme können sich aus elgocore ergeben
ergeben (z.B. "n")

6] a) Terminiert* mit $n=0$ oder $n=-1$ für
gerade/ungerade Zahlen. $x > 0$ wird die while Schleife
unterbrochen. Für $n \leq 0$ wird sie geschlossen.

b) Terminiert nicht: da x vom Typ int ist,
ist das Resultat von $8 - 1/2 = 0$.
Danach bleibt $x = x/2 = 0$. Die
Kondition $x < 0$ ist immer wahr.

(Falls (int)(-1/2) == 0, terminiert das Programm)

c) Terminiert.

Der Call-Stack: $g(5) \rightarrow g(3) \rightarrow g(1)$

In $g(1)$ greift der base case $n=1$ und
wir bubble zurück.