

Übungsarbeit Informatik

Nr.1

Du sollst eine Funktion zur Berechnung der Fakultät von (n!) entwickeln.

- Schreibe eine iterative Funktion zur Berechnung von n! !
- Stelle eine Rekursionsformel für n! auf!
- Schreibe eine rekursive Funktion zur Berechnung von n! !

Nr.2

Definiere, wann eine Funktion oder Prozedur rekursiv ist und beschreibe kurz den Rekursionsfall und den Basisfall!

Nr.3

a) Beweise, dass für alle $n > 2$ gilt: $n^2 - 2n - 1 > 0$ mit Hilfe der vollständigen Induktion!

b) Gegeben: Zahlenfolge a mit $a_1 = 1$, $a_{n+1} = 2 * a_n + 1$

Berechne einige Folgenglieder!

Dann stelle eine Formel zur Berechnung von a_n auf und Beweise sie mit Hilfe der vollständigen Induktion!

Nr.4

function wasmacheich(wort:string; i:integer) :integer;

begin

if i > length(wort) **then** wasmacheich := 0

else

if wort[i] = 'N' **then** wasmacheich := wasmacheich(wort, i+1);

else wasmacheich := wasmacheich(wort, i+1) + 1;

end;

a) Welcher Wert wird mit ('BONBON', 1) berechnet?

b) Was berechnet diese Funktion allgemein? Beschreibe verbal das Ergebnis dieser Funktion!

Lösungsvorschlag

Nr.1

Du sollst eine Funktion zur Berechnung der Fakultät von (n!) entwickeln.

- a) *Schreibe eine iterative Funktion zur Berechnung von n! !*
- b) *Stelle eine Rekursionsformel für n! auf!*
- c) *Schreibe eine rekursive Funktion zur Berechnung von n! !*

a)
function ifak (n:integer):integer;
var erg,i:integer;
begin
erg:=1;
for i:=1 **to** n **do** erg:=erg*i;
result:=erg;
end;

b)
$$n! \begin{cases} 1, & \text{wenn } : n = 1 (n = 0) \\ n * (n-1)!, & \text{wenn } : n > 1 \end{cases}$$

c)
function rfak(n:integer):integer;
begin
if n=1 **then** result:=1 //Basisfall
 else result:=n*rfak(n-1); //Rekursionsfall
end;

Nr.2

Definiere, wann eine Funktion oder Prozedur rekursiv ist und beschreibe kurz den Rekursionsfall und den Basisfall!

Funktionen oder Prozeduren heißen rekursiv, wenn sie sich selbst, aber mit kleinerem Argument (verringerte Problemgröße) aufrufen.

Rekursionsfall: Funktion ruft sich selbst mit verringerter Problemgröße auf.

Basisfall: Funktion liefert einen vordefinierten Rückgabewert. Somit Ende der Selbstaufrufe.

Nr.3

a) Beweise, dass für alle $n > 2$ gilt: $n^2 - 2n - 1 > 0$ mit Hilfe der vollständigen Induktion!

b) Gegeben: Zahlenfolge a mit $a_1 = 1$, $a_{n+1} = 2 * a_n + 1$

Berechne einige Folgenglieder!

Dann stelle eine Formel zur Berechnung von a_n auf und Beweise sie mit Hilfe der vollständigen Induktion!

a)

Induktionsanfang: $n=3$ $3^2 - 2*3 - 1 = 2 > 0$

Induktionsvoraussetzung: Für ein $n > 2$ gilt: $n^2 - 2n - 1 > 0$

Induktionsbehauptung: $(n+1)^2 - 2(n+1) - 1 > 0$

Induktionsschluss: $(n+1)^2 - 2(n+1) - 1 = n^2 + 2n + 1 - 2n - 2 - 1$
 $= n^2 - 2n - 1 + 2n - 1$

$n^2 - 2n - 1$: laut Induktionsvoraussetzung

$2n - 1 > 0$

b)

$a_2 = 3$

$a_3 = 7$

$a_4 = 15$

$a_5 = 31$

$a_6 = 63$

Vermutung: $a_n = 2^n - 1$

Induktionsanfang: $a_1 = 2^1 - 1 = 1$; $a_1 = 1$

Induktionsvoraussetzung: Für ein $n \in \mathbb{N}$ gilt: $a_n = 2^n - 1$

Induktionsbehauptung: $a_{n+1} = 2^{n+1} - 1$

Induktionsschluss: $a_{n+1} = 2 * a_n + 1 = 2 * (2^n - 1) + 1$ (laut Induktionsvoraussetzung)
 $= 2^{n+1} - 2 + 1 = 2^{n+1} - 1$

Nr.4

function wasmacheich(wort:string; i:integer) :integer;

begin

if $i > \text{length}(\text{wort})$ **then** wasmacheich:=0

else

if $\text{wort}[i] = 'N'$ **then** wasmacheich:=wasmakeich(wort,i+1);

else wasmakeich:=wasmakeich(wort,i+1)+1;

end;

a) Welcher Wert wird mit ('BONBON',1) berechnet?

b) Was berechnet diese Funktion allgemein? Beschreibe verbal das Ergebnis dieser Funktion!

a) Der Wert 4 wird berechnet.

b) Die Funktion zählt ab der Stelle i, wie viele Buchstaben im Wort enthalten sein, die kein 'N' sind.