Minne - heap - stack

Stack - the memory in which local variables exist in their "scope", is often limited.

Whats in the heap?

#include <stdio.h></stdio.h>			heap
int v1 = 42;			
	/* 0 */		
int main()	/* 1 */		
{ int v2 = 4;			
			stack ->
funk();		/* O */	
	/* 3 */	/* 1 */	
return 0;		/* 2 */	
}		/* 3 */	
void funk()	/* 2 */		
{ char v3:			
char v3; return;			
}			

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>			
int v1 = 42;			he
int main()	/* 0 */ /* 1 */		
{ int v2 = 4;			stack -
funk();		/* O */	
	/* 3 */	/* 1 */	
return 0;		/* 2 */	
)		/* 3 */	
void funk()	/* 2 */		
t char v3; return;			
}			

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

1	1	1

Stack - the memory in which local variables exist in their "scope", is often limited.

int v1 = 42;			hea
	/* O */		
int main() {	/* 0 */ /* 1 */		
int v2 = 4;			stack ->
funk();		/* O */	
	/* 3 */	/* 1 */	
return 0;		/* 2 */	
}		/* 3 */	
void funk() {	/* 2 */		
char v3; return;			

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->	Whats in the stack at 0 ?		

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>	
int v1 = 42;	
<pre>int main() { int v2 = 4;</pre>	/* 0 */ /* 1 */
funk();	
return 0; }	/* 3 */
<pre>void funk() { char v3; return; }</pre>	/* 2 */

heap

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->

/* 0 */

/* 1 */

/* 2 */

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>		
int v1 = 42;		
<pre>int main() { int v2 = 4;</pre>	/* 0 */ /* 1 */	
funk();		/* O */
return 0; }	/* 3 */	/* 1 */ /* 2 */ /* 3 */
<pre>void funk() { char v3; return; }</pre>	/* 2 */	

heap

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack -> Whats in the stack at 1?

v1		
	I	I

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>		
int v1 = 42;		
int main() { int v2 = 4;	/* 0 */ /* 1 */	
funk();		
return 0; }	/* 3 */	
<pre>void funk() { char v3; return; }</pre>	/* 2 */	

heap

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->

/* 0 */

/* 1 */

/* 2 */

v1		
v1	v2	

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>			
int v1 = 42;			
	/* 0 */		
int main()	/* 1 */		
{ int v2 = 4;			S
funk();		/* O */	
	/* 3 */	/* 1 */	
return 0;		/* 2 */	
}		/* 3 */	
void funk()	/* 2 */		
{ char v3;			
return;			
}			

	,		
heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	
stack ->	Whats in the stack	at 2 ?	

stack -> Whats in the stack at 2?			
v1			
v1	v2		

Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h> int v1 = 42; **/*** 0 ***/ /* 1 */** int main() int v2 = 4; funk(); **/* 3 */** return 0; void funk() **/* 2 */** char v3; return;

heap

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->

/* 0 */

/* 1 */

/* 2 */

v1			
v1	v2		
v1	v2	v3	

Stack - the memory in which local variables exist in their "scope", is often limited.

/* 0 */

/* 1 */

/* 2 */

/* 3 */

#include <stdio.h></stdio.h>	
int v1 = 42;	
<pre>int main() { int v2 = 4;</pre>	/* 0 */ /* 1 */
funk();	
return 0; }	/* 3 */
<pre>void funk() { char v3; return; }</pre>	/* 2 */

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->	Whats in the stack	Whats in the stack at 3?	
v1			
v1	v2		
v1	v2	v3	

Heap - the memory where the program code, global, static variables and dynamic allocated data is, size depends on platform. Stack - the memory in which local variables exist in their "scope", is often limited.

#include <stdio.h></stdio.h>	
int v1 = 42;	
	/* 0 */
int main()	/* 1 */
int v2 = 4;	
funk();	
	/* 3 */
return 0; }	
void funk()	/* 2 */
{ char v3;	
return;	
}	

heap

heapVariabel			
main	main kod	main kod	main kod
funk	funk kod	funk kod	

stack ->

/* 0 */

/* 1 */

/* 2 */

v1			
v1	v2		
v1	v2	v3	
v1	v2		

