Minne - heap - stack

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>

int v1 = 42;

                                /* 0 */
int main()                      /* 1 */
{
    int v2 = 4;

    funk();

                                /* 3 */
    return 0;
}

void funk()                     /* 2 */
{
    char v3;
    return;
}
```

Whats in the heap?

heap

stack ->

/* 0 */

/* 1 */

/* 2 */

/* 3 */

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>

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 */

/* 3 */

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>

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 ->

Whats in the stack at 0 ?

/* 0 */

/* 1 */

/* 2 */

/* 3 */

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>

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 */

/* 3 */

v1			

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>

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 ->

Whats in the stack at 1 ?

/* 0 */

/* 1 */

/* 2 */

/* 3 */

v1			

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>

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 */

/* 3 */

v1			
v1	v2		

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>

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 ->

Whats in the stack at 2 ?

/* 0 */

/* 1 */

/* 2 */

/* 3 */

v1			
v1	v2		

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>

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 */

/* 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>

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 ->

Whats in the stack at 3 ?

/* 0 */	v1			
/* 1 */	v1	v2		
/* 2 */	v1	v2	v3	
/* 3 */				

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>

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 */	v1			
/* 1 */	v1	v2		
/* 2 */	v1	v2	v3	
/* 3 */	v1	v2		

https://github.com/michellundell/2b-heap-stack