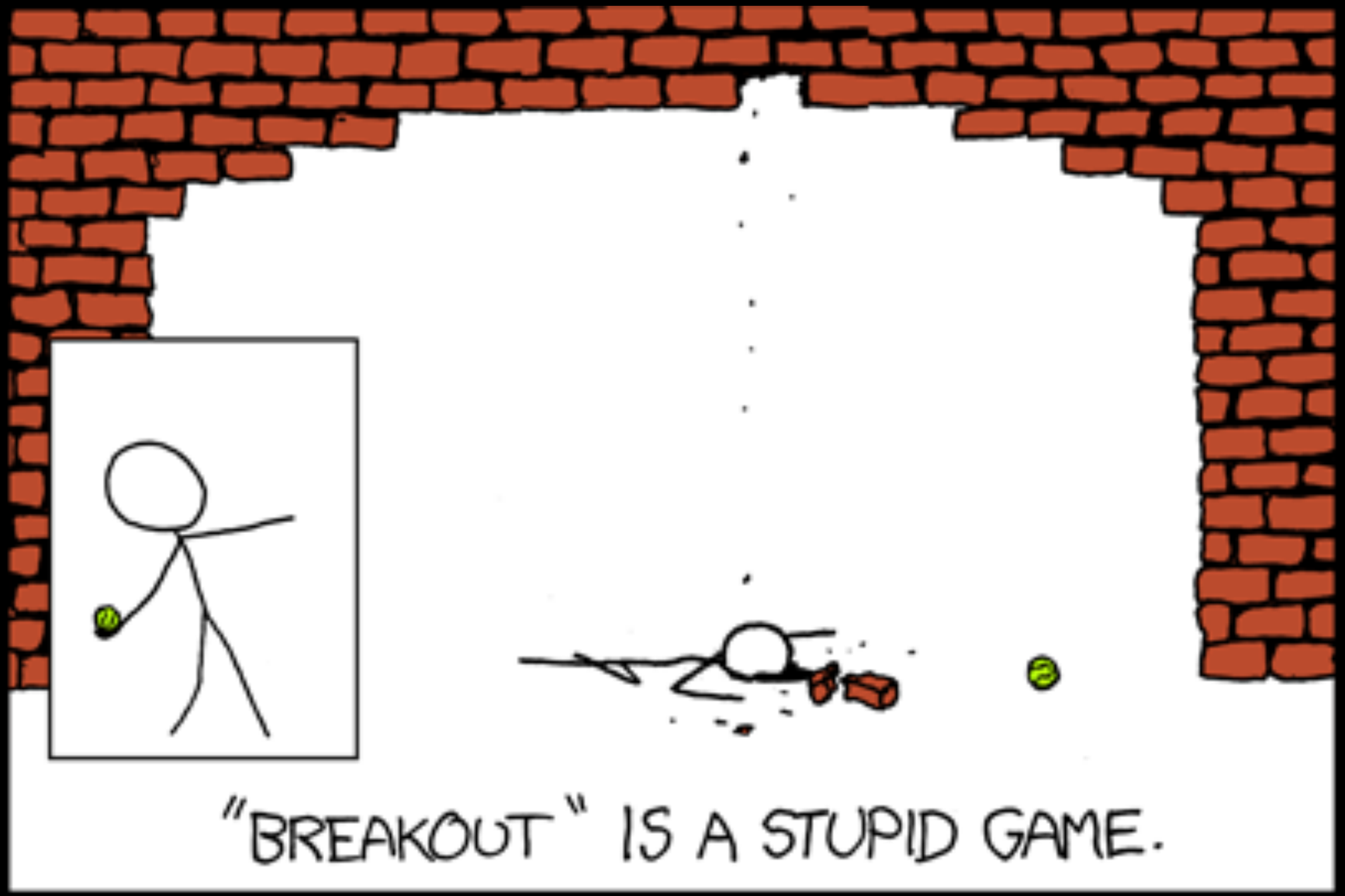


http://xkcd.com/347/

week 5

string

char*

```
string s = GetString();
string t = GetString();

if (s == t)
{
    printf("You typed the same thing!\n");
}
else
{
    printf("You typed different things!\n");
}
```

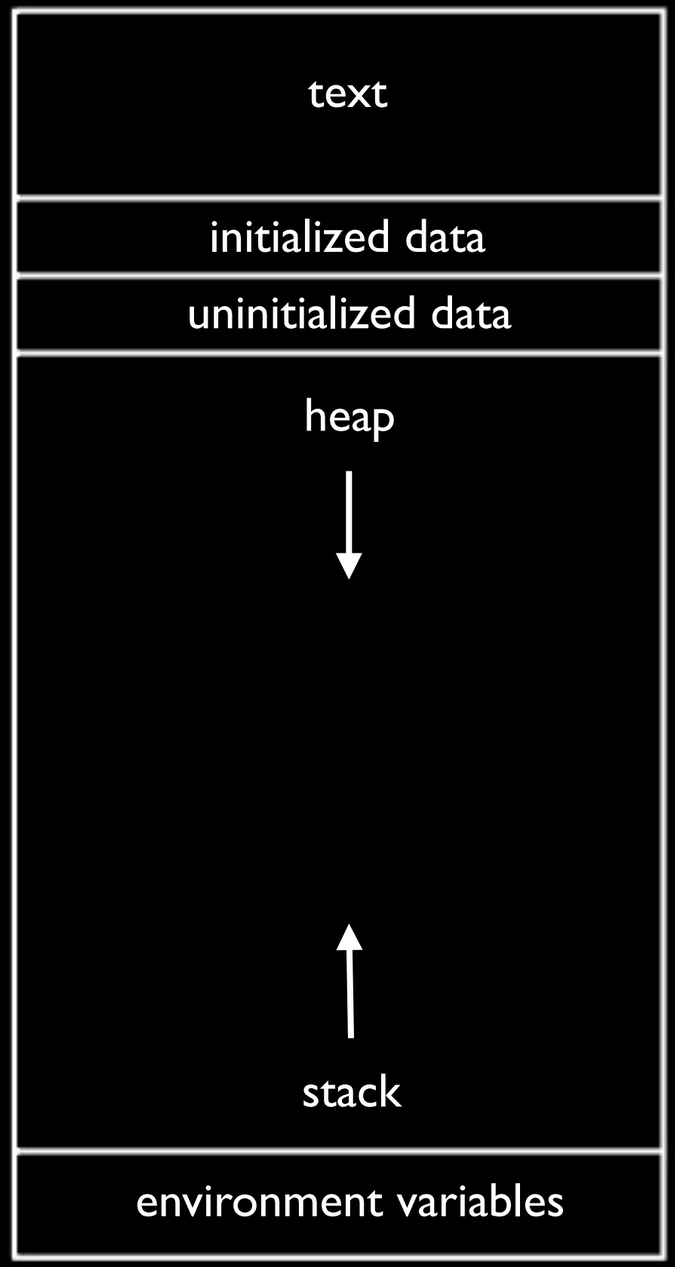
```
char* s = GetString();
char* t = GetString();

if (s != NULL && t != NULL)
{
    if (strcmp(s, t) == 0)
    {
        printf("You typed the same thing!\n");
    }
    else
    {
        printf("You typed different things!\n");
    }
}
```

```
string s = GetString();
if (s == NULL)
{
    return 1;
}
string t = s;
if (strlen(t) > 0)
{
    t[0] = toupper(t[0]);
}
printf("Original: %s\n", s);
printf("Copy:     %s\n", t);
```

```
char* s = GetString();
...
char* t = malloc((strlen(s) + 1) * sizeof(char));
...
for (int i = 0, n = strlen(s); i <= n; i++)
{
    t[i] = s[i];
}
...
```

```
void swap(int a, int b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
```

```
void swap(int* a, int* b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

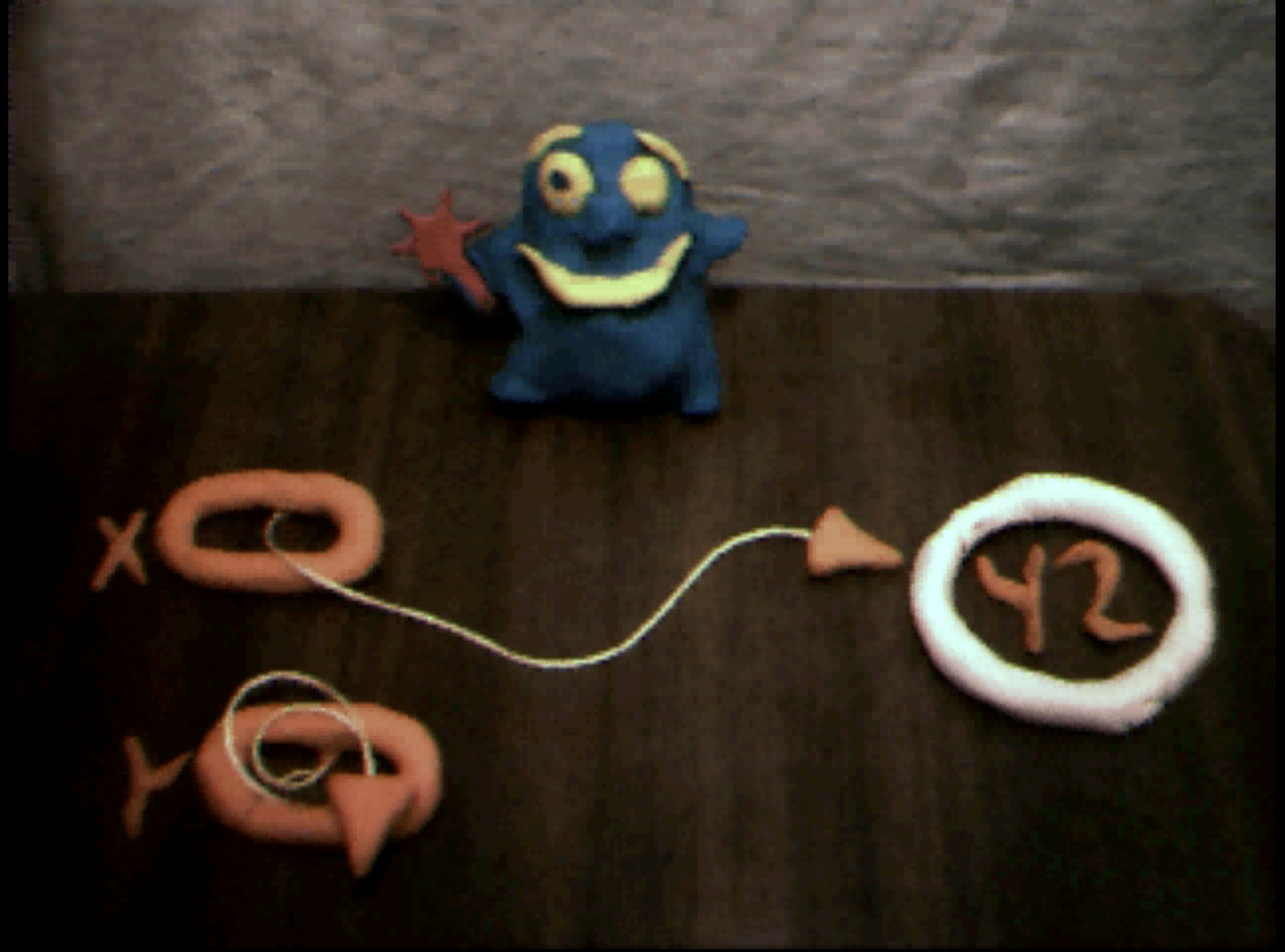

text
initialized data
uninitialized data
heap
stack
environment variables

malloc

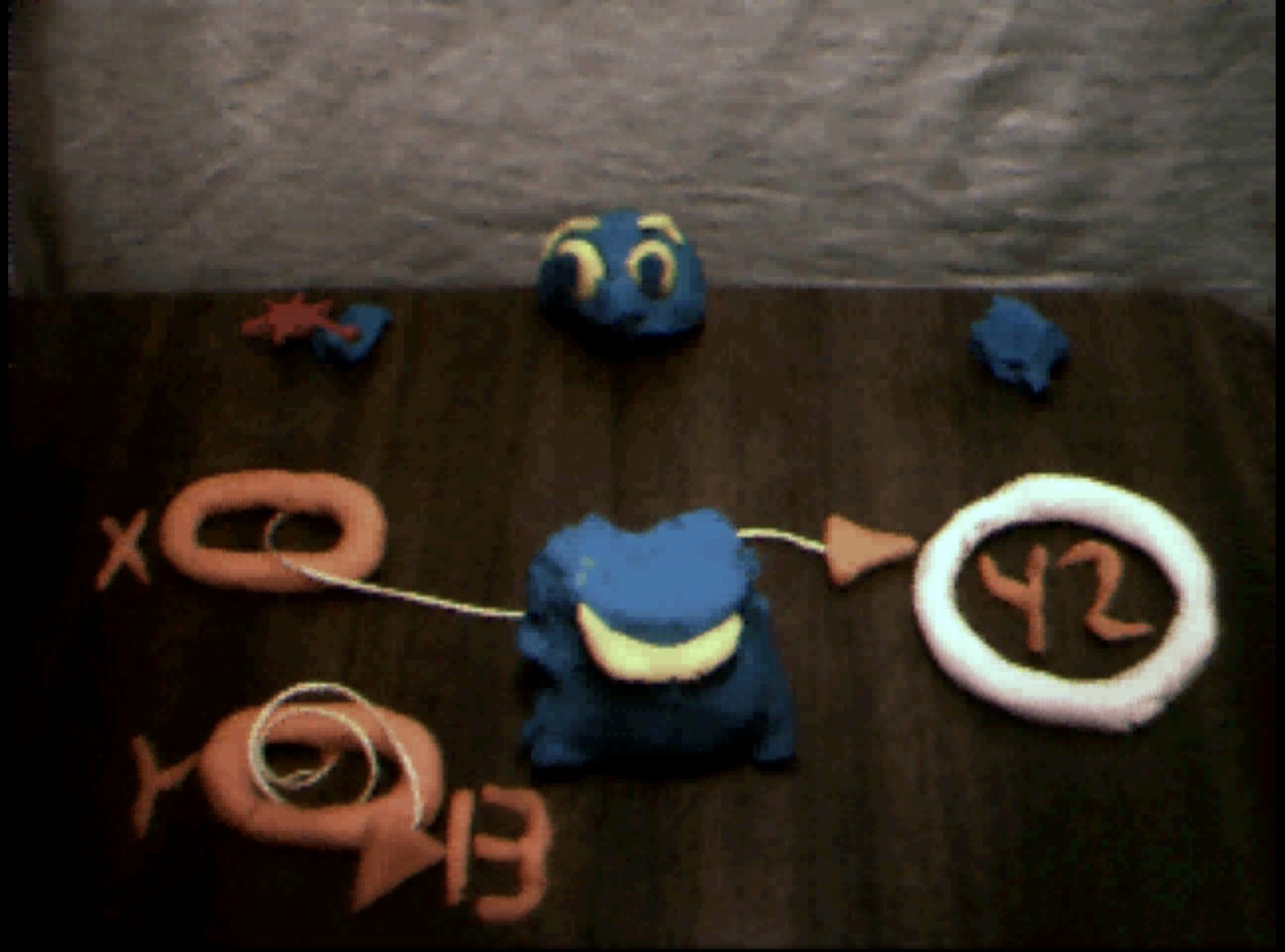
```
int main(void)
{
    int* x;
    int* y;

    x = malloc(sizeof(int));

    *x = 42;

    *y = 13;

    y = x;

    *y = 13;
}
```

```
int* x;
int* y;
```

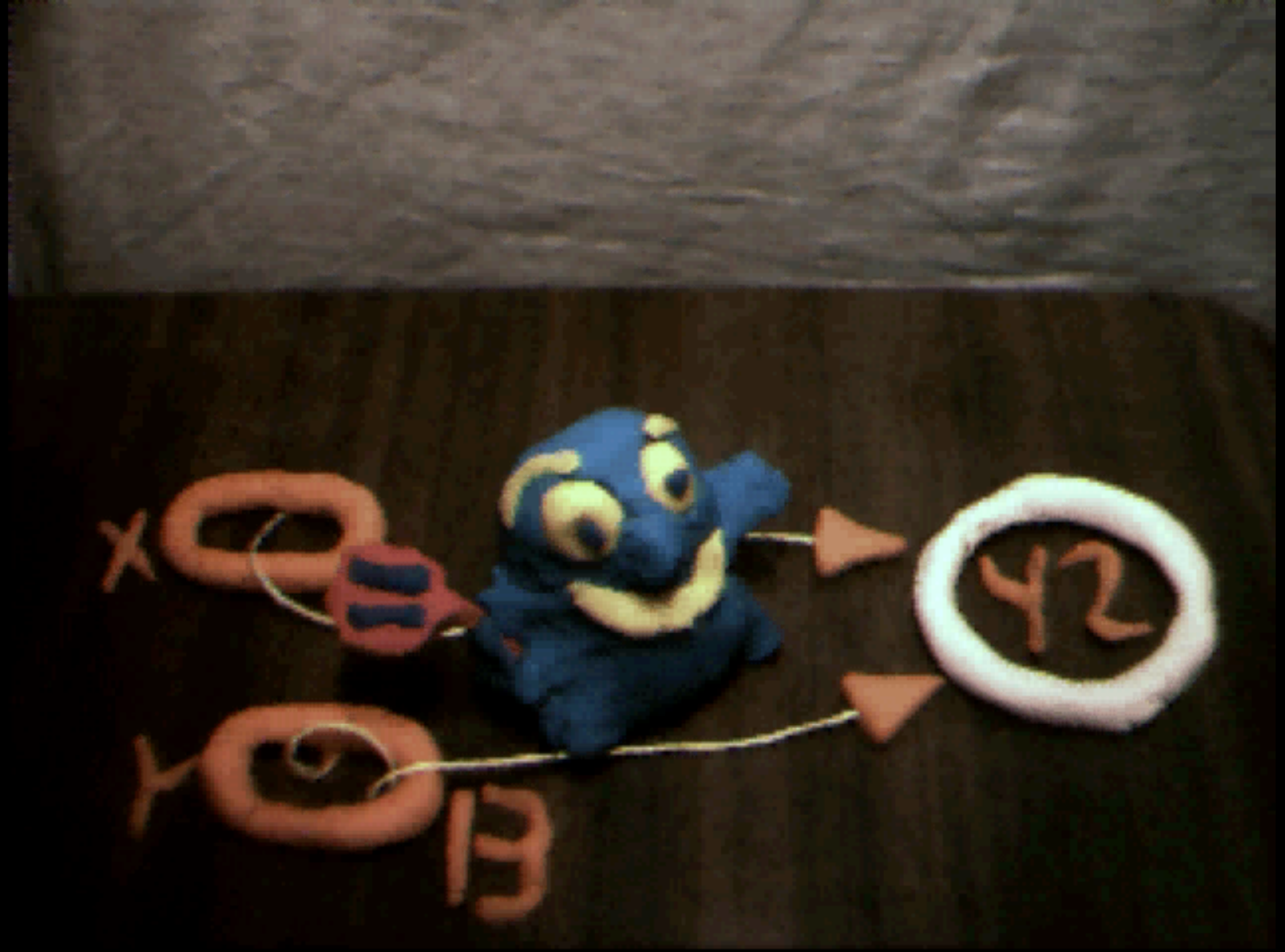
```
x = malloc(sizeof(int));
```



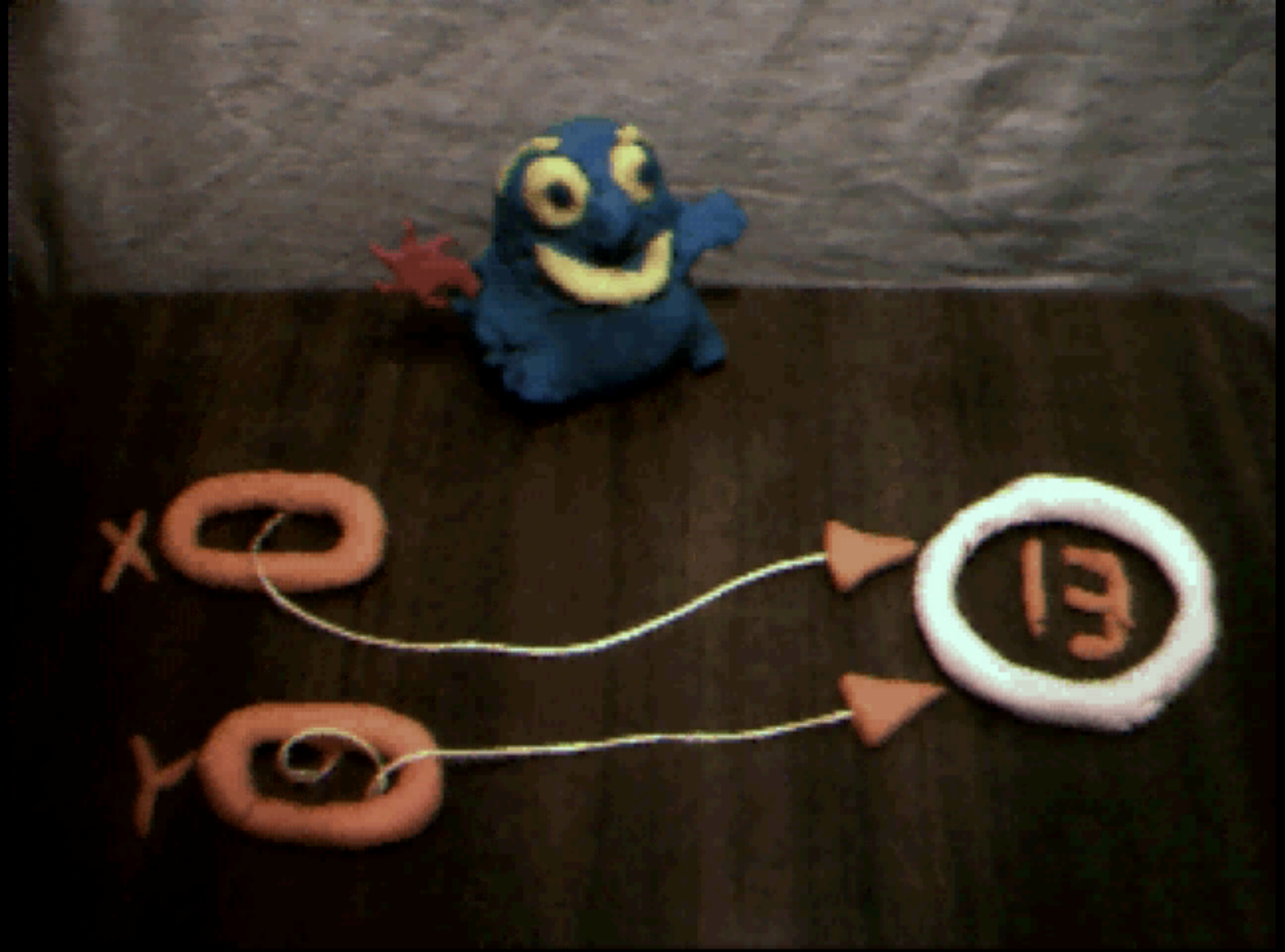

```
*x = 42;
```




```
*y = 13;
```




```
y = x;
```



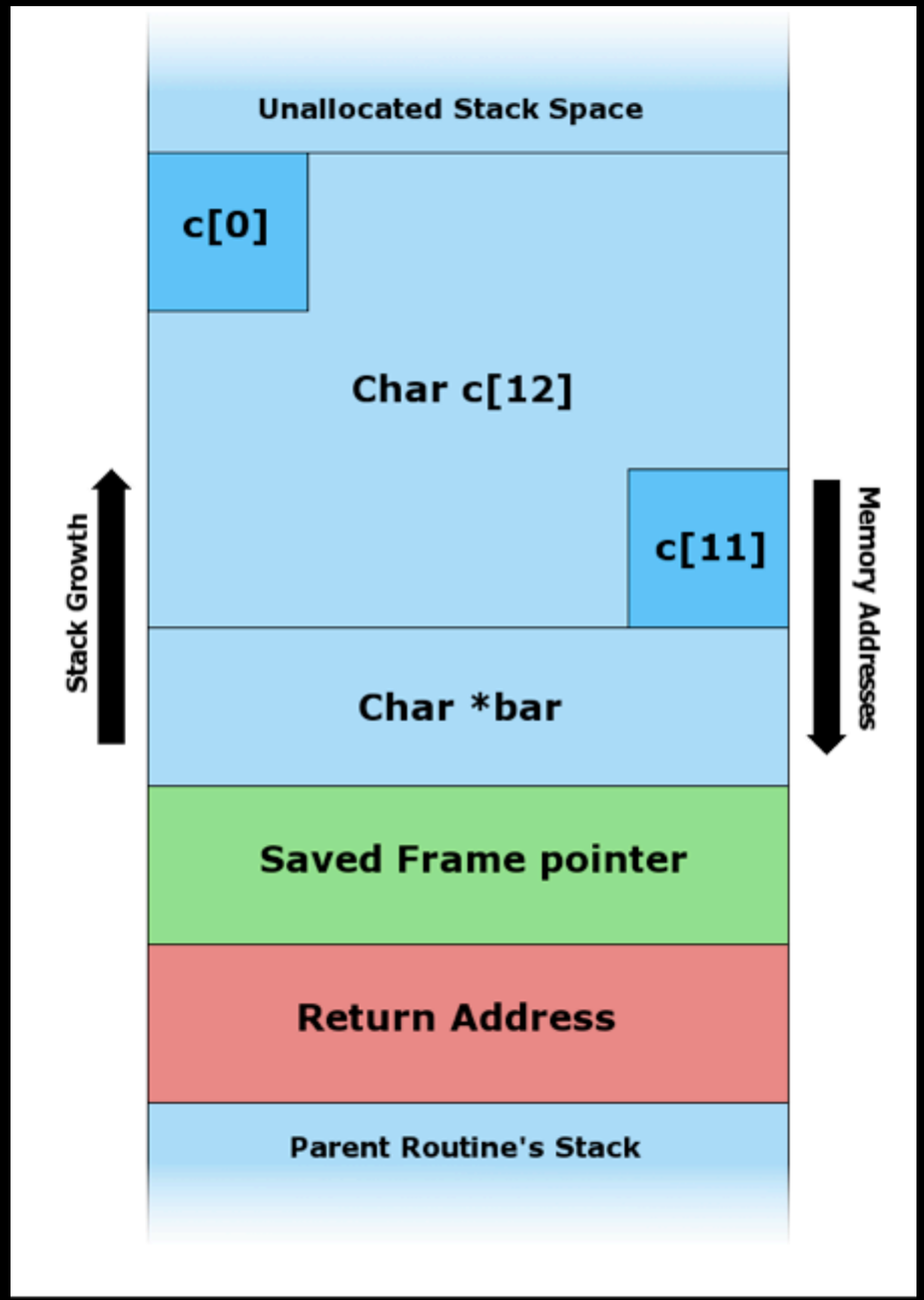

```
*y = 13;
```

```c
int main(void)
{
    int* x; // allocate the pointers x and y
    int* y; // (but not the pointees)

    x = malloc(sizeof(int)); // allocate an int pointee,
                             // and set x to point to it

    *x = 42; // dereference x to store 42 in its pointee

    *y = 13; // CRASH -- y does not have a pointee yet

    y = x; // pointer assignment sets y to point to x's pointee

    *y = 13; // dereference y to store 13 in its (shared) pointee
}
```

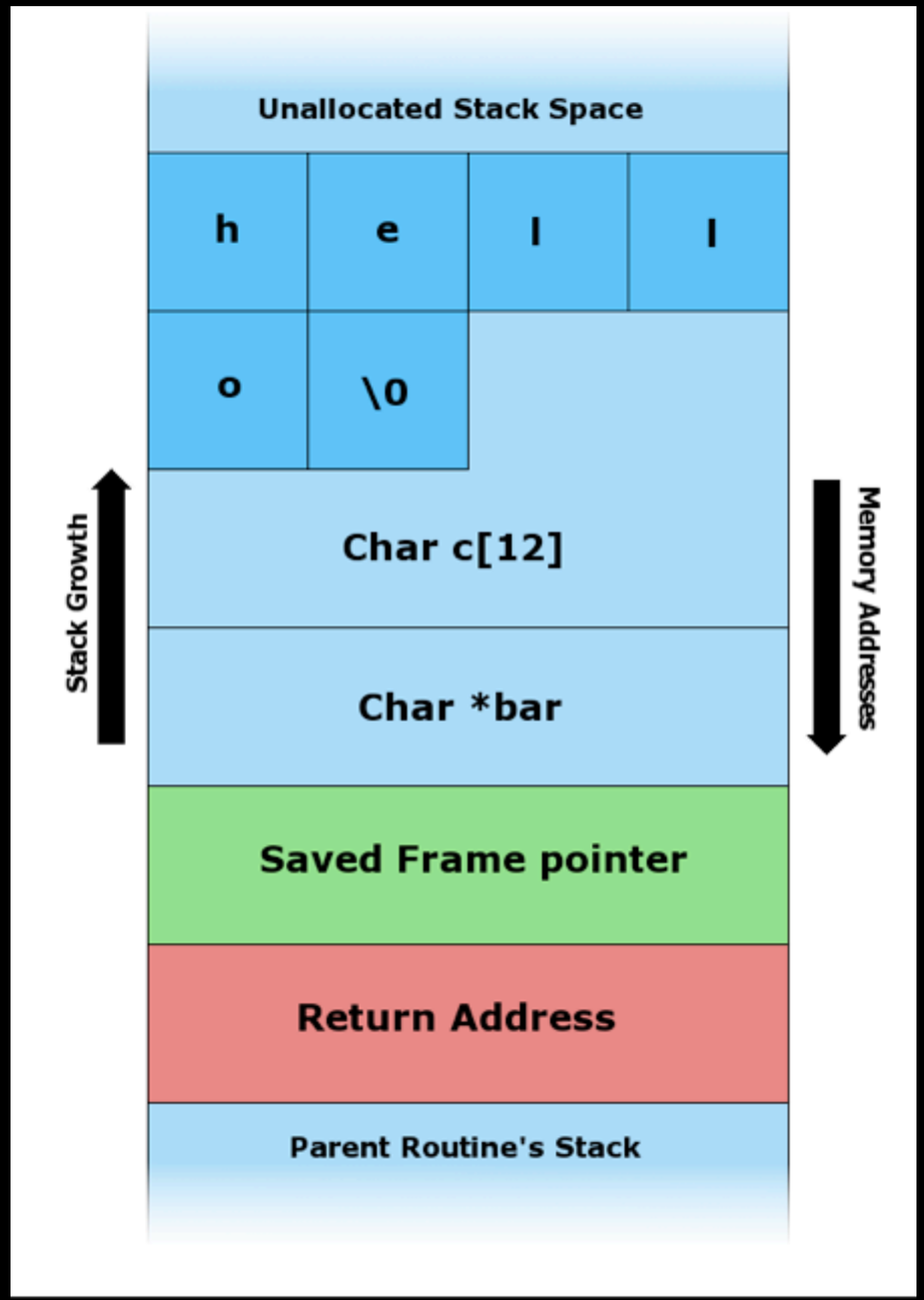
stack overflow

```
#include <string.h>

void foo(char* bar)
{
    char c[12];
    memcpy(c, bar, strlen(bar));
}

int main(int argc, char* argv[])
{
    foo(argv[1]);
}
```

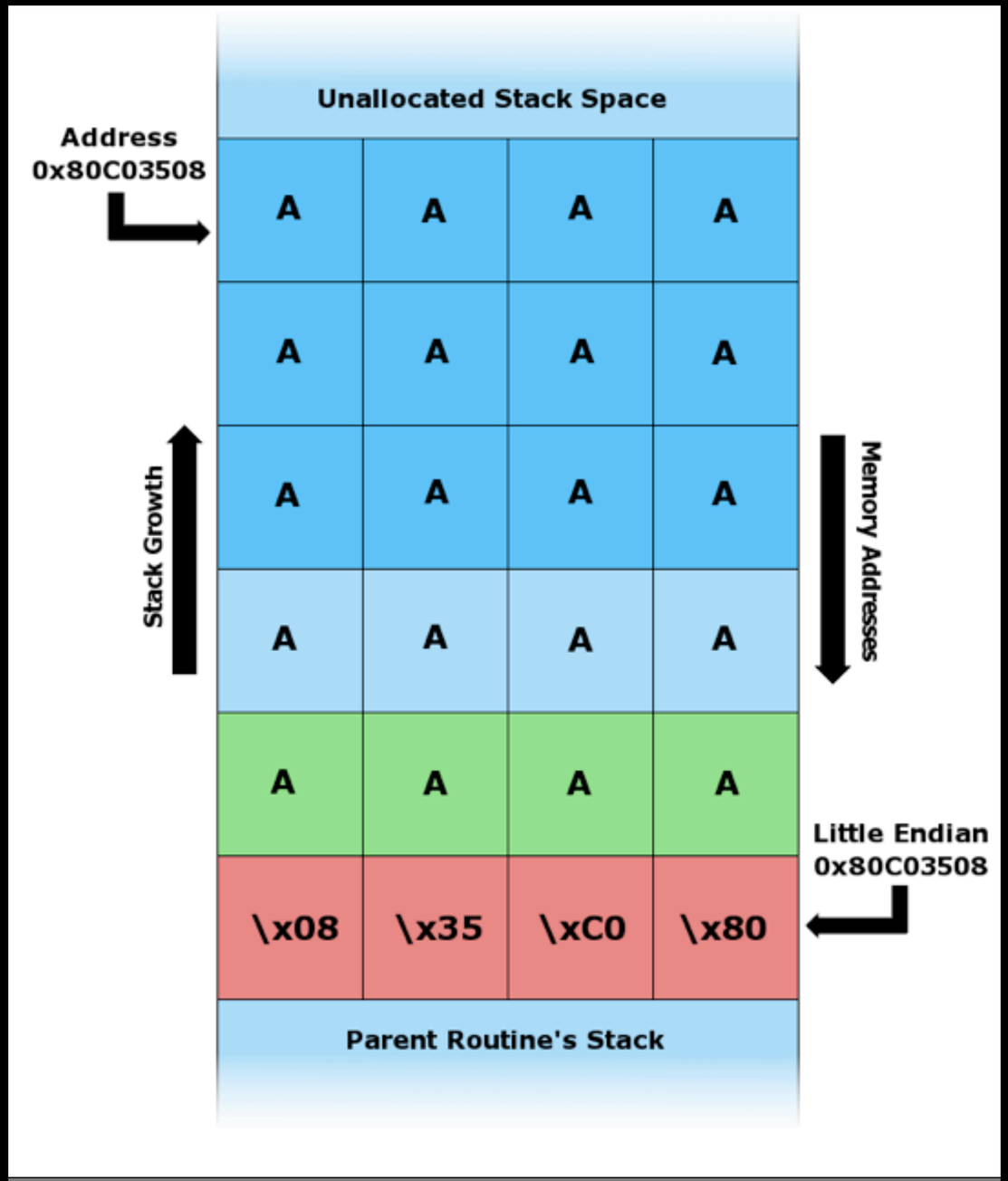
adapted from http://en.wikipedia.org/wiki/Stack_buffer_overflow



http://en.wikipedia.org/wiki/Stack_buffer_overflow



http://en.wikipedia.org/wiki/Stack_buffer_overflow



http://en.wikipedia.org/wiki/Stack_buffer_overflow

http://xkcd.com/138/

to be continued...