

```
#include <unistd.h>
```

```
void print_hex(unsigned char m)
{
    char values[16] = "0123456789abcdef";
    char trsl[2] = {0};
    int i = 1;

    if (!m)
    {
        write(1, "00", 2);
    }
    else
    {
        while (i >= 0)
        {
            trsl[i] = values[m % 16];
            m /= 16;
            i--;
        }
        write(1, trsl, 2);
    }
}
```

```
void print_ascii(unsigned char m)
{
    if (m >= 32 && m <= 126)
        write(1, &m, 1);
    else
        write(1, ".", 1);
}
```

```
int calc_pad(int pos)
{
    int i = 0;
    while (pos % 16)
    {
        pos++;
        i += 2;
    }
    i += i / 4;
    return (i);
}
```

```
void print_pad(int i)
{
    while (i > 0)
    {
        write(1, " ", 1);
        i--;
    }
}

void print_memory(const void *addr, size_t size)
{
    unsigned char *ptr;
    int i = 0;
    int count_pass;
    int tcpt;
    ptr = (unsigned char *)addr;
    while (i < (int)size)
    {
        count_pass = 0;
        tcpt = i;
        while (tcpt < (int)size && count_pass < 16)
        {
            print_hex(ptr[tcpt]);
            tcpt++;
            count_pass++;
            if (tcpt < (int)size)
            {
                print_hex(ptr[tcpt]);
                count_pass++;
                tcpt++;
            }
            write(1, " ", 1);
        }
        print_pad(calc_pad(count_pass));
        count_pass = 0;
        tcpt = i;
        while (tcpt < (int)size && count_pass < 16)
        {
            print_ascii(ptr[tcpt]);
            count_pass++;
            tcpt++;
        }
        write(1, "\n", 1);
        i += count_pass;
    }
}
```