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

#define N 16

void ft_int_to_hex(char *hex, long long int dir, int *size)
{
    long long int temp;

    *size=0;
    while (dir != 0)
    {
        temp = dir % N;
        if (temp < 10)
            temp = temp + '0';
        else
            temp = temp + 'a' - 10;
        hex[*size] = temp;
        (*size)++;
        dir = dir / N;
    }
}

void    ft_string(void *addr,int i,int j)
{
    char *p;
    int k;
    char hex[3];
    int tam;

    p = (char*)addr;
    k = i;
    while (k < j)
    {
        ft_int_to_hex(hex,(long long int)p[k],&tam);
        write(1,&hex[1],1);
        write(1,&hex[0],1);
        if(k % 2 == 1)
            write(1," ",1);
        k++;
    }
    while(i < j)
    {
        if(*(p+i)<=31 || *(p+i)==127)
            write(1,".",1);
        else
            write(1,p+i,1);
        i++;
    }
    write(1,"\n",1);
}

void ft_print_pointer(void *addr,int i)
{
    char hex[N];
    long long int dir;
    int hex_size;
    int k;

    dir = (long long int)(addr+i);
    ft_int_to_hex(hex,dir,&hex_size);
    k = 0;
    while (k + hex_size < N)
    {
        write(1,"0",1);
        k++;
    }
    k = 0;
    while (k < hex_size)
    {
        write(1,&hex[hex_size - k -1],1);
        k++;
```

```
    }
    write(1,": ",2);
}

void    *ft_print_memory(void *addr, unsigned int size)
{
    unsigned int j;
    unsigned int i;
    int finish;

    i = 0;
    j = 0;
    finish = 0;
    while(i < size && finish == 0)
    {
        if (i+16<size)
            j = j + 16;
        else
        {
            j = size - 1;
            finish = 1;
        }
        ft_print_pointer(addr,i);
        ft_string(addr,i,j);
        i = j;
    }
    return (addr);
}

int main()
{
    char str[]="Bonjour les aminches\n.\tc. est fou.tout\0ce qu on peur faire
avec...print memory\t\t..lol.lol. .";
    void *addr;
    addr = str;
    unsigned int size=sizeof(str);
    ft_print_memory(addr,size);
    return 0;
}
```