```c
#include <unistd.h>

#define is_alpha(c) (c >= 'a' && c <='z') ? 1 : 0
enum status{INVALID, HELP, SUCESS};
int     g_mem;

void    print_bin(int num)
{
        long r = 1;
        r <<= 32;
        char count = 1;
        while (r >>= 1)
        {
                (r & num) ? write(1, "1", 1) : write(1, "0", 1);
                if (count % 8 == 0 && count != 32)
                        write(1, " ", 1);
                count++;
        }
}

int             check_flags(char *str)
{
        unsigned i = 0;
        if (str[i] != '-')
                return (INVALID);
        while (str[++i])
                if (!is_alpha(str[i]))
                        return (INVALID);
        i = 1;
        while (str[i])
        {
                if (str[i] == 'h')
                        return (HELP);
                g_mem |= (1 << (str[i] - 'a'));
                i++;
        }
        return (SUCESS);
}
```

```c
int             main(int ac, char **av)
{
        unsigned i = 1;
        unsigned char status = 0;

        if (ac < 2)
        {
                write(1, "options: abcdefghijklmnopqrstuvwxyz\n", 36);
                return (0);
        }
        else
        {
                while (av[i])
                {
                        status = check_flags((av[i]));
                        if (status == INVALID)
                        {
                                write(1, "Invalid Option\n", 15);
                                return (0);
                        }
                        else if (status == HELP)
                        {
                                write(1, "options: abcdefghijklmnopqrstuvwxyz\n", 36);
                                return (0);
                        }
                        i++;
                }
                print_bin(g_mem);
        }
        return (0);
}
```