

Nom: Souala
Prenom:Elhoussine
Groupe:3

1.

```
#include <stdlib.h>
```

```
char * tab;  
int main() {  
    tab = malloc(16384 * sizeof(char));  
    for (int i = 0; i < 16384; i++) tab[i] = 'a';  
    return 0;  
}
```

A compiler avec - std = c99:

```
gcc - std = c99 - static exo3m.c - o exo3m
```

2.

```
#include <sys/types.h>
```

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

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

```
#include <stdlib.h>
```

```
#include <sys/mman.h>
```

```
#include <malloc.h>
```

```
char * tab;  
char * alloc;
```

```

int pagesize;
int main() {
    alloc = malloc(1024 L * 1024 L);
    tab = malloc(16384 * sizeof(char));
    for (int i = 0; i < 16384; i++) tab[i] = 'a';
    pagesize = sysconf(_SC_PAGE_SIZE);
    printf("début du tableau: 0x%lx\n", (long) tab);
    tab = memalign(pagesize, 4 * pagesize);
    mprotect(tab + pagesize, 2 * pagesize, PROT_WRITE);
    /* la mémoire peut être
accédée uniquement écriture*/
    /*Pour la lecture : PROT_READ*/
    printf("pagesize: %d\n", pagesize);
    printf("%c\n", tab[4098]);
    /* Segmentation fault, car la mémoire est protégée contre la
lecture*/
    /* pour tester la protection contre l'écriture, on peut mettre tab[4098]='b';*/
    sprintf(alloc, "cat /proc/%d/maps", getpid());
    system(alloc);
    return 0;
}

```

3.

Carte mémoire après la protection :

```

08048000-080ea000 r-xp 00000000 08:06 437065 /home/belfedhal/exo3m
080ea000-080ec000 rw-p 000a1000 08:06 437065 /home/belfedhal/exo3m
080ec000-080ee000 rw-p 00000000 00:00 0
08f9f000-08fa6000 rw-p 00000000 00:00 0 [heap]
08fa6000-08fa8000 ---p 00000000 00:00 0 [heap]
08fa8000-08fc1000 rw-p 00000000 00:00 0 [heap]

```

b7669000-b776b000 rw-p 00000000 00:00 0

b776b000-b776d000 r--p 00000000 00:00 0 [vvar]

b776d000-b776f000 r-xp 00000000 00:00 0 [vdso]

bfd8d000-bfdde000 rw-p 00000000 00:00 0 [stack]

Avec l'option PROT_NONE de la fonction mprotect, on a enlevé toutes les autorisation (lecture, écriture et exécution)