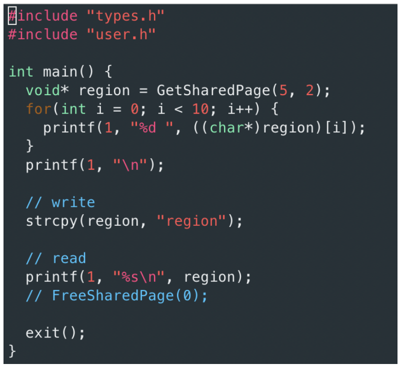
Timur Gaimakov: A20415319

Hassan Alamri: A204730047

Test Cases

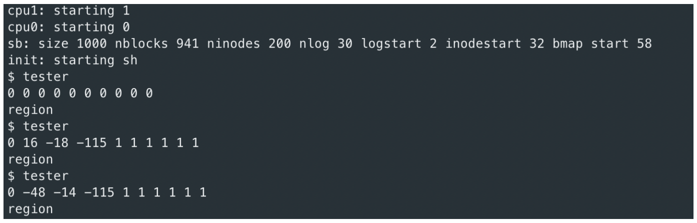
Test case 1

GetSharedPage(5, 2)



```c
#include "types.h"
#include "user.h"

int main() {
  void* region = GetSharedPage(5, 2);
  for(int i = 0; i < 10; i++) {
    printf(1, "%d ", ((char*)region)[i]);
  }
  printf(1, "\n");

  // write
  strcpy(region, "region");

  // read
  printf(1, "%s\n", region);
  // FreeSharedPage(0);

  exit();
}
```

Result:

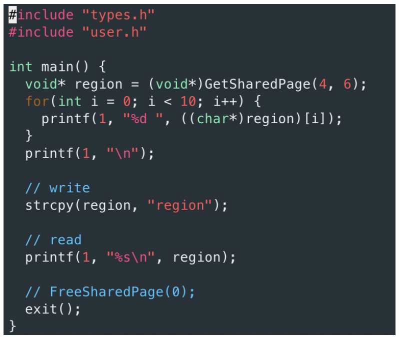


```
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester
0 0 0 0 0 0 0 0 0 0
region
$ tester
0 16 -18 -115 1 1 1 1 1 1
region
$ tester
0 -48 -14 -115 1 1 1 1 1 1
region
```

Explanation: this test case shows a different set of pages that form a virtual address. What is interesting is that the pages change after each call of tester.c

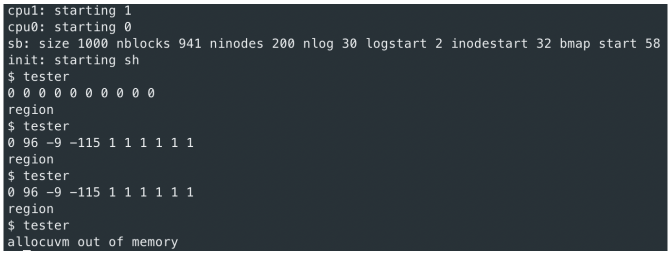
Test case 2

GetSharedPage(4, 6)



```c
#include "types.h"
#include "user.h"

int main() {
  void* region = (void*)GetSharedPage(4, 6);
  for(int i = 0; i < 10; i++) {
    printf(1, "%d ", ((char*)region)[i]);
  }
  printf(1, "\n");

  // write
  strcpy(region, "region");

  // read
  printf(1, "%s\n", region);

  // FreeSharedPage(0);
  exit();
}
```

Result:

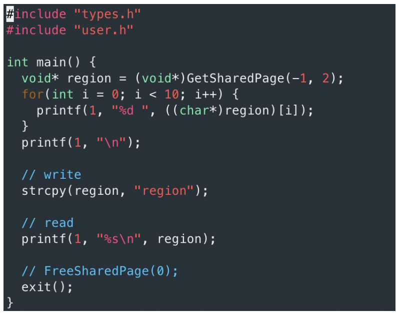


```
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester
0 0 0 0 0 0 0 0 0 0
region
$ tester
0 96 -9 -115 1 1 1 1 1 1
region
$ tester
0 96 -9 -115 1 1 1 1 1 1
region
$ tester
allocuvm out of memory
```

Explanation: this test case has a different key number, and a number of pages that exceed key. It prints out only 1 page and this page stays the same.

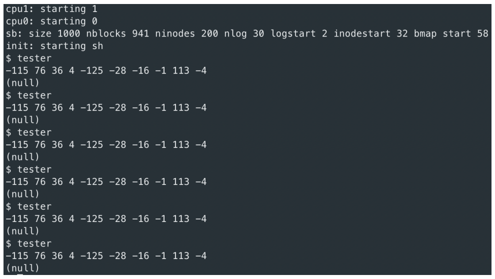
Test case 3

GetSharedPage(-1, 2)



```
#include "types.h"
#include "user.h"

int main() {
  void* region = (void*)GetSharedPage(-1, 2);
  for(int i = 0; i < 10; i++) {
    printf(1, "%d ", ((char*)region)[i]);
  }
  printf(1, "\n");

  // write
  strcpy(region, "region");

  // read
  printf(1, "%s\n", region);

  // FreeSharedPage(0);
  exit();
}
```

Result:

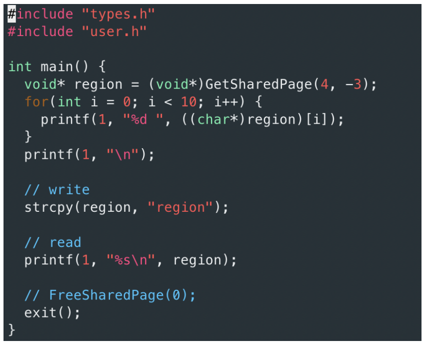


```
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
$ tester
-115 76 36 4 -125 -28 -16 -1 113 -4
(null)
```

Explanation: this test case shows the outcome if the key is set to be negative. For some reason, the code returns a virtual address with a negative key value.

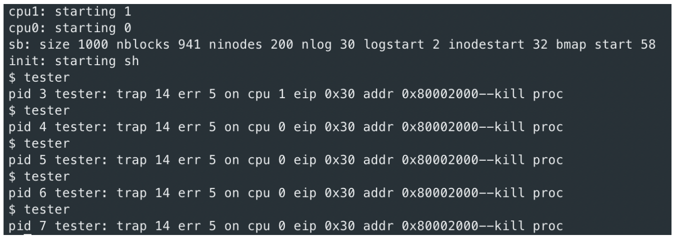
Test case 4

GetSharedPage(4, -3)



```
#include "types.h"
#include "user.h"

int main() {
  void* region = (void*)GetSharedPage(4, -3);
  for(int i = 0; i < 10; i++) {
    printf(1, "%d ", ((char*)region)[i]);
  }
  printf(1, "\n");

  // write
  strcpy(region, "region");

  // read
  printf(1, "%s\n", region);

  // FreeSharedPage(0);
  exit();
}
```

Result:



```
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester
pid 3 tester: trap 14 err 5 on cpu 1 eip 0x30 addr 0x80002000--kill proc
$ tester
pid 4 tester: trap 14 err 5 on cpu 0 eip 0x30 addr 0x80002000--kill proc
$ tester
pid 5 tester: trap 14 err 5 on cpu 0 eip 0x30 addr 0x80002000--kill proc
$ tester
pid 6 tester: trap 14 err 5 on cpu 0 eip 0x30 addr 0x80002000--kill proc
$ tester
pid 7 tester: trap 14 err 5 on cpu 0 eip 0x30 addr 0x80002000--kill proc
```

Explanation: this test case the outcome if the key is positive but num_of_pages is set to be negative. The result is an error that shows that num_of_pages is negative.