```c
// Pointers and arrays

#include <stdio.h>

int main()
{
int s[3];

    s[0] = 55;
    s[1] = 22;
    s[2] = 30;

    printf("%d\n",s[0]);
    printf("%d\n",s[1]);
    printf("%d\n",s[2]);

    printf("%d\n",*s);
    printf("%d\n",*(s+1));
    printf("%d\n",*(s+2));
/*

    *s = 1000;
    *(s+1) = 2000;
    *(s+2) = 3000;

    printf("%d\n",*s);
    printf("%d\n",*(s+1));
    printf("%d\n",*(s+2));
*/

}
~
"main.c" 31L, 378C written
pod1-1 10 % gcc main.c
pod1-1 11 % a.out
55
22
30
55
22
pod1-1 12 %
```

this → this s is little → 5000 | 1000 |

→ s is not. variable

S[0] = 55

| 1011 |
| 1010 |
| 1009 |
S[2] → 1008 | 20 | ] 4 bytes

*s
↓

s is address

a variable  *(s+1)

| 1007 |
| 1006 |
| 1005 |
S[1] → 1004 | 22 | ] 4 bytes
22        | 1003 |
| 1002 |
| 1001 |  ] int 4 bytes

1000+4
= 1004   S[0] → 1000 | 55 |

size of (int)



```
#include <stdio.h>

int main()
{
int s[3];

s[0] = 55;
s[1] = 22;
s[2] = 30;

printf("%d\n",s[0]);
printf("%d\n",s[1]);
printf("%d\n",s[2]);

printf("%d\n", *s);
printf("%d\n", *(s+1));
printf("%d\n", *(s+2));

*s = 1000;
*(s+1) = 2000;
*(s+2) = 3000;

printf("%d\n",s[0]);
printf("%d\n", *(s+1));
printf("%d\n",s[2]);

}
```

"main.c" 29L, 372C written
pod1-1 14 %



```
printf("%d\n",s[0]);
printf("%d\n",s[1]);
printf("%d\n",s[2]);

printf("%d\n", *s);
printf("%d\n", *(s+1));
printf("%d\n", *(s+2));

*s = 1000;
*(s+1) = 2000;
*(s+2) = 3000;

printf("%d\n",s[0]);
printf("%d\n", *(s+1));
printf("%d\n",s[2]);
}
```

"main.c" 29L, 372C written
pod1-1 14 % gcc main.c
pod1-1 15 % a.out
55
22
30
55
22
30
1000
2000
3000
pod1-1 16 %

```
// Pointers and arrays

#include <stdio.h>

void main()
{
int *h;

*h = 100;
*(h+1) = 200;
*(h+2) = 300;

printf("%d\n",*h);
printf("%d\n",*(h+1));
printf("%d\n",*(h+2));

}
```
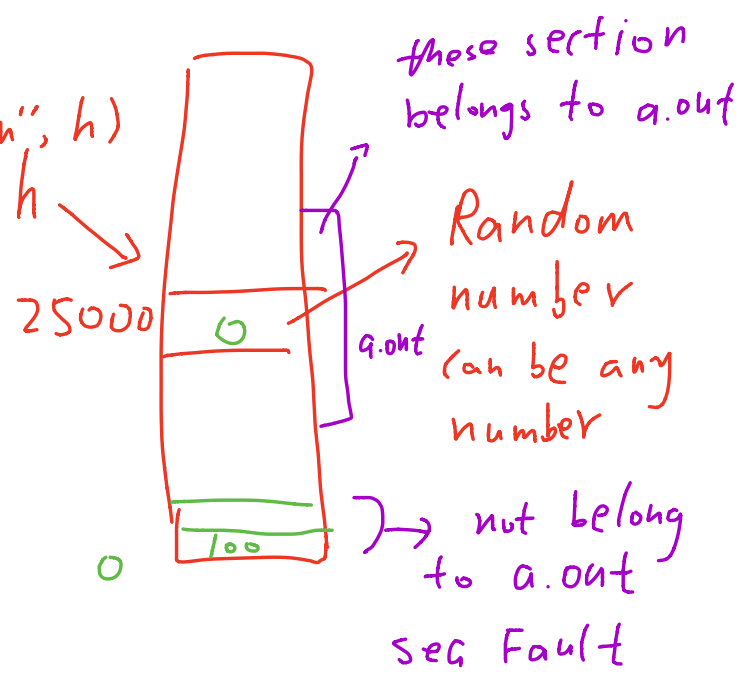"main.c" 17L, 186C

printf("%p\n", h)

↓ Result

(nil)

↓ 0

Seg fault

these section belongs to a.out

h → 25000

Random number can be any number

a.out

0

not belong to a.out

seg Fault

0    100

---



```
#include <stdio.h>

int main()
{
int i;
int a[5];

    for(i=0; i<5; i++)
        a[i] = i;

    for(i=0; i<5; i++)
        printf("%d\n",a[i]);
//      printf("%d\n",*(a+i));

    for(i=0; i<6; i++)
        a[i] = i;

    for(i=0; i<6; i++)
        printf("%d\n",a[i]);

}
```
"main.c" 26L, 327C written
pod1-1 41 % !gcc
gcc main.c
pod1-1 42 %

print
0
1
2
3
4
0
1
2
3
4
5

← Bug

not crush
Bad luck

Because a[5] still belong to our process in this running time
if it belong to us, this a[5] may contain other contents, will be overwrite

```
// Basic string processing.

#include <stdio.h>

int main()
{
int i;
char a[6];

    a[0] = 'a';
    a[1] = 'b';
    a[2] = '\0';
    a[3] = 'd';
    a[4] = 'e';
    a[5] = '\0';

    printf("%s\n",a);

}
```

→ end of string

print

a b

hot

a b   de