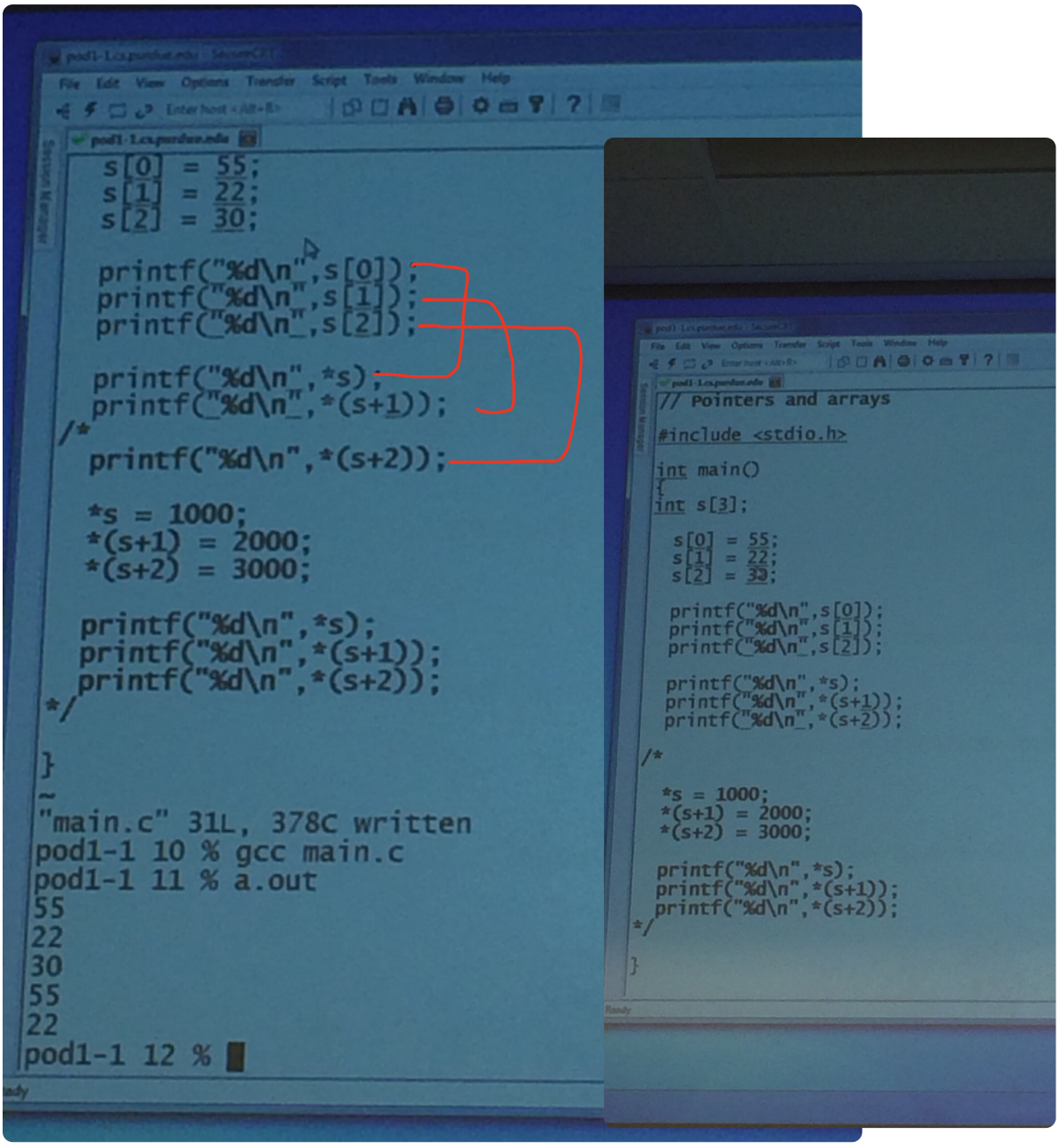
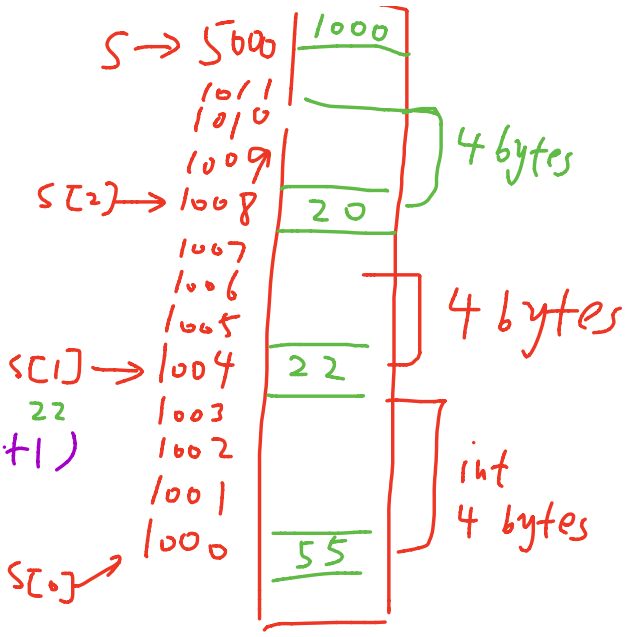

s[0] = 55;
s[1] = 22;
s[2] = 30;
printf("%d\n",s[0]);
printf("%d\n",s[1]);
printf("%d\n",s[2]);
printf("%d\n",*s);
printf("%d\n",*(s+1));
/*
printf("%d\n",*(s+2));
*s = 1000;
*(s+1) = 2000;
*(s+2) = 3000;
printf("%d\n",*s);
printf("%d\n",*(s+1));
printf("%d\n",*(s+2));
*/
}
"main.c" 31L, 378C written
pod1-1 10 % gcc main.c
pod1-1 11 % a.out
55
22
30
55
22
pod1-1 12 %
// 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));
*/
}

s[0] = 55

*s
↓
s is address
a variable



22

*(s+1)

↓

1000+4
=1004

sizeof(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 %
```

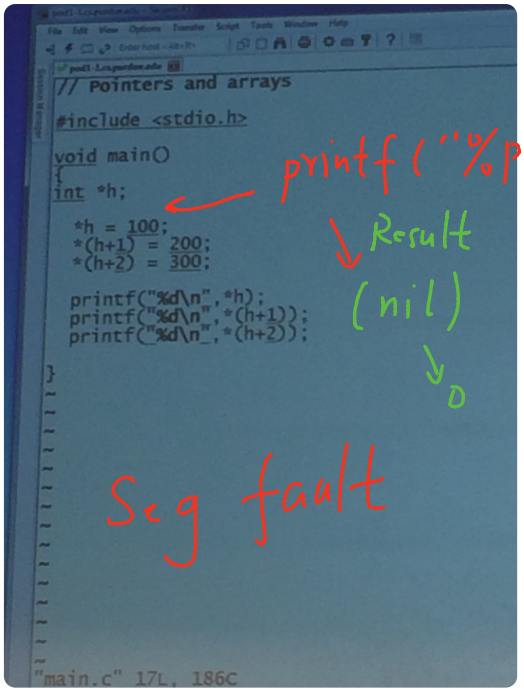
```
  printf("%d\n",s[0]);
  printf("%d\n",s[1]);
  printf("%d\n",s[2]);

  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 %
```



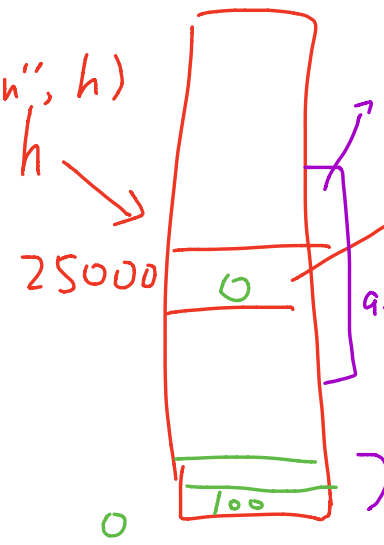
printf("%p\n", h)

Result

(nil)

0

Seg fault

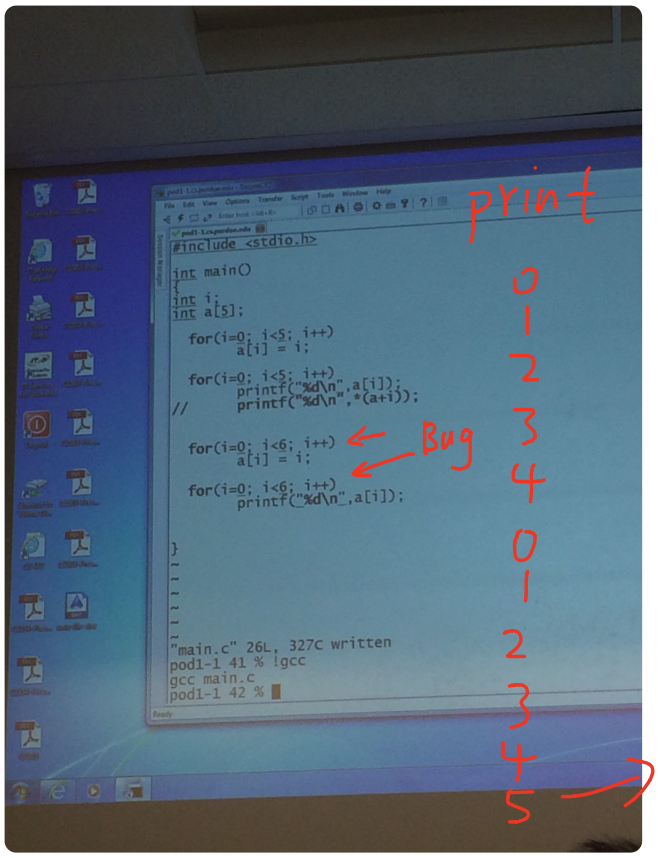


these section belongs to a.out

Random number can be any number

not belong to a.out

seg Fault



Bug

print

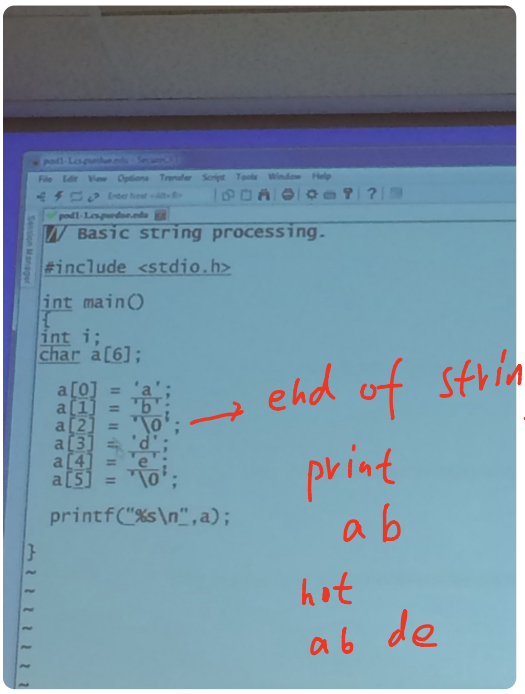
0
1
2
3
4
0
1
2
3
4
5

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

not crush

Bad luck


// 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
ab de