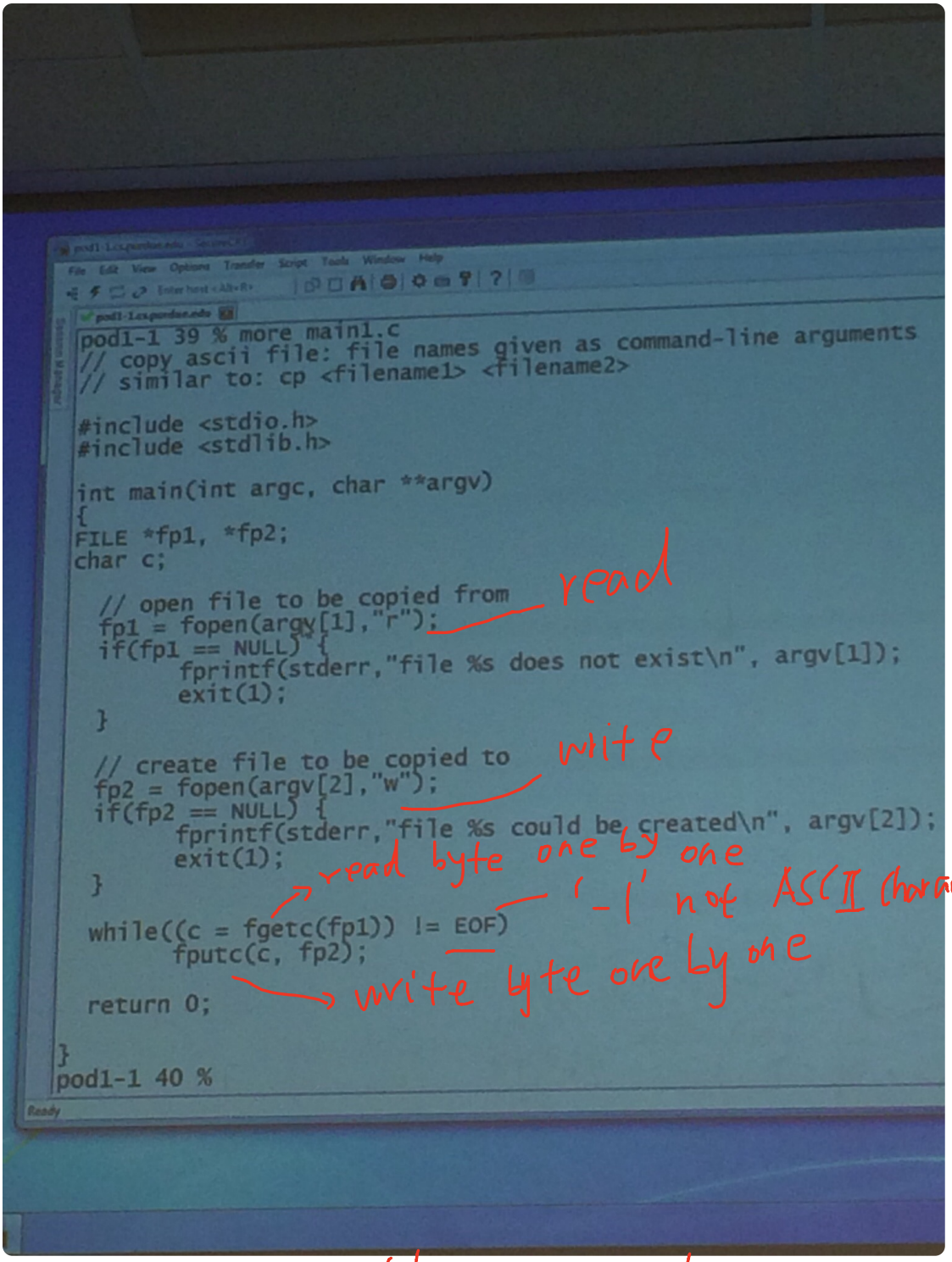


```
pod1-1 39 % more main1.c
// copy ascii file: file names given as command-line arguments
// similar to: cp <filename1> <filename2>

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
FILE *fp1, *fp2;
char c;

  // open file to be copied from
  fp1 = fopen(argv[1],"r");
  if(fp1 == NULL) {
        fprintf(stderr,"file %s does not exist\n", argv[1]);
        exit(1);
  }

  // create file to be copied to
  fp2 = fopen(argv[2],"w");
  if(fp2 == NULL) {
        fprintf(stderr,"file %s could be created\n", argv[2]);
        exit(1);
  }

  while((c = fgetc(fp1)) != EOF)
        fputc(c, fp2);

  return 0;

}
pod1-1 40 %
```

read

write

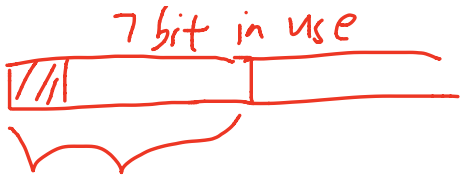
read byte one by one

'-1' not ASCII character

write byte one by one

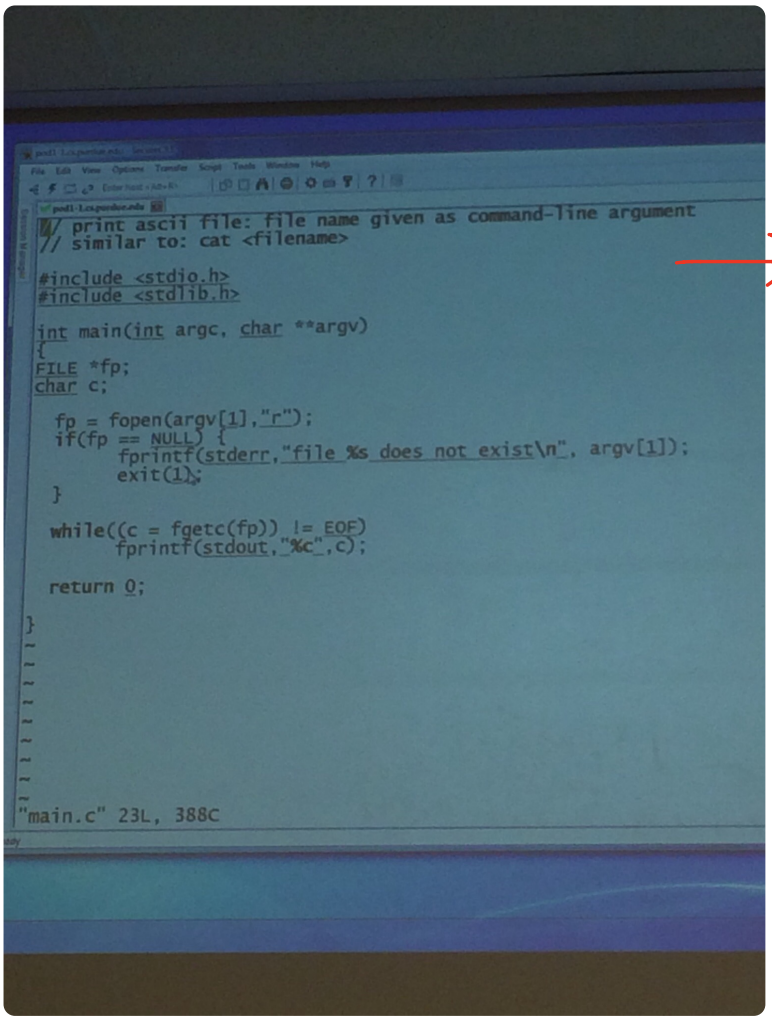
Can copy binary file, ASCII file;

ASCII file



binary file

use all 8 bits

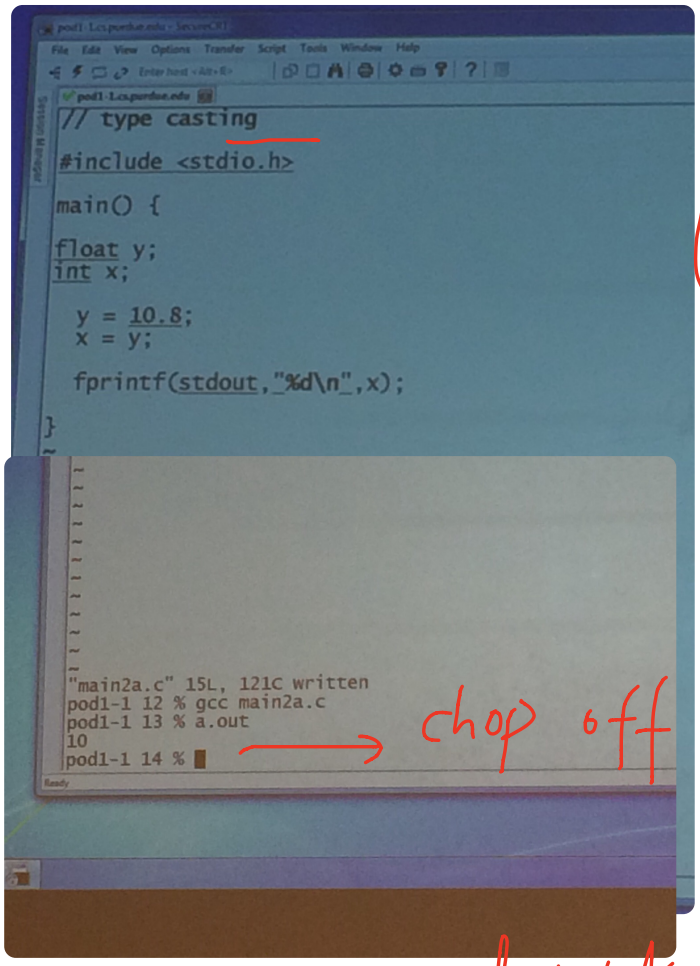


```
// print ascii file: file name given as command-line argument
// similar to: cat <filename>

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
FILE *fp;
char c;

  fp = fopen(argv[1],"r");
  if(fp == NULL) {
        fprintf(stderr,"file %s does not exist\n", argv[1]);
        exit(1);
  }

  while((c = fgetc(fp)) != EOF)
        fprintf(stdout,"%c",c);

  return 0;

}
```

"main.c" 23L, 388C

->mycat.c

Can do binary file too



explicit type casting

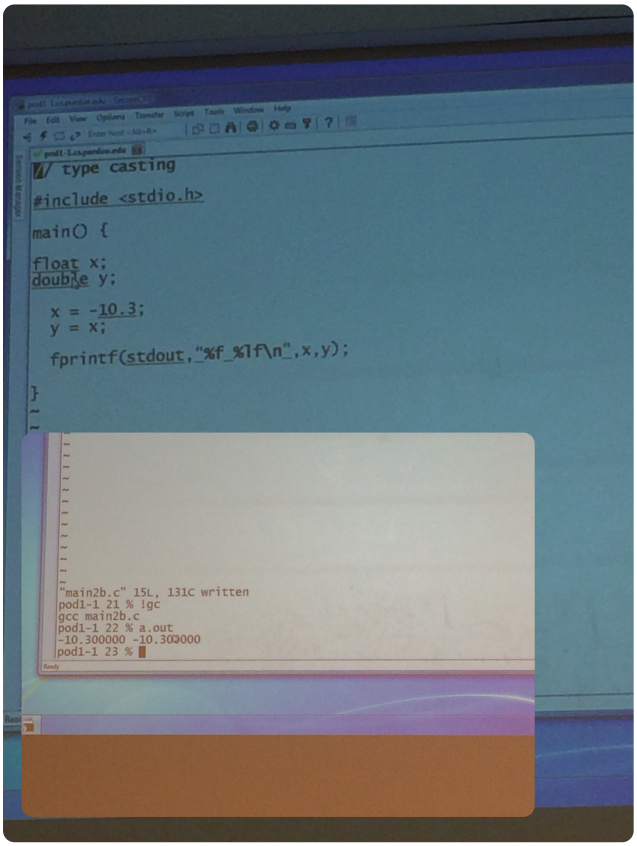
(int *) malloc( )

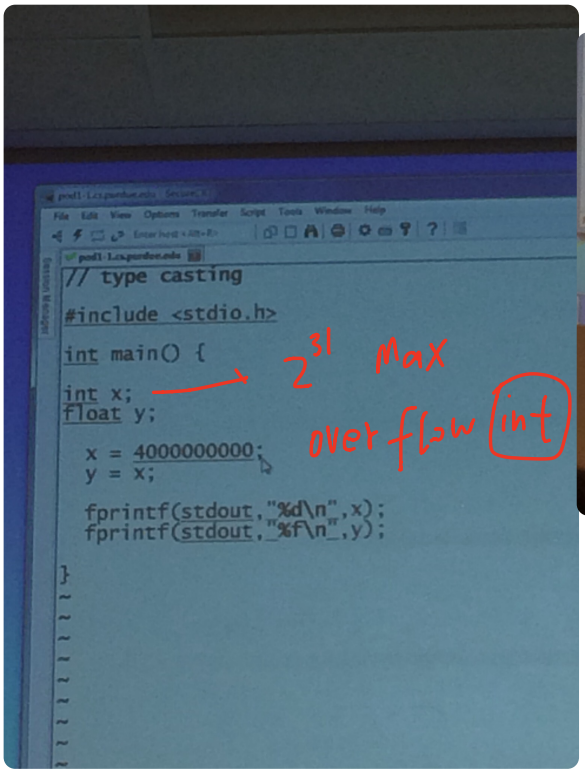
char *

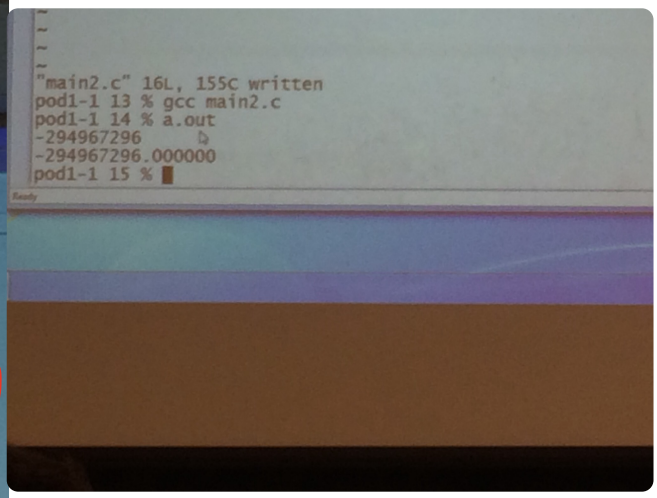
chop off decimal part

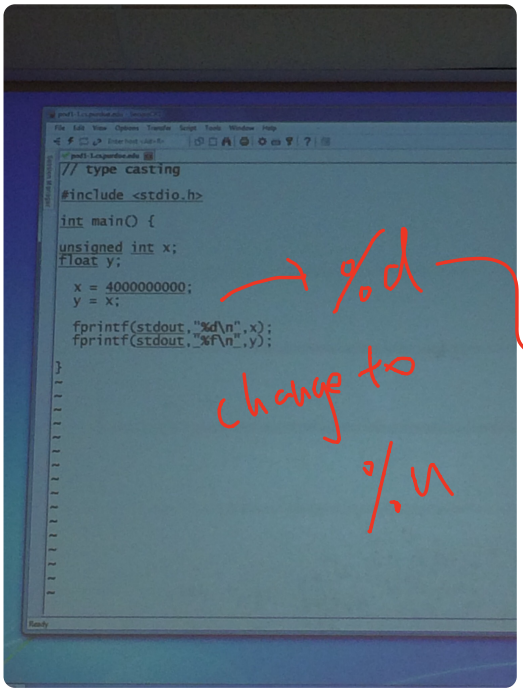
not round it up

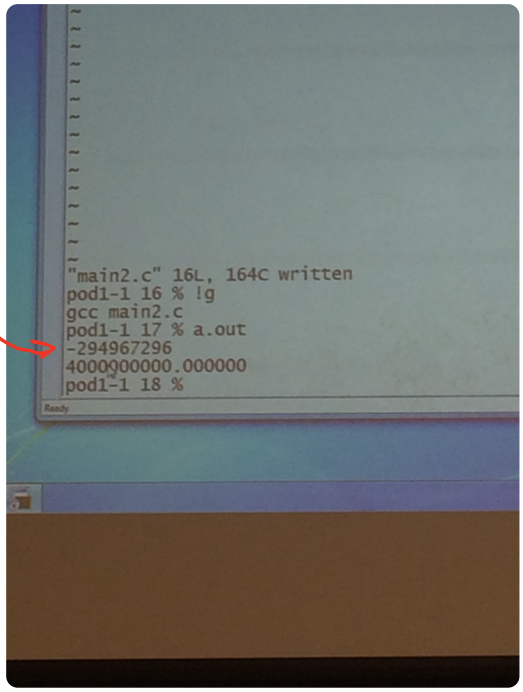
depends on compiler

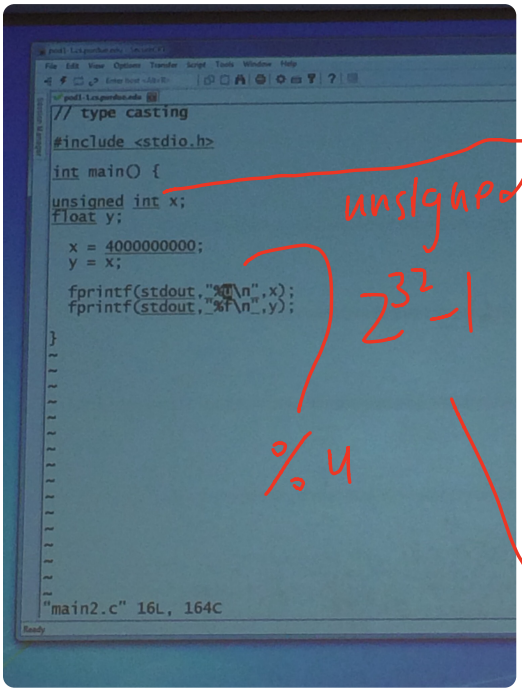

// type casting
#include <stdio.h>
main() {
float x;
double y;
x = -10.3;
y = x;
fprintf(stdout,"%f_%lf\n",x,y);
}
"main2b.c" 15L, 131C written
pod1-1 21 % !gc
gcc main2b.c
pod1-1 22 % a.out
-10.300000 -10.300000
pod1-1 23 %

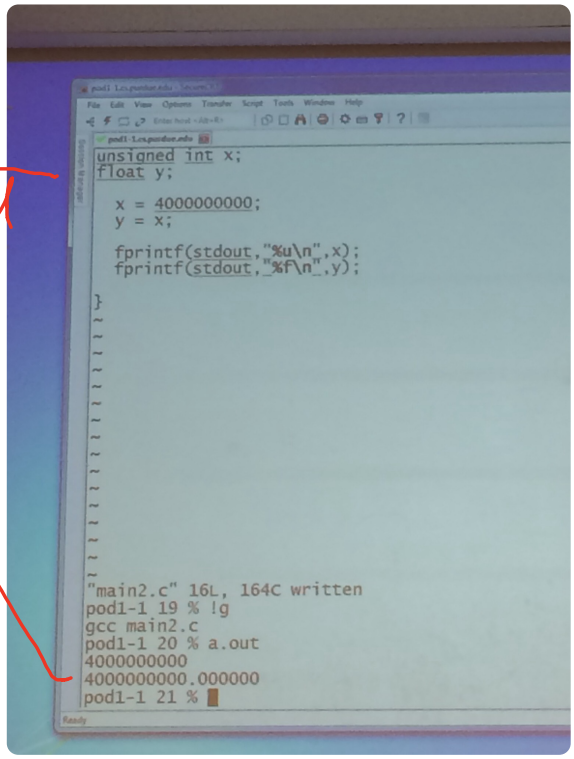

// type casting
#include <stdio.h>
int main() {
int x;
float y;
x = 4000000000;
y = x;
fprintf(stdout,"%d\n",x);
fprintf(stdout,"%f\n",y);
}
2^31
Max
over flow
int

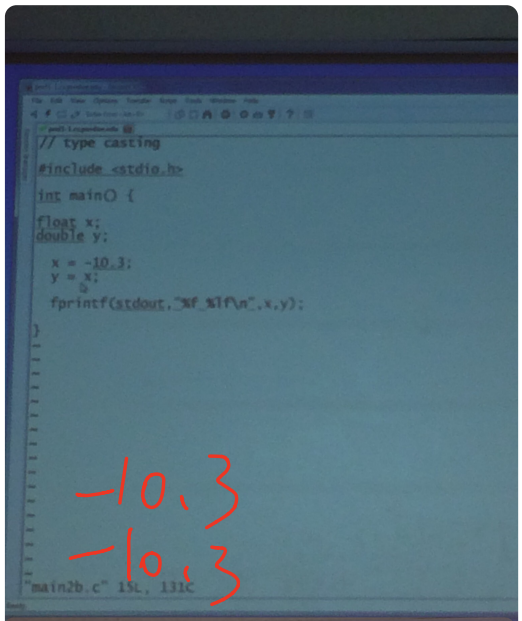

"main2.c" 16L, 155C written
pod1-1 13 % gcc main2.c
pod1-1 14 % a.out
-294967296
-294967296.000000
pod1-1 15 %

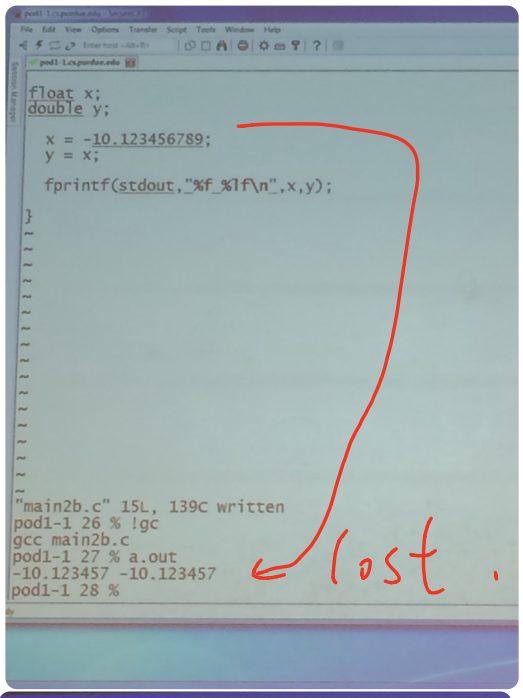

// type casting
#include <stdio.h>
int main() {
unsigned int x;
float y;
x = 4000000000;
y = x;
fprintf(stdout,"%d\n",x);
fprintf(stdout,"%f\n",y);
}
%d
change to
%u

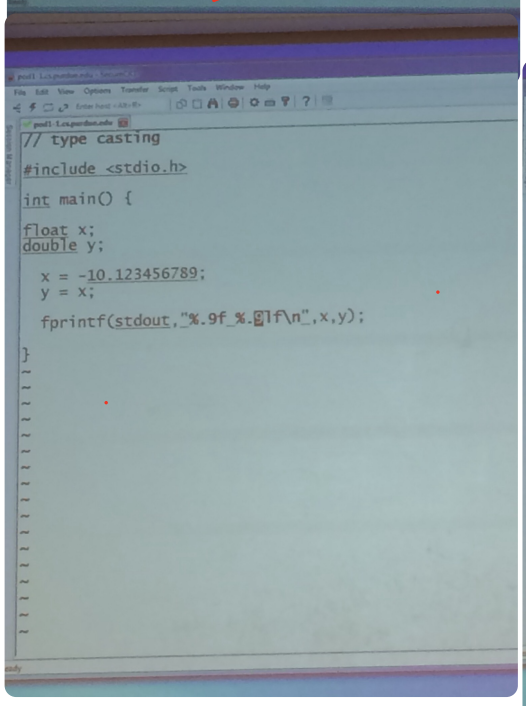

"main2.c" 16L, 164C written
pod1-1 16 % !g
gcc main2.c
pod1-1 17 % a.out
-294967296
4000000000.000000
pod1-1 18 %

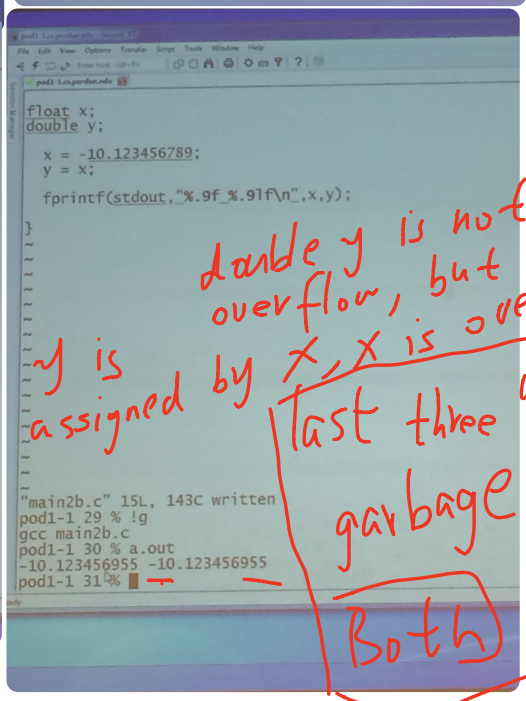

// type casting
#include <stdio.h>
int main() {
unsigned int x;
float y;
x = 4000000000;
y = x;
fprintf(stdout,"%u\n",x);
fprintf(stdout,"%f\n",y);
}
"main2.c" 16L, 164C
unsigned
$2^{32}-1$
%u

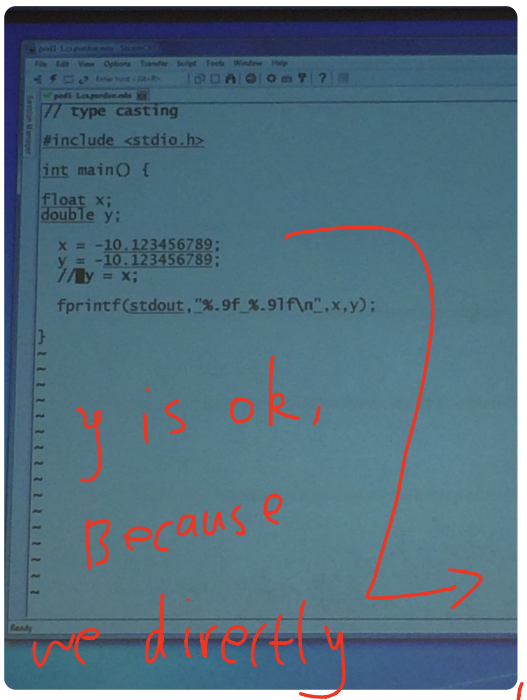

unsigned int x;
float y;
x = 4000000000;
y = x;
fprintf(stdout,"%u\n",x);
fprintf(stdout,"%f\n",y);
}
"main2.c" 16L, 164C written
pod1-1 19 % !g
gcc main2.c
pod1-1 20 % a.out
4000000000
4000000000.000000
pod1-1 21 %

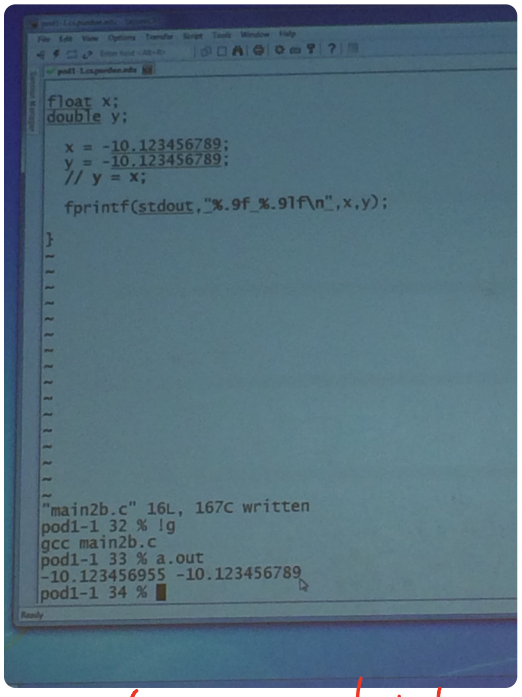

// type casting
#include <stdio.h>
int main() {
float x;
double y;
x = -10.3;
y = x;
fprintf(stdout,"%f %lf\n",x,y);
}
-10.3
-10.3
"main2b.c" 15L, 131C


float x;
double y;
x = -10.123456789;
y = x;
fprintf(stdout,"%f %lf\n",x,y);
}
"main2b.c" 15L, 139C written
pod1-1 26 % !gc
gcc main2b.c
pod1-1 27 % a.out
-10.123457 -10.123457
pod1-1 28 %
lost .89


// type casting
#include <stdio.h>
int main() {
float x;
double y;
x = -10.123456789;
y = x;
fprintf(stdout,"%.9f %.9lf\n",x,y);
}

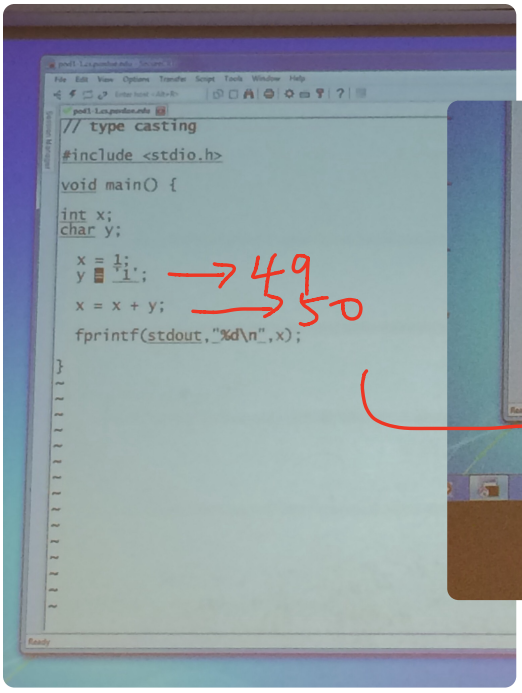

float x;
double y;
x = -10.123456789;
y = x;
fprintf(stdout,"%.9f %.9lf\n",x,y);
}
"main2b.c" 15L, 143C written
pod1-1 29 % !g
gcc main2b.c
pod1-1 30 % a.out
-10.123456955 -10.123456955
pod1-1 31 %
double y is not overflow, but y is assigned by x, x is overflowed
last three digits garbage
Both

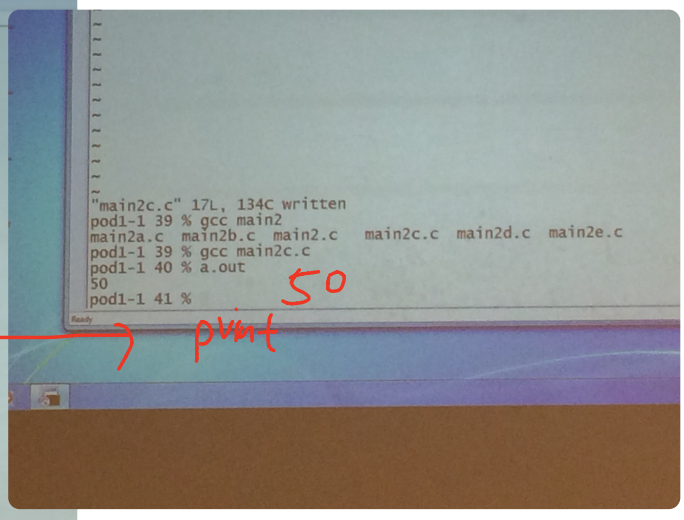


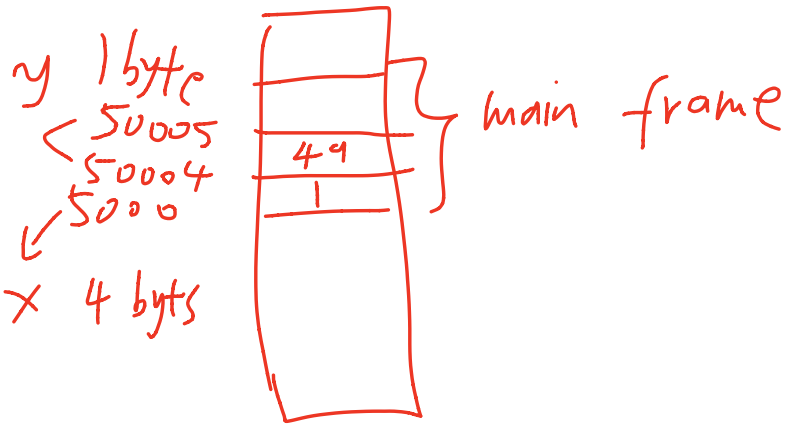


assign this big value to y, which is not overflow.

x is still overflow.



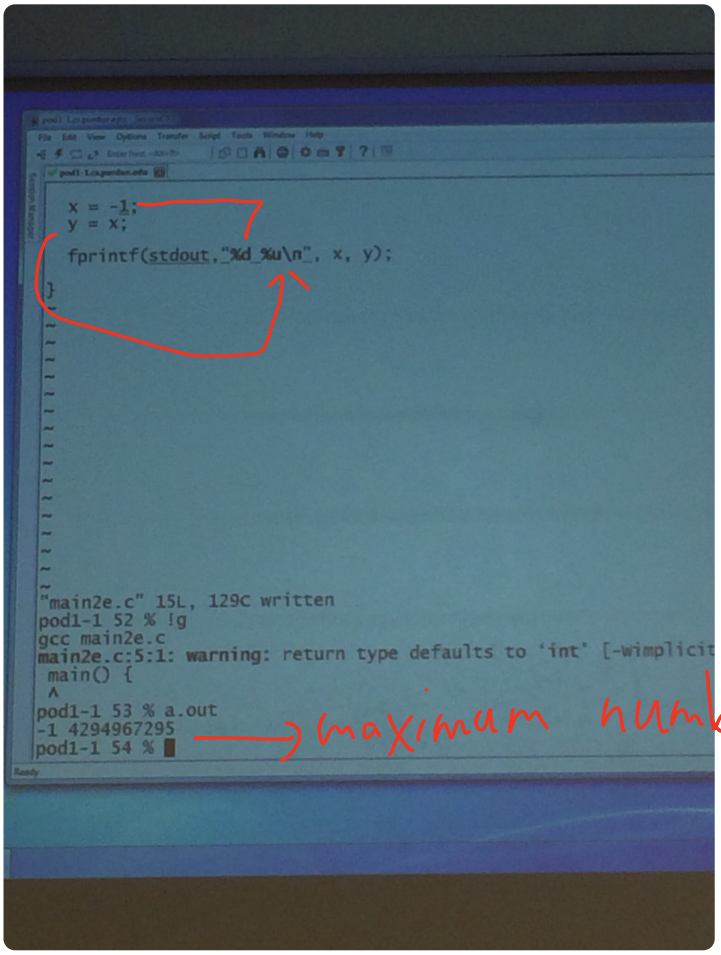




after x = x + y;

y promote from char to int
1 byte 4 byts

Compiler generate a new plare which is 4 bytes, subsitute the old address of y.


x = -1;
y = x;
fprintf(stdout,"%d %u\n", x, y);
}
"main2e.c" 15L, 129C written
pod1-1 52 % !g
gcc main2e.c
main2e.c:5:1: warning: return type defaults to 'int' [-Wimplicit
main() {
^
pod1-1 53 % a.out
-1 4294967295
pod1-1 54 %
maximum number