C Programming
Day10.B

2017.10.18

Arrays, Sort, Random Numbers, Bit Shift

```c
#include <stdio.h>

// printf("a[%d]=%d a[%d]=%d \n", j, a[j], j+1, a[j+1]);

int main(void) {
  int a[] = { 3, 5, 2, 4, 1, 6};
  int i, j, k, hold;

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

  for (i=0; i<6; ++i) {
    for (j=0; j<5; ++j) {
       if ( a[j] > a[j+1]) {
          hold = a[j];
          a[j] = a[j+1];
          a[j+1] = hold;
       }
       for (k=0; k<6; ++k) printf("%d ", a[k]); puts("");
    }
    puts("--------------------");
  }


}
```

```
3 5 2 4 1 6
----------------------
3 5 2 4 1 6
3 2 5 4 1 6
3 2 4 5 1 6
3 2 4 1 5 6
3 2 4 1 5 6
----------------------
2 3 4 1 5 6
2 3 4 1 5 6
2 3 1 4 5 6
2 3 1 4 5 6
2 3 1 4 5 6
----------------------
2 3 1 4 5 6
2 1 3 4 5 6
2 1 3 4 5 6
2 1 3 4 5 6
2 1 3 4 5 6
----------------------
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
----------------------
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
----------------------
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
----------------------
```

```c
#include <stdio.h>
#define SIZE 6

void bubble_sort( int a[] ) {
  int i, j, hold;

  for (i=0; i<SIZE; ++i) {
    for (j=0; j<SIZE-1; ++j) {
       if ( a[j] > a[j+1]) {
          hold = a[j];
          a[j] = a[j+1];
          a[j+1] = hold;
       }
    }
  }

}

int main(void) {
  int a[] = { 3, 5, 2, 4, 1, 6};
  int i;

  for (i=0; i<SIZE; ++i) printf("%d ", a[i]); puts("");

  bubble_sort( a );

  for (i=0; i<SIZE; ++i) printf("%d ", a[i]); puts("");


}
```

```c
#include <stdio.h>
#define SIZE 6

// for 1-d array
// int *x
// int x[]
// in the parameter list of a funciton prototype

void func( int *x ) {
  int i;

  printf("func: x = %p\n", x);

  for (i=0; i<SIZE; ++i)
    printf("func: x+%d = %p \n", i, x+i);

  for (i=0; i<SIZE; ++i)
    printf("func: *(x+%d) = %d \n", i, *(x+i));

  for (i=0; i<SIZE; ++i)
    printf("func: x[%d] = %d \n", i, x[i]);

  for (i=0; i<SIZE; ++i)
    printf("func: x[%d] = %d \n", i, x[i]=0);
}


int main(void) {
  int a[] = { 3, 5, 2, 4, 1, 6};
  int i;

  for (i=0; i<SIZE; ++i)
    printf("main: a+%d = %p \n", i, a+i);

  func( a );

  for (i=0; i<SIZE; ++i)
    printf("main: *(a+%d) = %d \n", i, *(a+i));


}
```

```
main: a+0 = 0x7ffd952d0850
main: a+1 = 0x7ffd952d0854
main: a+2 = 0x7ffd952d0858
main: a+3 = 0x7ffd952d085c
main: a+4 = 0x7ffd952d0860
main: a+5 = 0x7ffd952d0864
func: x = 0x7ffd952d0850
func: x+0 = 0x7ffd952d0850
func: x+1 = 0x7ffd952d0854
func: x+2 = 0x7ffd952d0858
func: x+3 = 0x7ffd952d085c
func: x+4 = 0x7ffd952d0860
func: x+5 = 0x7ffd952d0864
func: *(x+0) = 3
func: *(x+1) = 5
func: *(x+2) = 2
func: *(x+3) = 4
func: *(x+4) = 1
func: *(x+5) = 6
func: x[0] = 3
func: x[1] = 5
func: x[2] = 2
func: x[3] = 4
func: x[4] = 1
func: x[5] = 6
func: x[0] = 0
func: x[1] = 0
func: x[2] = 0
func: x[3] = 0
func: x[4] = 0
func: x[5] = 0
main: *(a+0) = 0
main: *(a+1) = 0
main: *(a+2) = 0
main: *(a+3) = 0
main: *(a+4) = 0
main: *(a+5) = 0
```

changed by func()

seen by main()

```c
#include <stdio.h>

// int a[] = {1, 2, 3, 4};

int main(void) {
  int a[5] = {1, 2, 3, 4};
  int b[5] = {0};
  int i;

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

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

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

  for (i=0; i<5; ++i)
    printf("b[%d]= %d \n", i, b[i]);
}
```

```
a[0]= 1
a[1]= 2
a[2]= 3
a[3]= 4
a[4]= 0
a[0]= 2
a[1]= 3
a[2]= 4
a[3]= 5
a[4]= 1
b[0]= 1
b[1]= 2
b[2]= 3
b[3]= 4
b[4]= 0
```

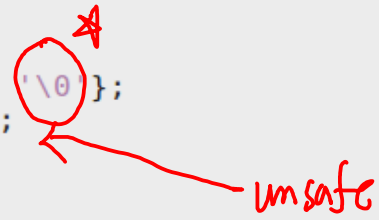
initializer = {1, 2, 3, 4}

= {1, 2, 3, 4, 0}

a[i]++ ..... (a[i])++

++a[i] ..... ++(a[i])

```c
#include <stdio.h>

int main(void) {
  char a[] = {'h', 'e', 'l', 'l', 'o', '\0'};
  char b[] = {'h', 'e', 'l', 'l', 'o'};
  char c[] = "hello";
  char d[10];
  int i;


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

  for (i=0; i<6; ++i)
    printf("b[%d]= %c %d\n", i, b[i], b[i]);
  puts("--------------------");

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

  for (i=0; i<10; ++i)
    printf("d[%d]= %c %d\n", i, d[i], d[i]);
  puts("--------------------");


  printf("a= %p \n", a);
  printf("b= %p \n", b);
  printf("c= %p \n", c);
  printf("c= %p \n", c);

  for (i=0; i<10; ++i)  d[i] = 'A';
  d[9] = 0;
  for (i=0; i<5; ++i)  d[i] = b[i];

  for (i=0; i<10; ++i)
    printf("d[%d]= %c %d\n", i, d[i], d[i]);
  puts("--------------------");


  printf("a= %s \n", a);
  printf("b= %s \n", b);
  printf("c= %s \n", c);
  printf("d= %s \n", d);
}
```



```
a[0]= h 104
a[1]= e 101
a[2]= l 108
a[3]= l 108
a[4]= o 111
a[5]=  0
--------------------
b[0]= h 104
b[1]= e 101
b[2]= l 108
b[3]= l 108
b[4]= o 111
b[5]=  0
--------------------
c[0]= h 104
c[1]= e 101
c[2]= l 108
c[3]= l 108
c[4]= o 111
c[5]=  0
--------------------
d[0]= ? -80
d[1]= 8
d[2]= @ 64
d[3]=  0
d[4]=  0
d[5]=  0
d[6]=  0
d[7]=  0
d[8]= ? -32
d[9]= ? 4
```

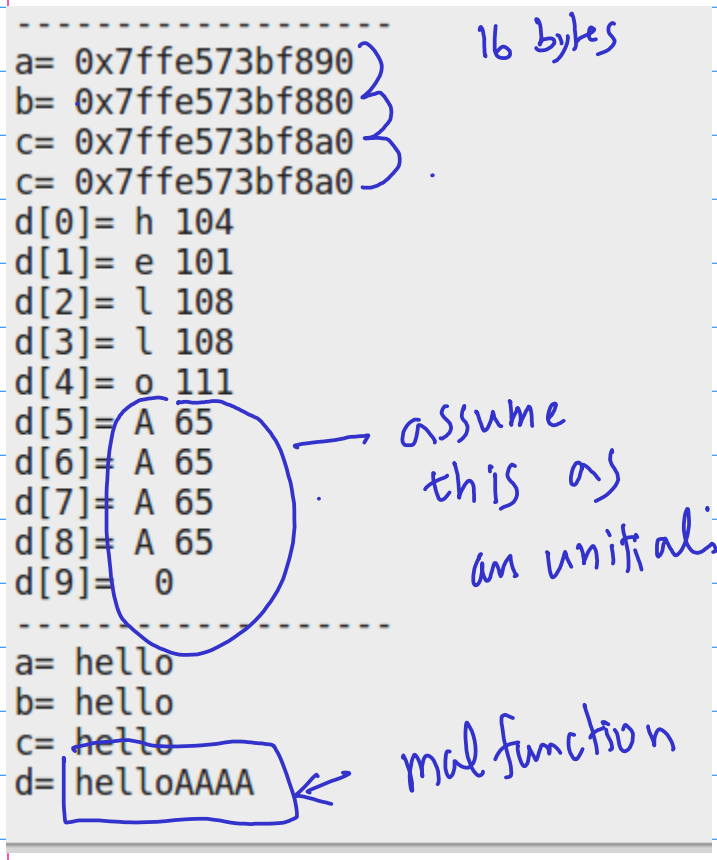
'0'

uninitialized data (random zero)

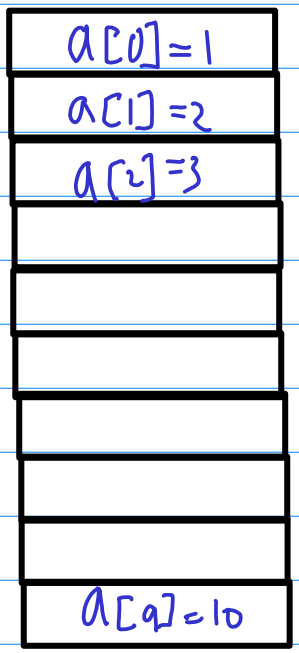
happen to be zero

uninitialized data

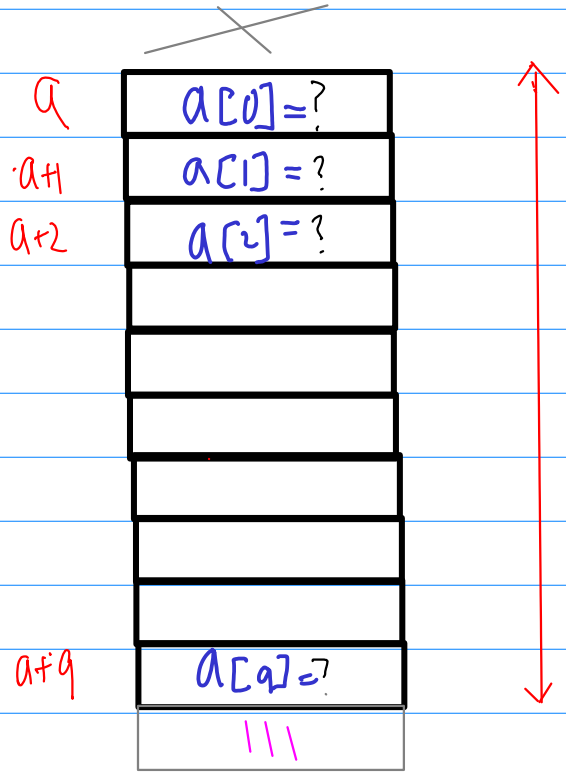
some of them are zero!




```
--------------------
a= 0x7ffe573bf890
b= 0x7ffe573bf880
c= 0x7ffe573bf8a0
c= 0x7ffe573bf8a0
d[0]= h 104
d[1]= e 101
d[2]= l 108
d[3]= l 108
d[4]= o 111
d[5]= A 65
d[6]= A 65
d[7]= A 65
d[8]= A 65
d[9]=  0
--------------------
a= hello
b= hello
c= hello
d= helloAAAA
```


a[0]=1
a[1]=2
a[2]=3
a[9]=10

```
int a[10];
```

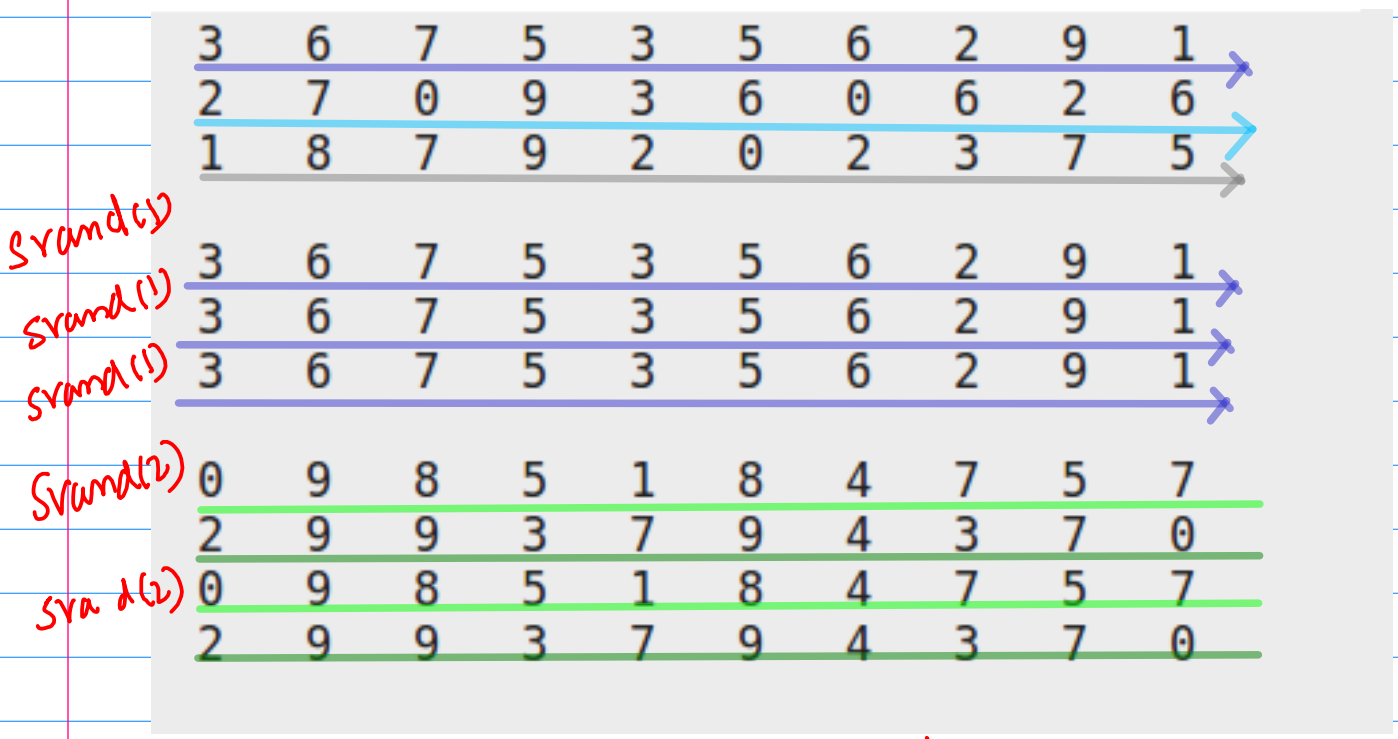



```
a[10] = ???;
```

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

void func(void) {
  int i, m;

  for (i=0; i<10; ++i) {
    m = rand() % 10;
    printf("%3d ", m);
  }
  printf("\n");
}


int main(void) {
  func();
  func();
  func();
  puts(" ");

  srand(1);
  func();
  srand(1);
  func();
  srand(1);
  func();
  puts(" ");

  srand(2);
  func();
  func();
  srand(2);
  func();
  func();


}
```

← srand(1); by default

srand(1)



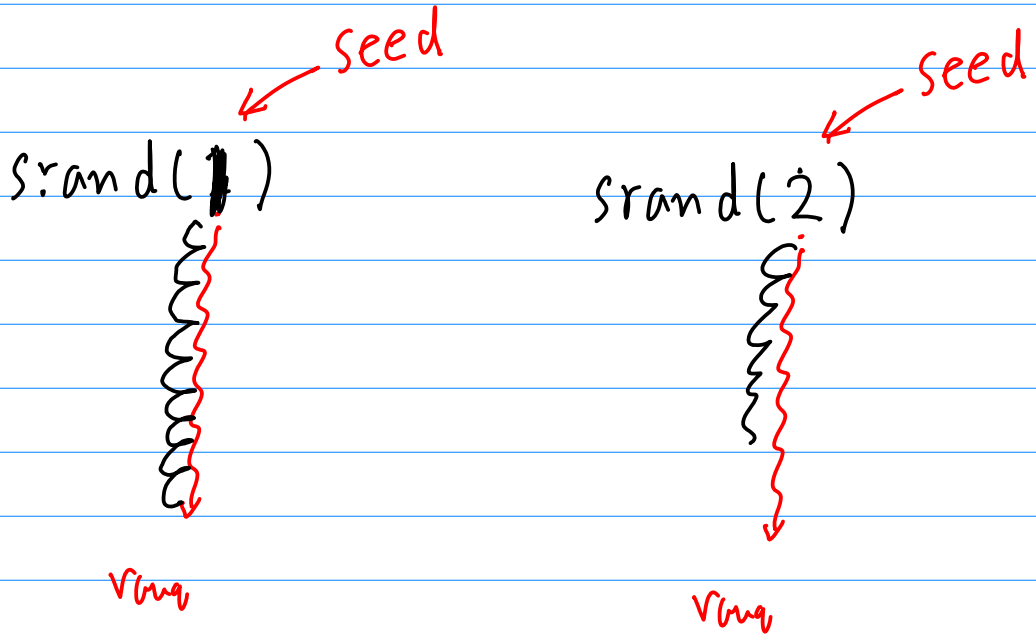
Srand (1)

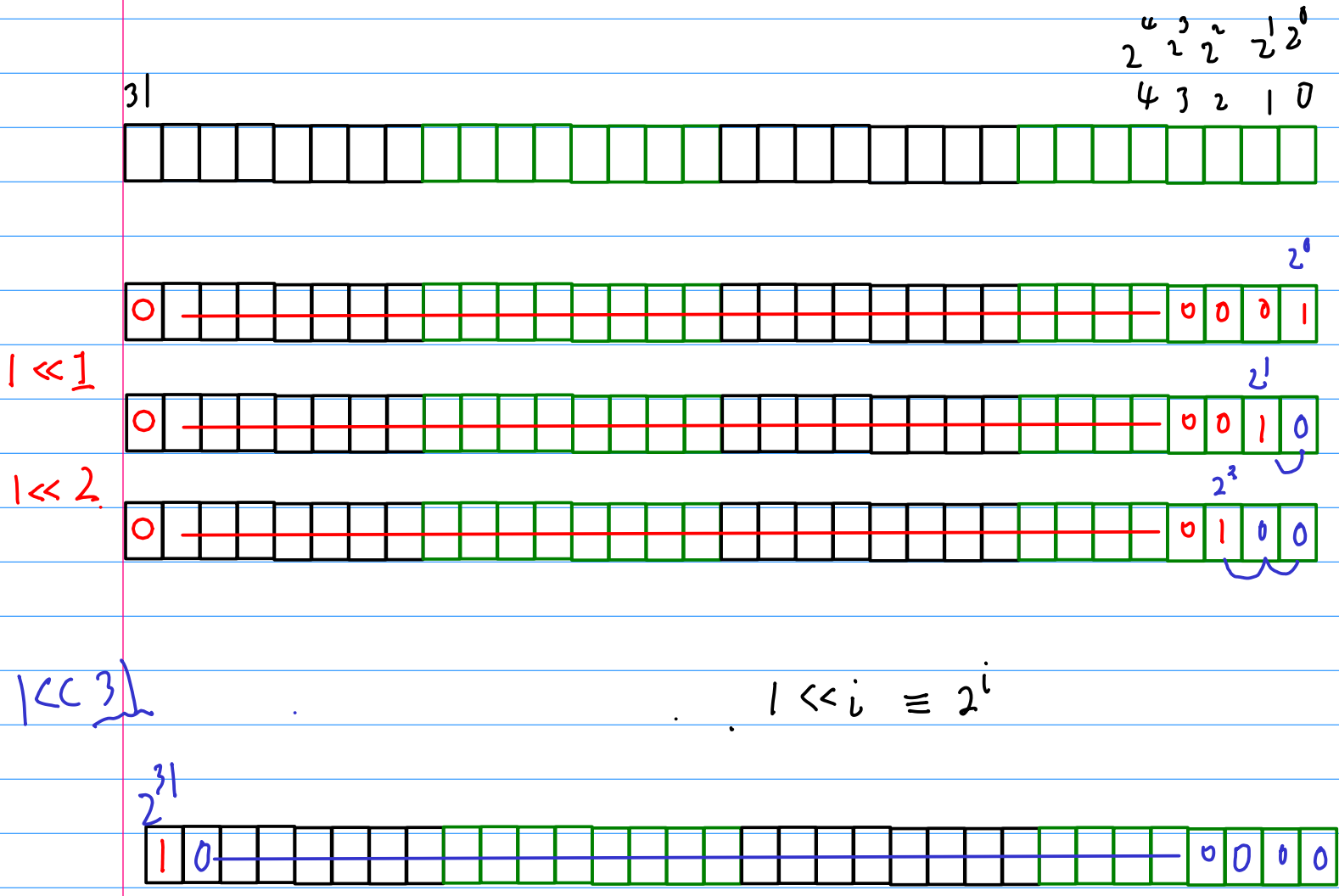
3 6 7 5 3 5 6 2 9 1 2 7 0 9 3 6 0 6 2 6 1 8 7 9 2 0 2 3 7 5

srand(2)

0 9 8 5 1 8 4 7 5 7 2 9 9 3 7 9 4 3 7 0

...


seed
srand(1)
rand
seed
srand(2)
rand



1 << 3

$1 << i \equiv 2^i$

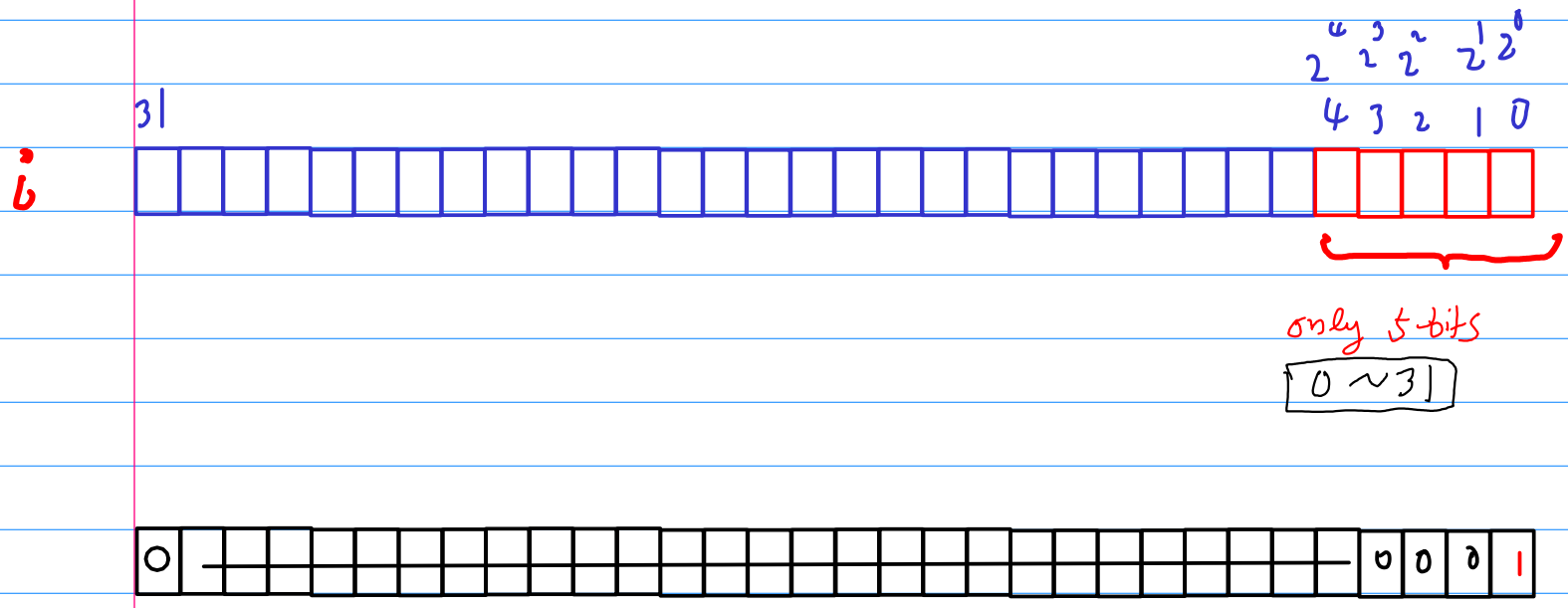
1 << 32

1<<31

$1 << i \equiv 2^i$

- 2147483648

1 << i



$$1 << 1$$
$$= 1 << 33$$
$$= 1 << 65$$

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

int main(void) {
  int i;
  int m;

  for (i=0; i<256; ++i) {
    m = (1 << i);
    printf("2^%2d    = %16d\n", i, m);
  }

  printf("\nRAND_MAX= %16d\n", RAND_MAX);
}
```

2^ 0	=	1	2^32	=	1	2^64	=	1
2^ 1	=	2	2^33	=	2	2^65	=	2
2^ 2	=	4	2^34	=	4	2^66	=	4
2^ 3	=	8	2^35	=	8	2^67	=	8
2^ 4	=	16	2^36	=	16	2^68	=	16
2^ 5	=	32	2^37	=	32	2^69	=	32
2^ 6	=	64	2^38	=	64	2^70	=	64
2^ 7	=	128	2^39	=	128	2^71	=	128
2^ 8	=	256	2^40	=	256	2^72	=	256
2^ 9	=	512	2^41	=	512	2^73	=	512
2^10	=	1024	2^42	=	1024	2^74	=	1024
2^11	=	2048	2^43	=	2048	2^75	=	2048
2^12	=	4096	2^44	=	4096	2^76	=	4096
2^13	=	8192	2^45	=	8192	2^77	=	8192
2^14	=	16384	2^46	=	16384	2^78	=	16384
2^15	=	32768	2^47	=	32768	2^79	=	32768
2^16	=	65536	2^48	=	65536	2^80	=	65536
2^17	=	131072	2^49	=	131072	2^81	=	131072
2^18	=	262144	2^50	=	262144	2^82	=	262144
2^19	=	524288	2^51	=	524288	2^83	=	524288
2^20	=	1048576	2^52	=	1048576	2^84	=	1048576
2^21	=	2097152	2^53	=	2097152	2^85	=	2097152
2^22	=	4194304	2^54	=	4194304	2^86	=	4194304
2^23	=	8388608	2^55	=	8388608	2^87	=	8388608
2^24	=	16777216	2^56	=	16777216	2^88	=	16777216
2^25	=	33554432	2^57	=	33554432	2^89	=	33554432
2^26	=	67108864	2^58	=	67108864	2^90	=	67108864
2^27	=	134217728	2^59	=	134217728	2^91	=	134217728
2^28	=	268435456	2^60	=	268435456	2^92	=	268435456
2^29	=	536870912	2^61	=	536870912	2^93	=	536870912
2^30	=	1073741824	2^62	=	1073741824	2^94	=	1073741824
2^31	=	-2147483648	2^63	=	-2147483648	2^95	=	-2147483648

2^64	=	1	2^128	=	1	2^128	=	1
2^65	=	2	2^129	=	2	2^129	=	2
2^66	=	4	2^130	=	4	2^130	=	4
2^67	=	8	2^131	=	8	2^131	=	8
2^68	=	16	2^132	=	16	2^132	=	16
2^69	=	32	2^133	=	32	2^133	=	32
2^70	=	64	2^134	=	64	2^134	=	64
2^71	=	128	2^135	=	128	2^135	=	128
2^72	=	256	2^136	=	256	2^136	=	256
2^73	=	512	2^137	=	512	2^137	=	512
2^74	=	1024	2^138	=	1024	2^138	=	1024
2^75	=	2048	2^139	=	2048	2^139	=	2048
2^76	=	4096	2^140	=	4096	2^140	=	4096
2^77	=	8192	2^141	=	8192	2^141	=	8192
2^78	=	16384	2^142	=	16384	2^142	=	16384
2^79	=	32768	2^143	=	32768	2^143	=	32768
2^80	=	65536	2^144	=	65536	2^144	=	65536
2^81	=	131072	2^145	=	131072	2^145	=	131072
2^82	=	262144	2^146	=	262144	2^146	=	262144
2^83	=	524288	2^147	=	524288	2^147	=	524288
2^84	=	1048576	2^148	=	1048576	2^148	=	1048576
2^85	=	2097152	2^149	=	2097152	2^149	=	2097152
2^86	=	4194304	2^150	=	4194304	2^150	=	4194304
2^87	=	8388608	2^151	=	8388608	2^151	=	8388608
2^88	=	16777216	2^152	=	16777216	2^152	=	16777216
2^89	=	33554432	2^153	=	33554432	2^153	=	33554432
2^90	=	67108864	2^154	=	67108864	2^154	=	67108864
2^91	=	134217728	2^155	=	134217728	2^155	=	134217728
2^92	=	268435456	2^156	=	268435456	2^156	=	268435456
2^93	=	536870912	2^157	=	536870912	2^157	=	536870912
2^94	=	1073741824	2^158	=	1073741824	2^158	=	1073741824
2^95	=	-2147483648	2^159	=	-2147483648	2^159	=	-2147483648