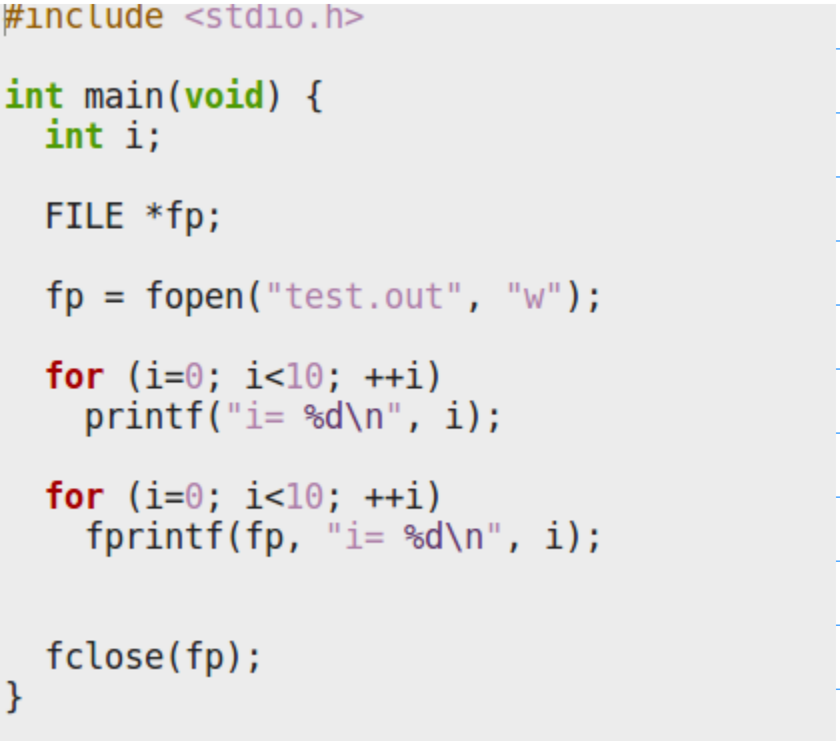
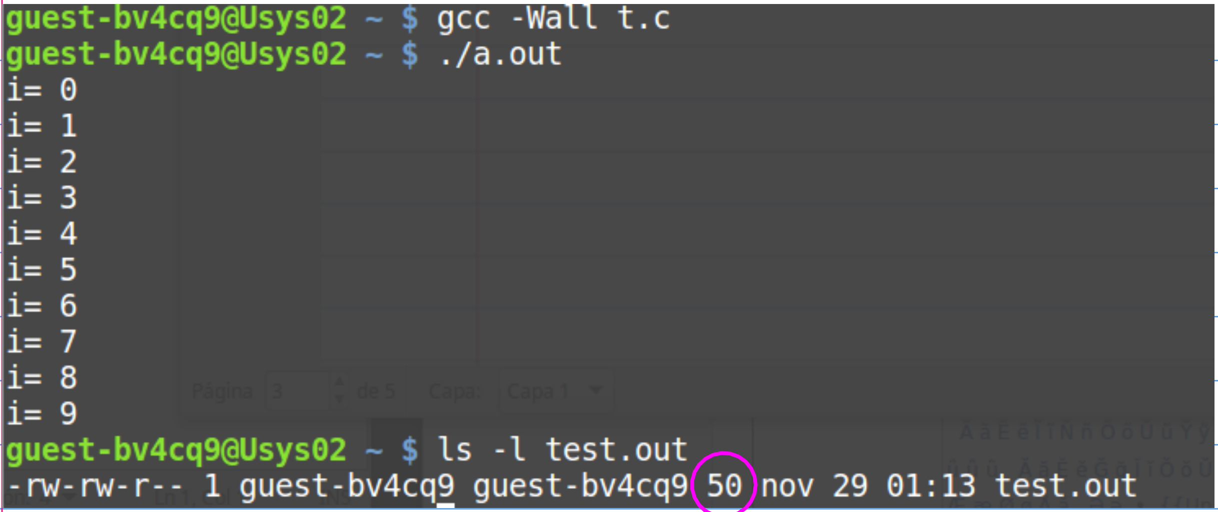
C Programming
Day21.B

2017.12.08

File Processing

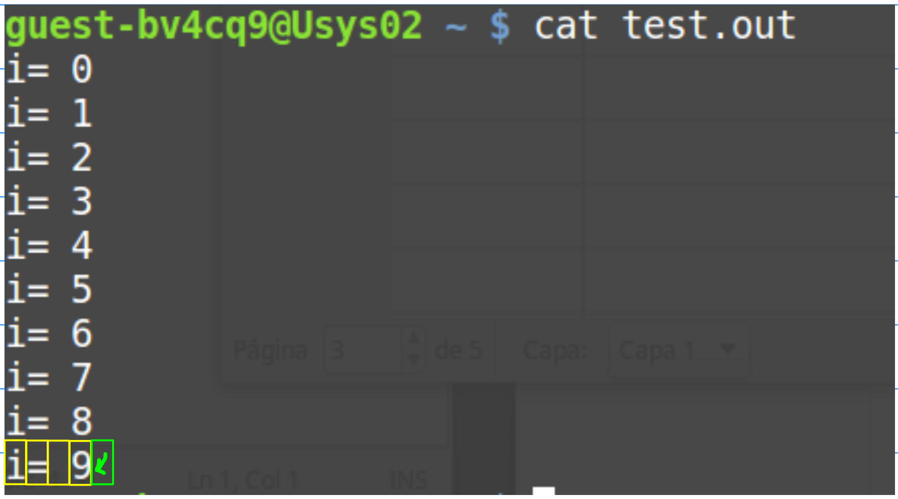


```
#include <stdio.h>

int main(void) {
  int i;

  FILE *fp;

  fp = fopen("test.out", "w");

  for (i=0; i<10; ++i)
    printf("i= %d\n", i);

  for (i=0; i<10; ++i)
    fprintf(fp, "i= %d\n", i);


  fclose(fp);
}
```



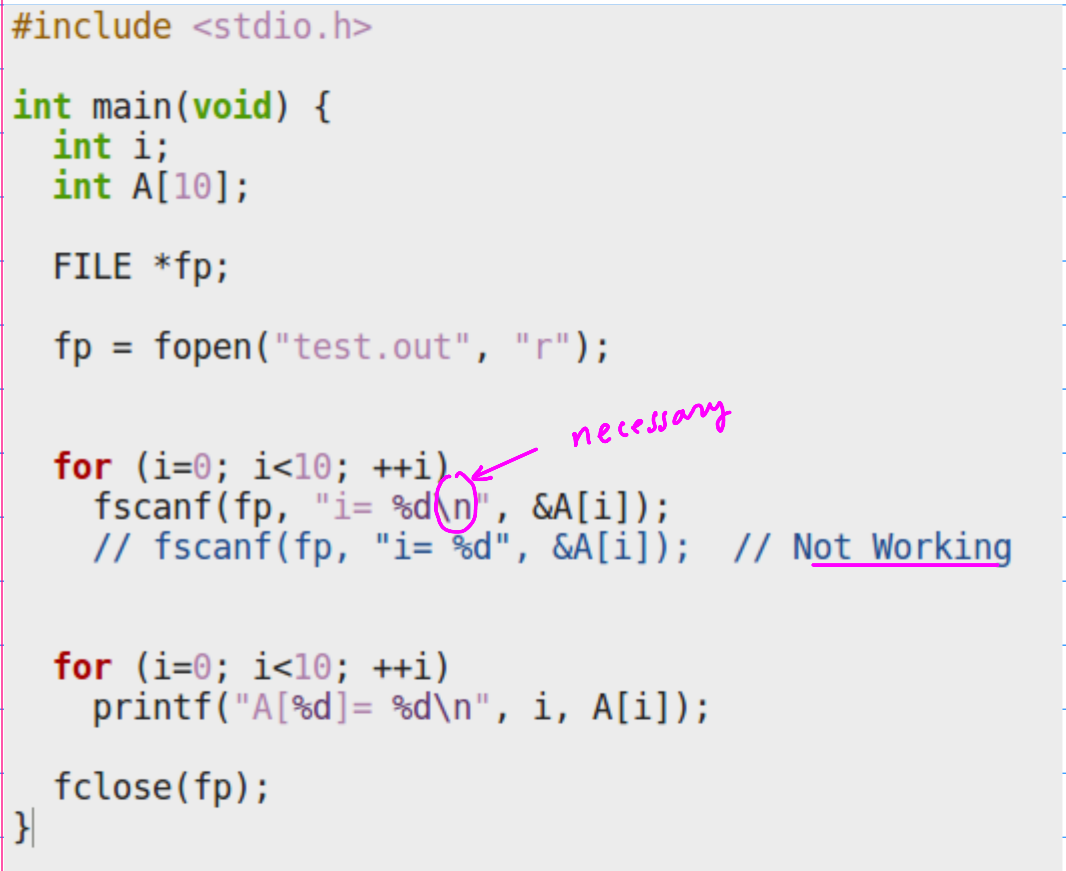
```
guest-bv4cq9@Usys02 ~ $ gcc -Wall t.c
guest-bv4cq9@Usys02 ~ $ ./a.out
i= 0
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9
guest-bv4cq9@Usys02 ~ $ ls -l test.out
-rw-rw-r-- 1 guest-bv4cq9 guest-bv4cq9 50 nov 29 01:13 test.out
```

50 bytes



```
guest-bv4cq9@Usys02 ~ $ cat test.out
i= 0
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9
```

5 bytes x 10 lines = 50 bytes

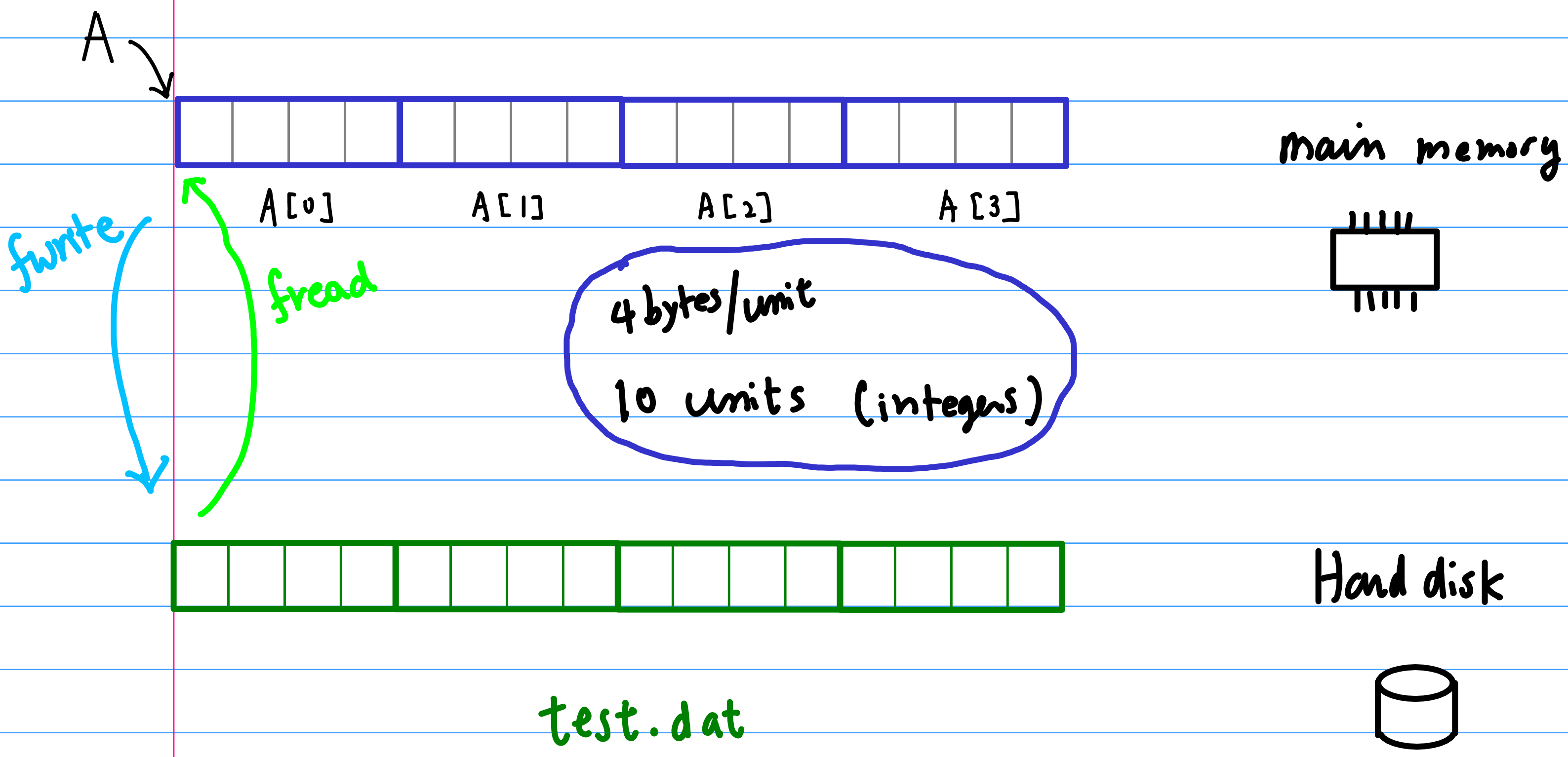


```
#include <stdio.h>

int main(void) {
  int i;
  int A[10];

  FILE *fp;

  fp = fopen("test.out", "r");


  for (i=0; i<10; ++i)
    fscanf(fp, "i= %d\n", &A[i]);
    // fscanf(fp, "i= %d", &A[i]);  // Not Working


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

  fclose(fp);
}
```

necessary

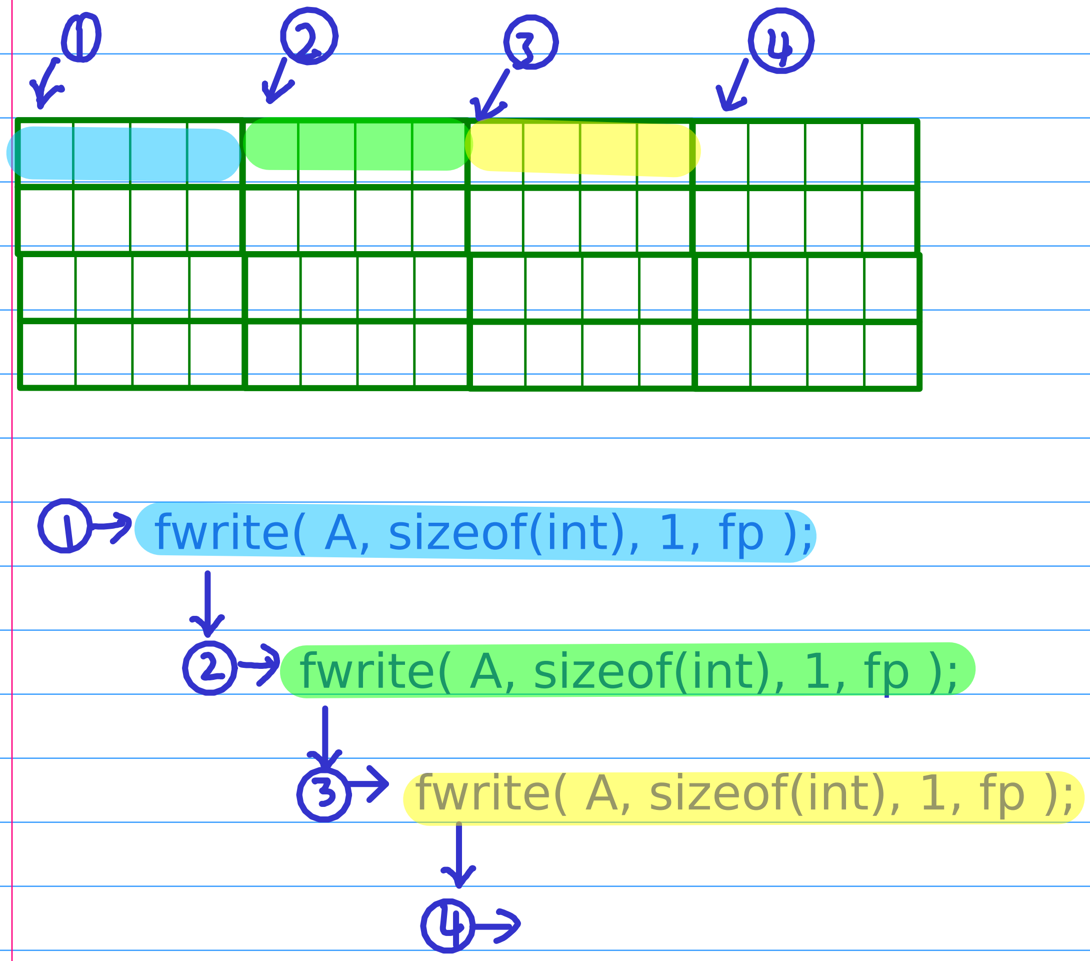
```
  FILE * fp;

                  fp = fopen( "test.dat", "w" ) ;


printf( "%d\n", a );
fprintf( fp, "%d\n", a );


                  fp = fopen( "test.dat", "r" ) ;


scanf( "%d", &b )
fscanf( fp, "%d", &b )
```

```
int A[10];
```



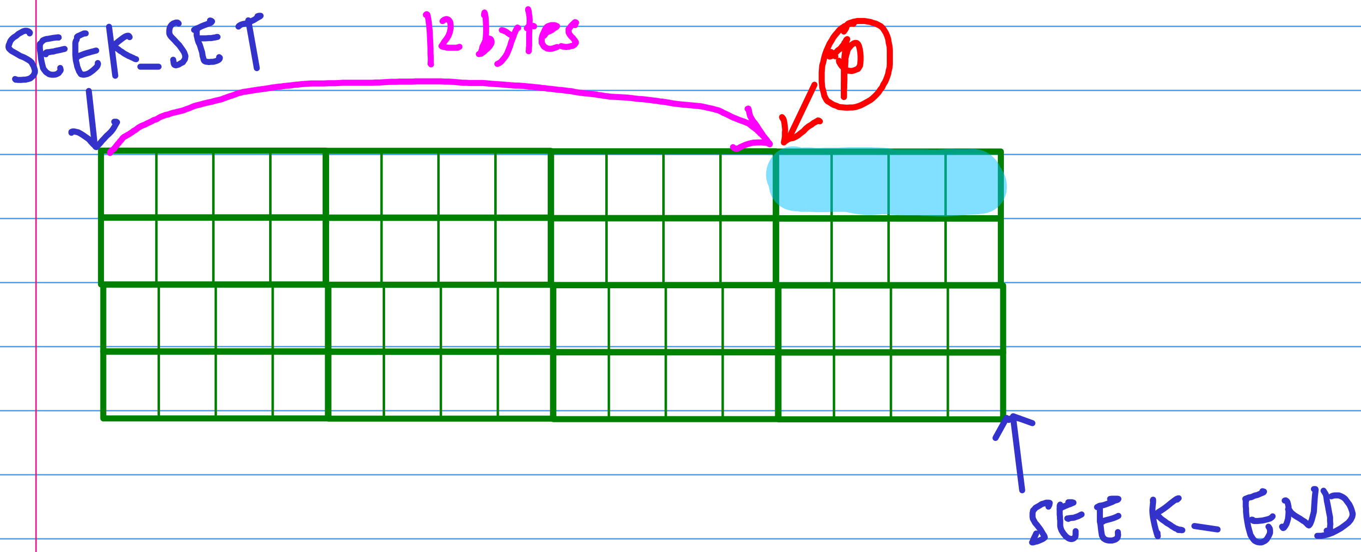
```
FILE * fp;

                    fp = fopen( "test.dat", "w") ;

fwrite( A, sizeof(int), 10, fp );


                    fp = fopen( "test.dat", "r") ;

fread( A, sizeof(int), 10, fp );
```

File Position Pointer



```
1 → fwrite( A, sizeof(int), 1, fp );
2 → fwrite( A, sizeof(int), 1, fp );
3 → fwrite( A, sizeof(int), 1, fp );
4 →
```

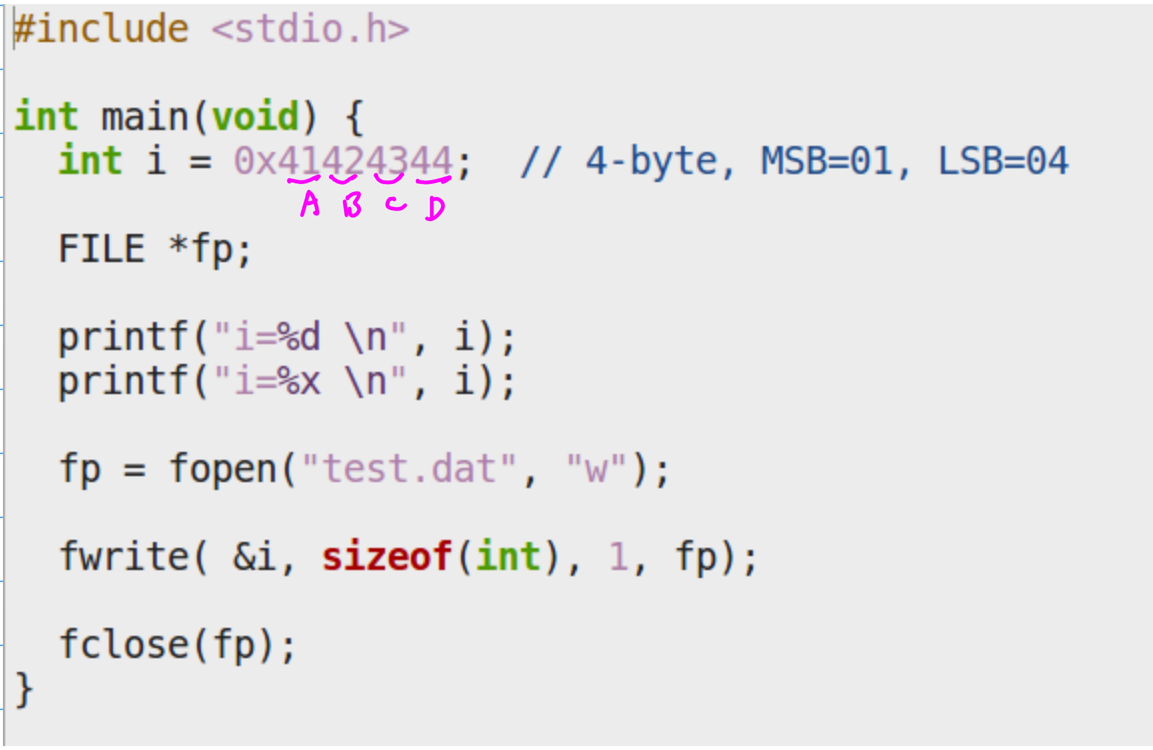
can move the file position pointer
without actual writing or reading
operations --- fseek



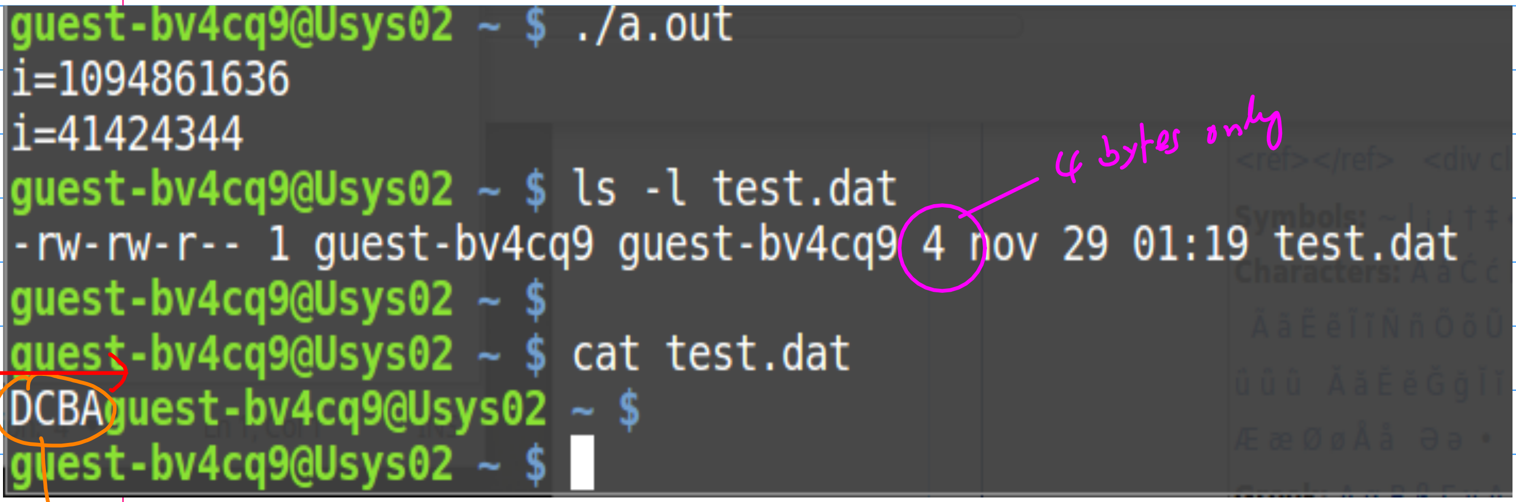
```
fseek( fp, 3*4, SEEK_SET )
```

```
fp → fwrite( A, sizeof(int), 1, fp );
```

Random Access



```
#include <stdio.h>

int main(void) {
  int i = 0x41424344;  // 4-byte, MSB=01, LSB=04

  FILE *fp;

  printf("i=%d \n", i);
  printf("i=%x \n", i);

  fp = fopen("test.dat", "w");

  fwrite( &i, sizeof(int), 1, fp);

  fclose(fp);
}
```



```
guest-bv4cq9@Usys02 ~ $ ./a.out
i=1094861636
i=41424344
guest-bv4cq9@Usys02 ~ $ ls -l test.dat
-rw-rw-r-- 1 guest-bv4cq9 guest-bv4cq9 4 nov 29 01:19 test.dat
guest-bv4cq9@Usys02 ~ $
guest-bv4cq9@Usys02 ~ $ cat test.dat
DCBAguest-bv4cq9@Usys02 ~ $
guest-bv4cq9@Usys02 ~ $ █
```

4 bytes only

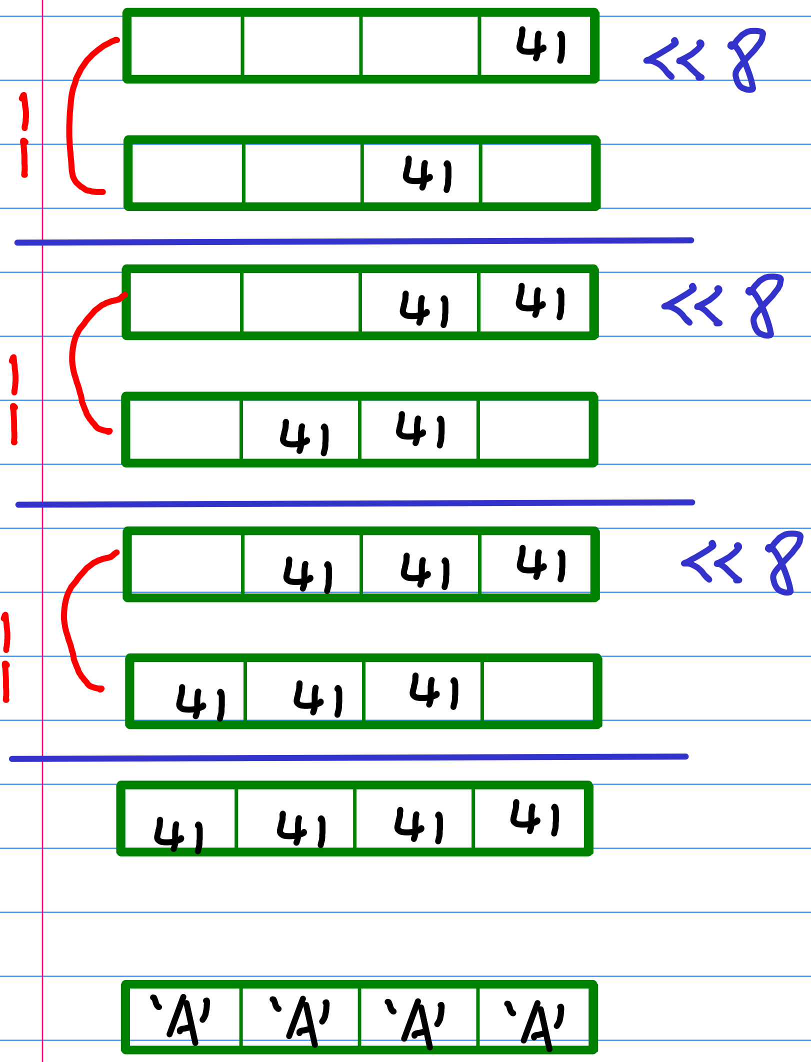
LSByte first

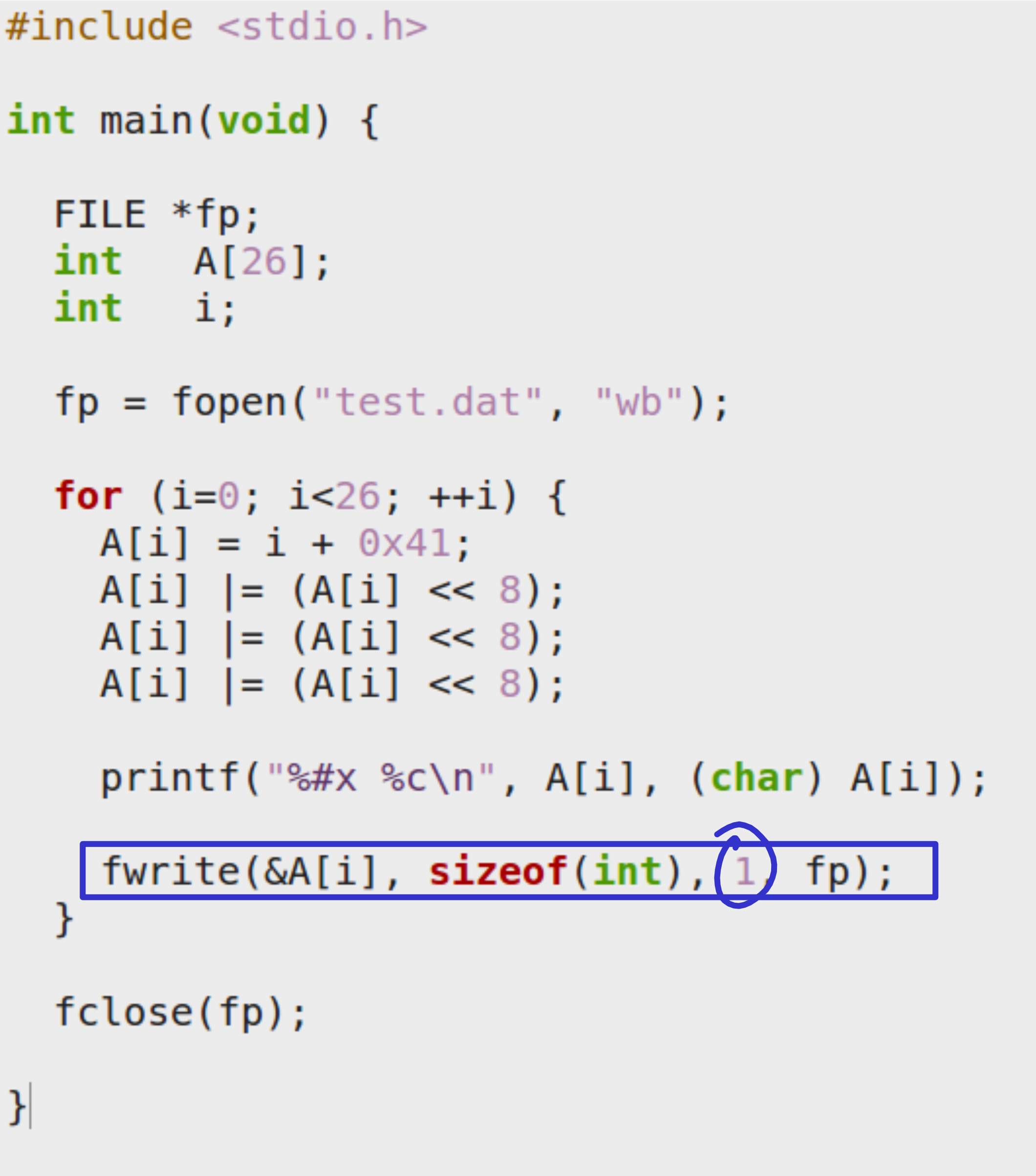
A B C D

i = 0x41 42 43 44

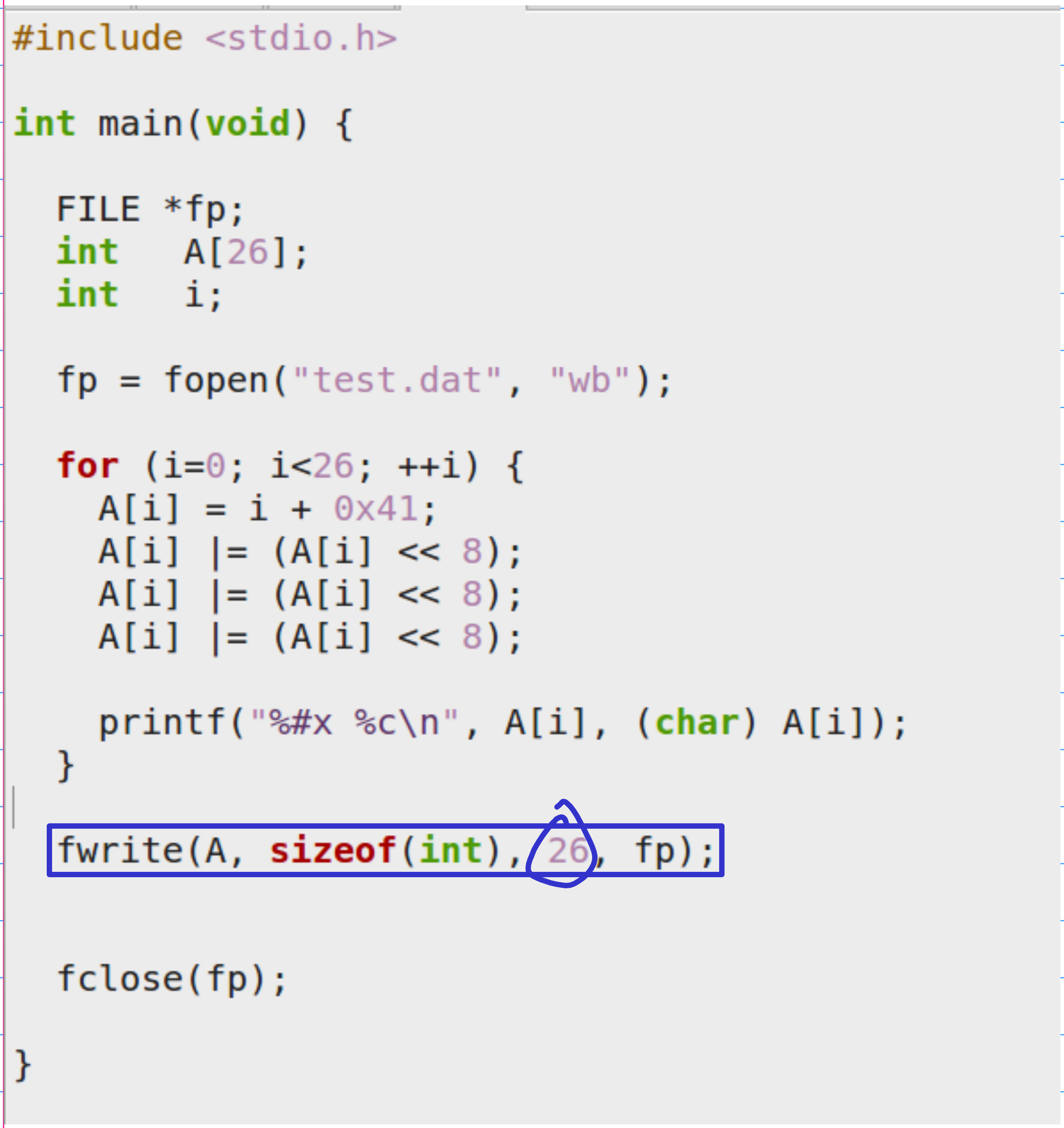
MSB LSB

bitwise OR

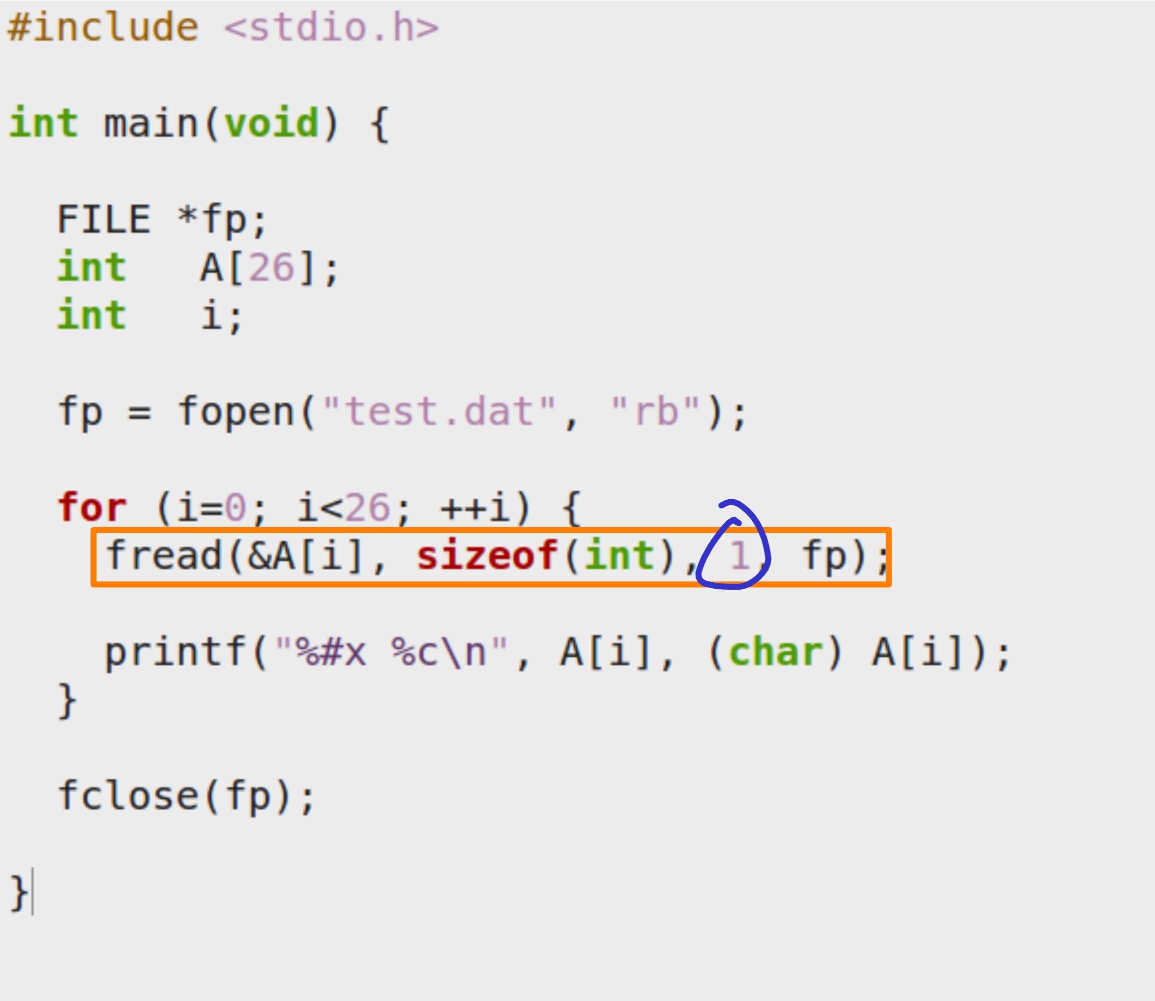




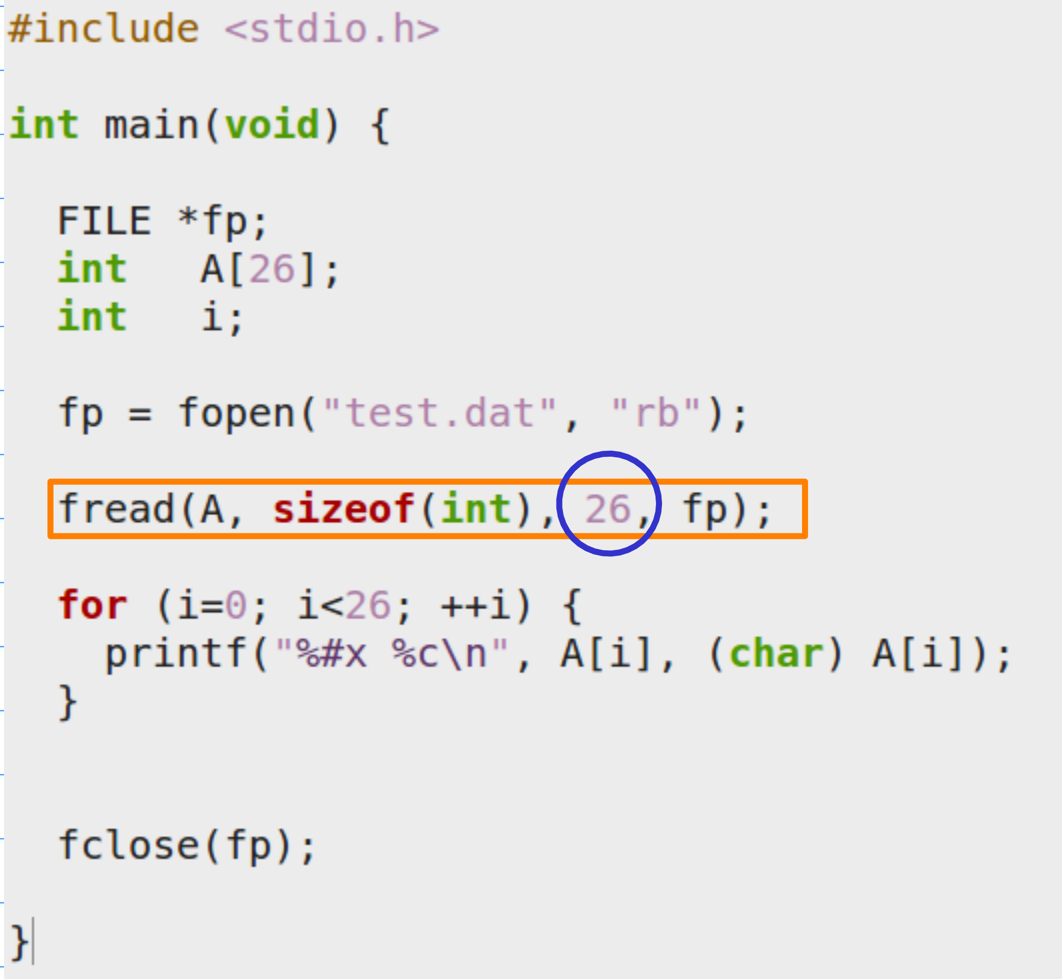
```c
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[26];
  int   i;

  fp = fopen("test.dat", "wb");

  for (i=0; i<26; ++i) {
    A[i] = i + 0x41;
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);

    printf("%#x %c\n", A[i], (char) A[i]);

    fwrite(&A[i], sizeof(int), 1, fp);
  }

  fclose(fp);

}
```



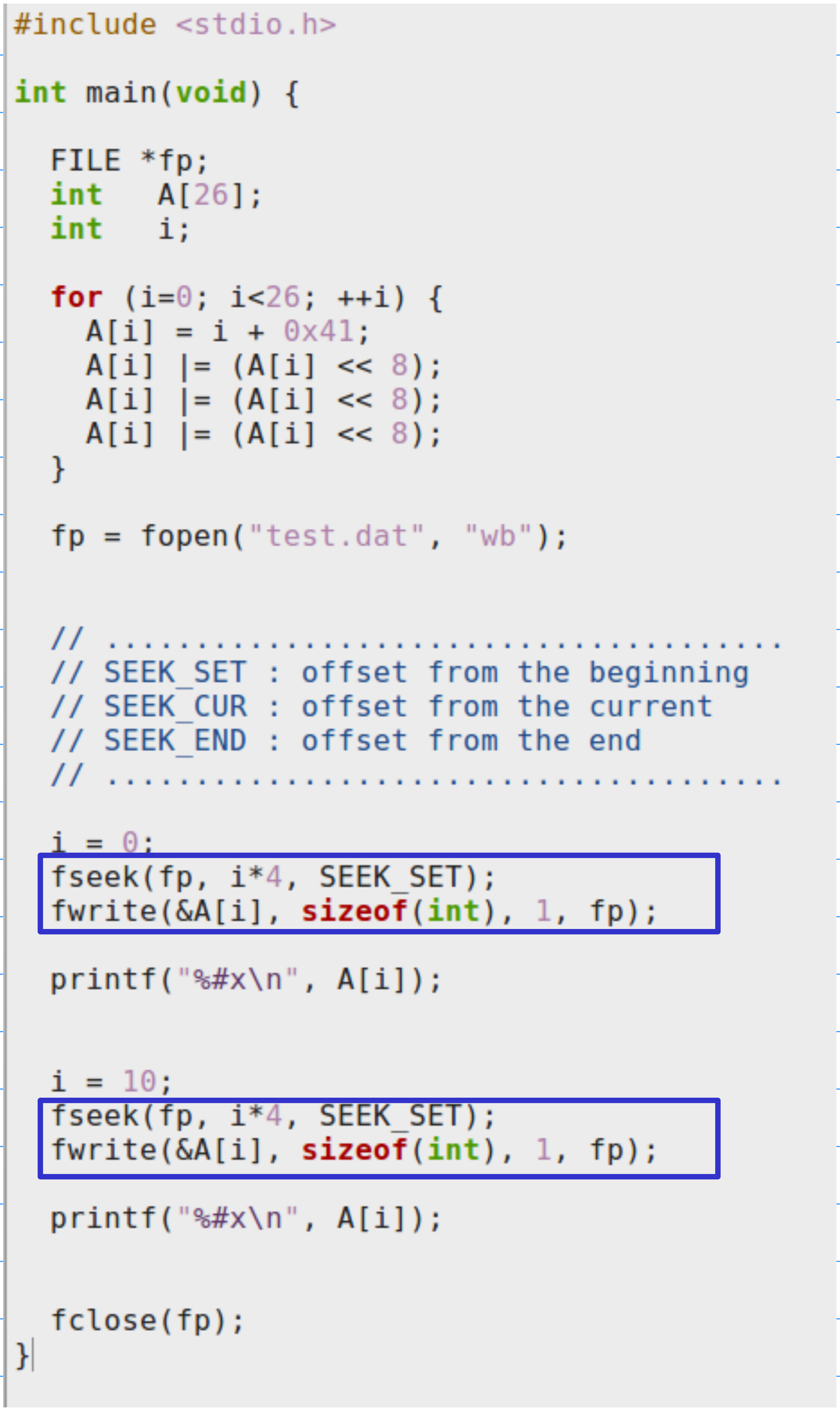
```c
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[26];
  int   i;

  fp = fopen("test.dat", "wb");

  for (i=0; i<26; ++i) {
    A[i] = i + 0x41;
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);

    printf("%#x %c\n", A[i], (char) A[i]);
  }

  fwrite(A, sizeof(int), 26, fp);


  fclose(fp);

}
```



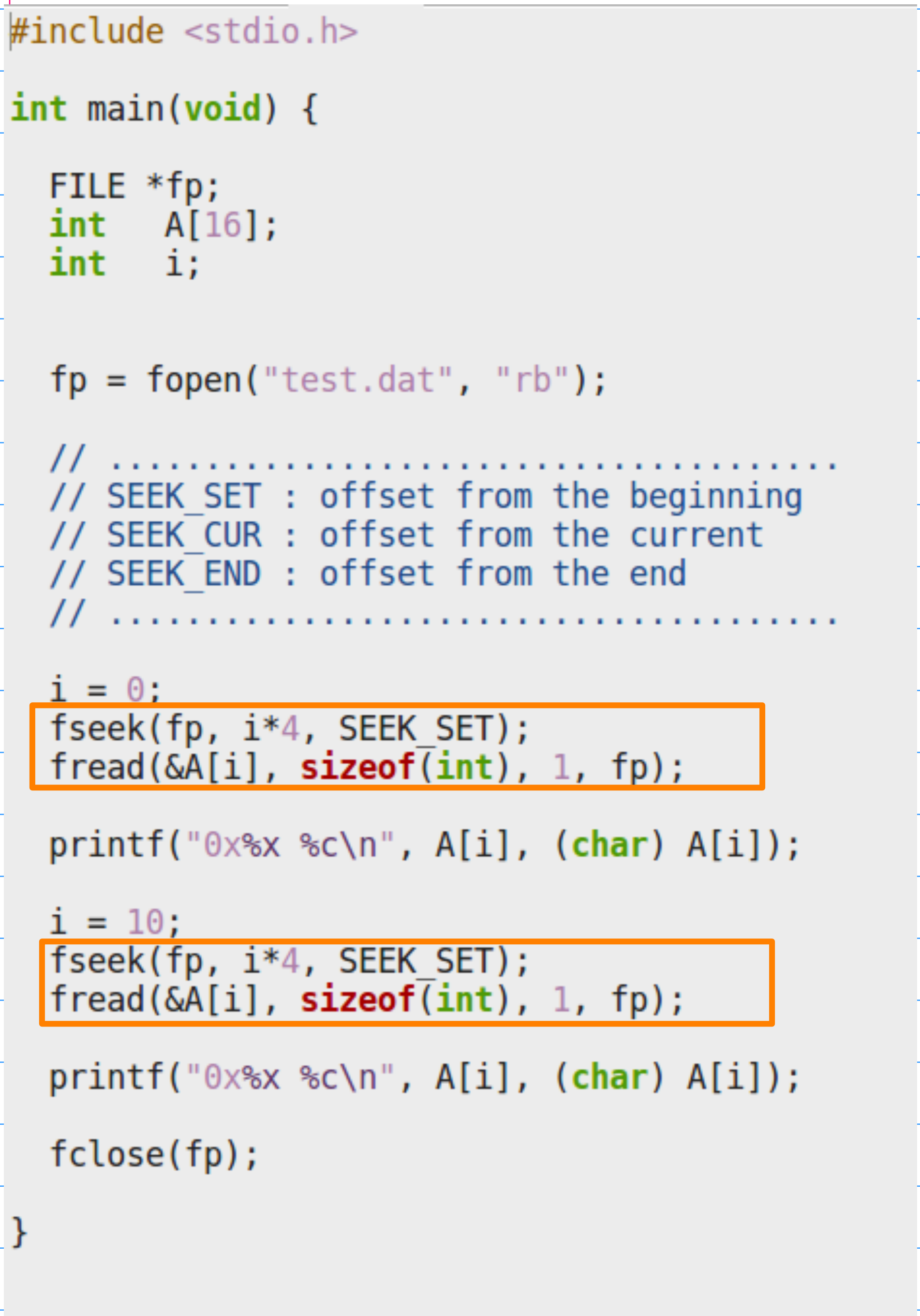
```
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[26];
  int   i;

  fp = fopen("test.dat", "rb");

  for (i=0; i<26; ++i) {
    fread(&A[i], sizeof(int), 1, fp);

    printf("%#x %c\n", A[i], (char) A[i]);
  }

  fclose(fp);

}
```



```c
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[26];
  int   i;

  fp = fopen("test.dat", "rb");

  fread(A, sizeof(int), 26, fp);

  for (i=0; i<26; ++i) {
    printf("%#x %c\n", A[i], (char) A[i]);
  }


  fclose(fp);

}
```

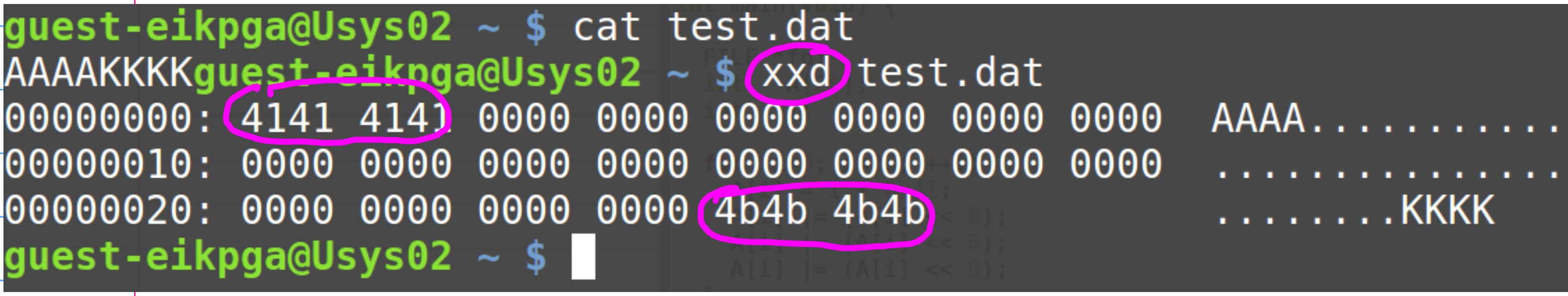


```c
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[26];
  int   i;

  for (i=0; i<26; ++i) {
    A[i] = i + 0x41;
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);
    A[i] |= (A[i] << 8);
  }

  fp = fopen("test.dat", "wb");


  // ......................................
  // SEEK_SET : offset from the beginning
  // SEEK_CUR : offset from the current
  // SEEK_END : offset from the end
  // ......................................

  i = 0;
  fseek(fp, i*4, SEEK_SET);
  fwrite(&A[i], sizeof(int), 1, fp);

  printf("%#x\n", A[i]);


  i = 10;
  fseek(fp, i*4, SEEK_SET);
  fwrite(&A[i], sizeof(int), 1, fp);

  printf("%#x\n", A[i]);


  fclose(fp);
}
```

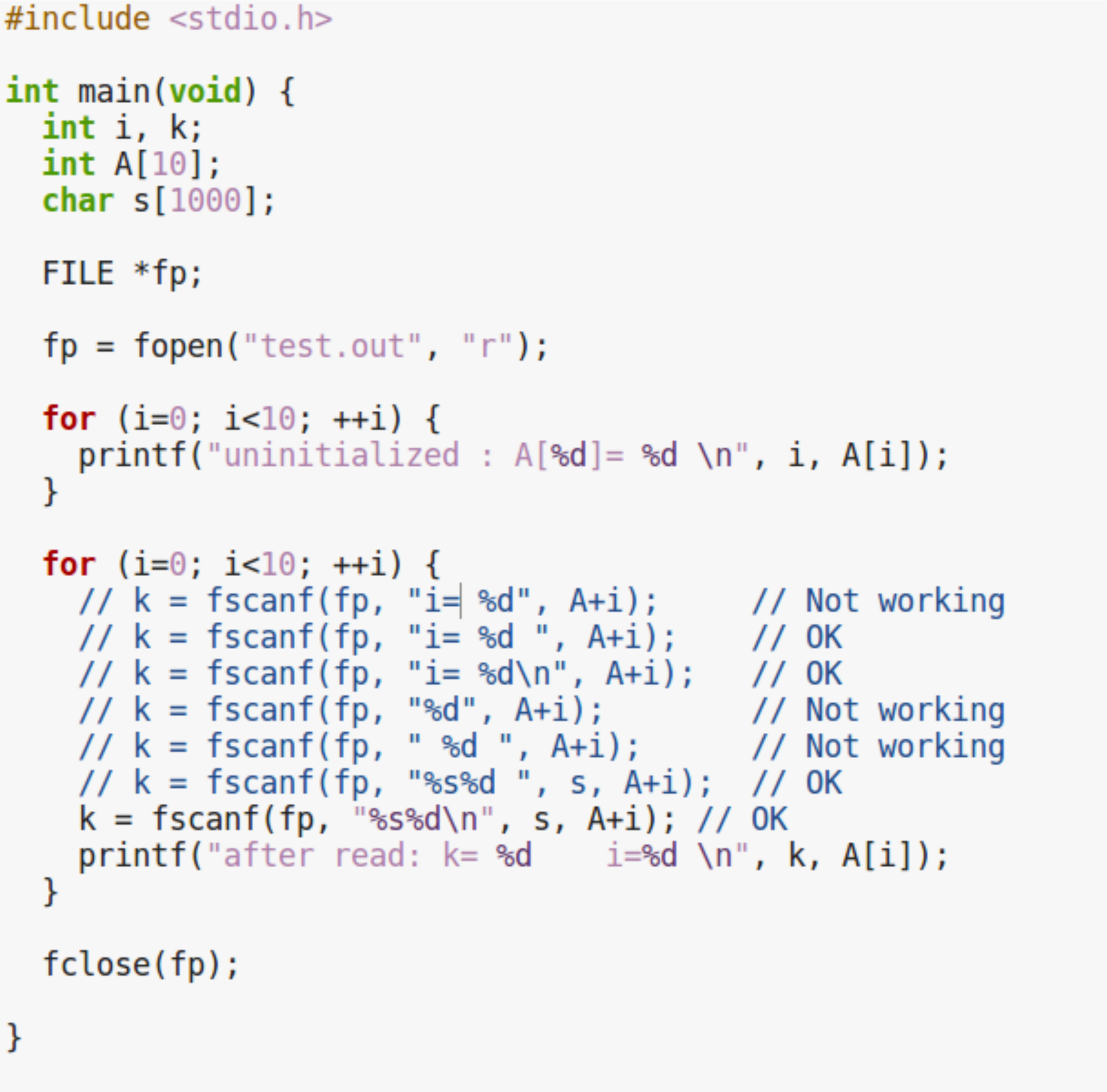


```c
#include <stdio.h>

int main(void) {

  FILE *fp;
  int   A[16];
  int   i;


  fp = fopen("test.dat", "rb");

  // ....................................
  // SEEK_SET : offset from the beginning
  // SEEK_CUR : offset from the current
  // SEEK_END : offset from the end
  // ....................................

  i = 0;
  fseek(fp, i*4, SEEK_SET);
  fread(&A[i], sizeof(int), 1, fp);

  printf("0x%x %c\n", A[i], (char) A[i]);

  i = 10;
  fseek(fp, i*4, SEEK_SET);
  fread(&A[i], sizeof(int), 1, fp);

  printf("0x%x %c\n", A[i], (char) A[i]);

  fclose(fp);

}
```

xxd

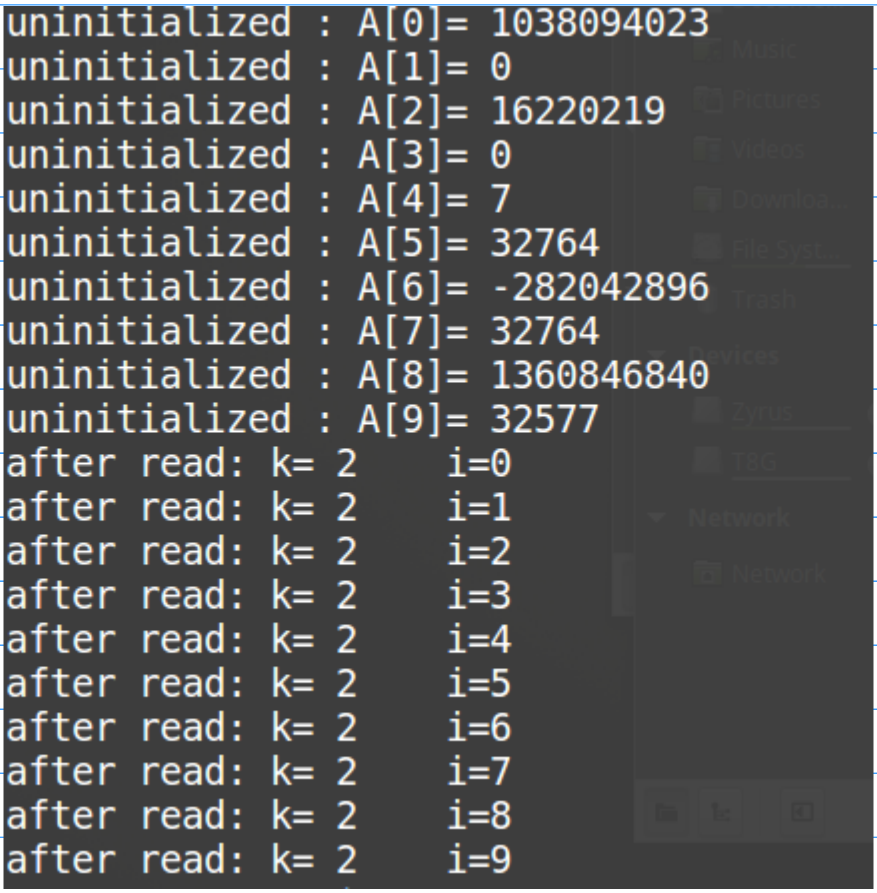


```
guest-eikpga@Usys02 ~ $ cat test.dat
AAAAKKKKguest-eikpga@Usys02 ~ $ xxd test.dat
00000000: 4141 4141 0000 0000 0000 0000 0000 0000  AAAA............
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 4b4b 4b4b            ........KKKK
guest-eikpga@Usys02 ~ $ 
```



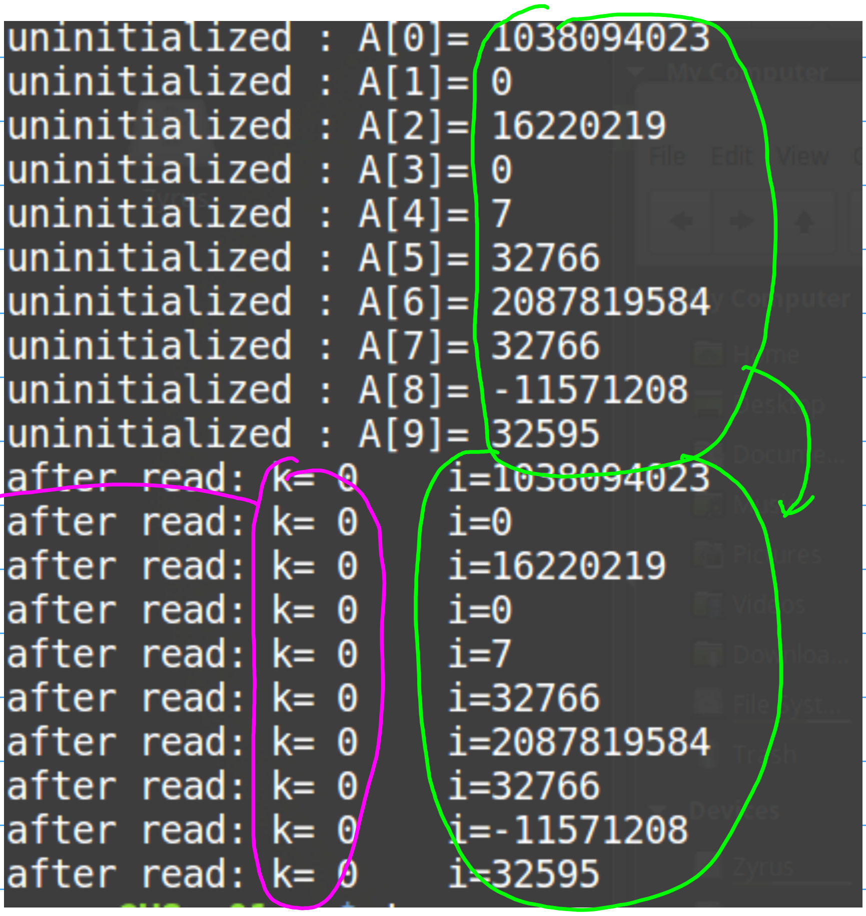
```c
#include <stdio.h>

int main(void) {
  int i, k;
  int A[10];
  char s[1000];

  FILE *fp;

  fp = fopen("test.out", "r");

  for (i=0; i<10; ++i) {
    printf("uninitialized : A[%d]= %d \n", i, A[i]);
  }

  for (i=0; i<10; ++i) {
    // k = fscanf(fp, "i= %d", A+i);     // Not working
    // k = fscanf(fp, "i= %d ", A+i);    // OK
    // k = fscanf(fp, "i= %d\n", A+i);   // OK
    // k = fscanf(fp, "%d", A+i);        // Not working
    // k = fscanf(fp, " %d ", A+i);      // Not working
    // k = fscanf(fp, "%s%d ", s, A+i);  // OK
    k = fscanf(fp, "%s%d\n", s, A+i); // OK
    printf("after read: k= %d    i=%d \n", k, A[i]);
  }

  fclose(fp);

}
```

Working Cases



```
uninitialized : A[0]= 1038094023
uninitialized : A[1]= 0
uninitialized : A[2]= 16220219
uninitialized : A[3]= 0
uninitialized : A[4]= 7
uninitialized : A[5]= 32764
uninitialized : A[6]= -282042896
uninitialized : A[7]= 32764
uninitialized : A[8]= 1360846840
uninitialized : A[9]= 32577
after read: k= 2     i=0
after read: k= 2     i=1
after read: k= 2     i=2
after read: k= 2     i=3
after read: k= 2     i=4
after read: k= 2     i=5
after read: k= 2     i=6
after read: k= 2     i=7
after read: k= 2     i=8
after read: k= 2     i=9
```

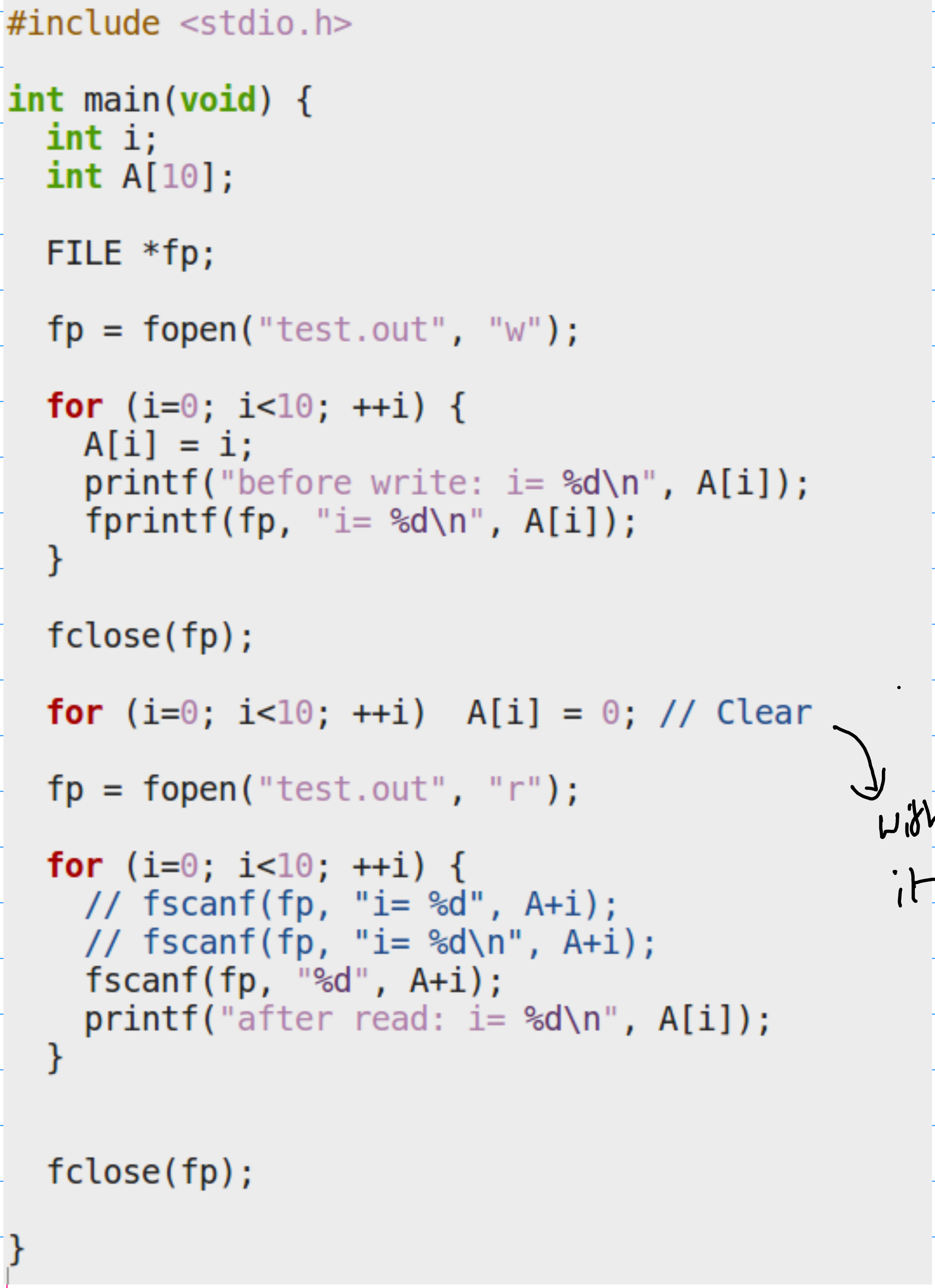
Not working cases



```
uninitialized : A[0]= 1038094023
uninitialized : A[1]= 0
uninitialized : A[2]= 16220219
uninitialized : A[3]= 0
uninitialized : A[4]= 7
uninitialized : A[5]= 32766
uninitialized : A[6]= 2087819584
uninitialized : A[7]= 32766
uninitialized : A[8]= -11571208
uninitialized : A[9]= 32595
after read: k= 0    i=1038094023
after read: k= 0    i=0
after read: k= 0    i=16220219
after read: k= 0    i=0
after read: k= 0    i=7
after read: k= 0    i=32766
after read: k= 0    i=2087819584
after read: k= 0    i=32766
after read: k= 0    i=-11571208
after read: k= 0    i=32595
```

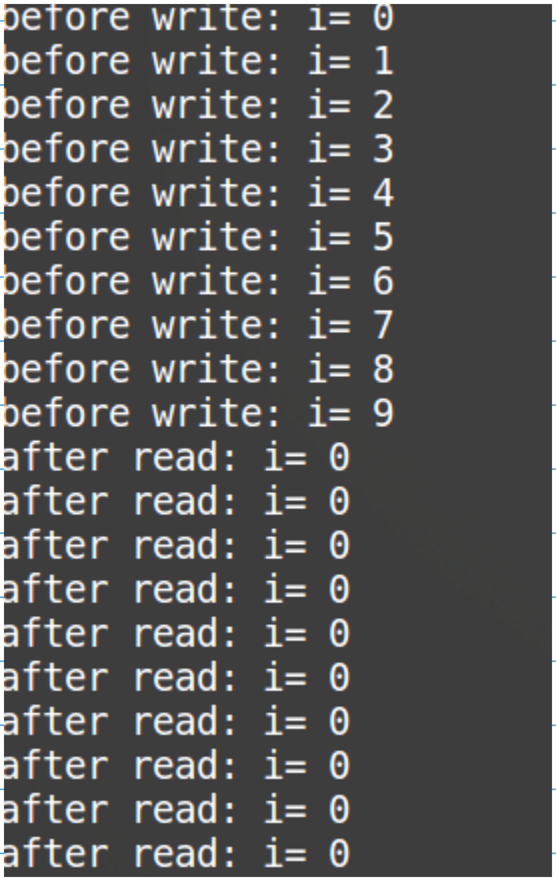
nothing is read

A[i] didn't changed



```c
#include <stdio.h>

int main(void) {
  int i;
  int A[10];

  FILE *fp;

  fp = fopen("test.out", "w");

  for (i=0; i<10; ++i) {
    A[i] = i;
    printf("before write: i= %d\n", A[i]);
    fprintf(fp, "i= %d\n", A[i]);
  }

  fclose(fp);

  for (i=0; i<10; ++i)  A[i] = 0; // Clear

  fp = fopen("test.out", "r");

  for (i=0; i<10; ++i) {
    // fscanf(fp, "i= %d", A+i);
    // fscanf(fp, "i= %d\n", A+i);
    fscanf(fp, "%d", A+i);
    printf("after read: i= %d\n", A[i]);
  }


  fclose(fp);

}
```

without this
it seemed that
it's working
but it's
not



```
before write: i= 0
before write: i= 1
before write: i= 2
before write: i= 3
before write: i= 4
before write: i= 5
before write: i= 6
before write: i= 7
before write: i= 8
before write: i= 9
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
after read: i= 0
```