Fibonacci Sequence.

$$F_n = F_{n-1} + F_{n-2}$$

$$F_0 = 0$$

$$F_1 = 1$$

input: n

output: F_0 F_1 ... F_n

-

```
int    fibo ( int n ){
    if (n<=1) Return 1 ; else if (n<=20) Return 0 ;
Return  fibo (n-2) + fibo (n-1) ;

}

int main ( ) {
    int n ;
    scanf("%d", &n) ;
Printf("%d", fibo (n)) ;
}
```

F_n

$2^{20} \times 2^{10} \times 2^{10} \times 2^{10}$

2^{30} GB

F_{10}

F_9

F_8

F_7

F_8

F_6

F_7

2^{n+1}

2^{10^9} bytes

n 0 2^{101}

$n=0$

n=6

F_0 F_1

$F_2 = F_1 + F_0$ | $F_a = 1$ $F_b = 0$

for (i = 2 ; i ≤ n ; i++) {

F = F_a + F_b

F_{i-1} F_{i-2}

$F_b = F_a$; $F_a = F$

}

Return F ;

10

f_a f_b

9 , 8

11

$n = \infty$

$f_{S0} \rightarrow$ crosses

f_{S1}

$f_{S2} \rightarrow$

4 bytes

INT_MAX

STRING operation

F = Fopen("input")

→ scanf

→ gets ( )

→ fgets (f, "%s", st)

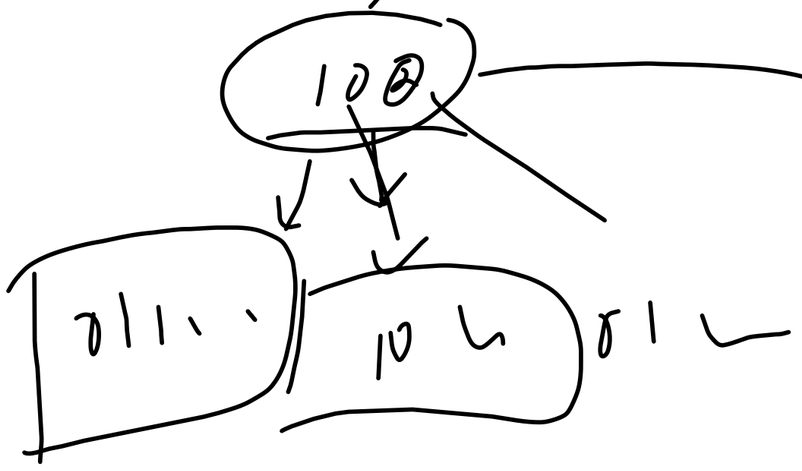
)

-

Name Roll Grade

Fname, Lname, 012 → A

./a.exe



scanf("%d", &n);

4 byte

0x00010000

4 32

Convert String to Int

int / unsigned / long / LL

int atoi(str) char*

len = Strlen(str) → res = 0

for(i=0; i<len; i++)

→ r = str[i] − '0'; // 48

res = res*10 + r

"1234" → 2^3 2^2 → O(n)

str[0] = '1'

str[1] = '2'

str[2] = '3'

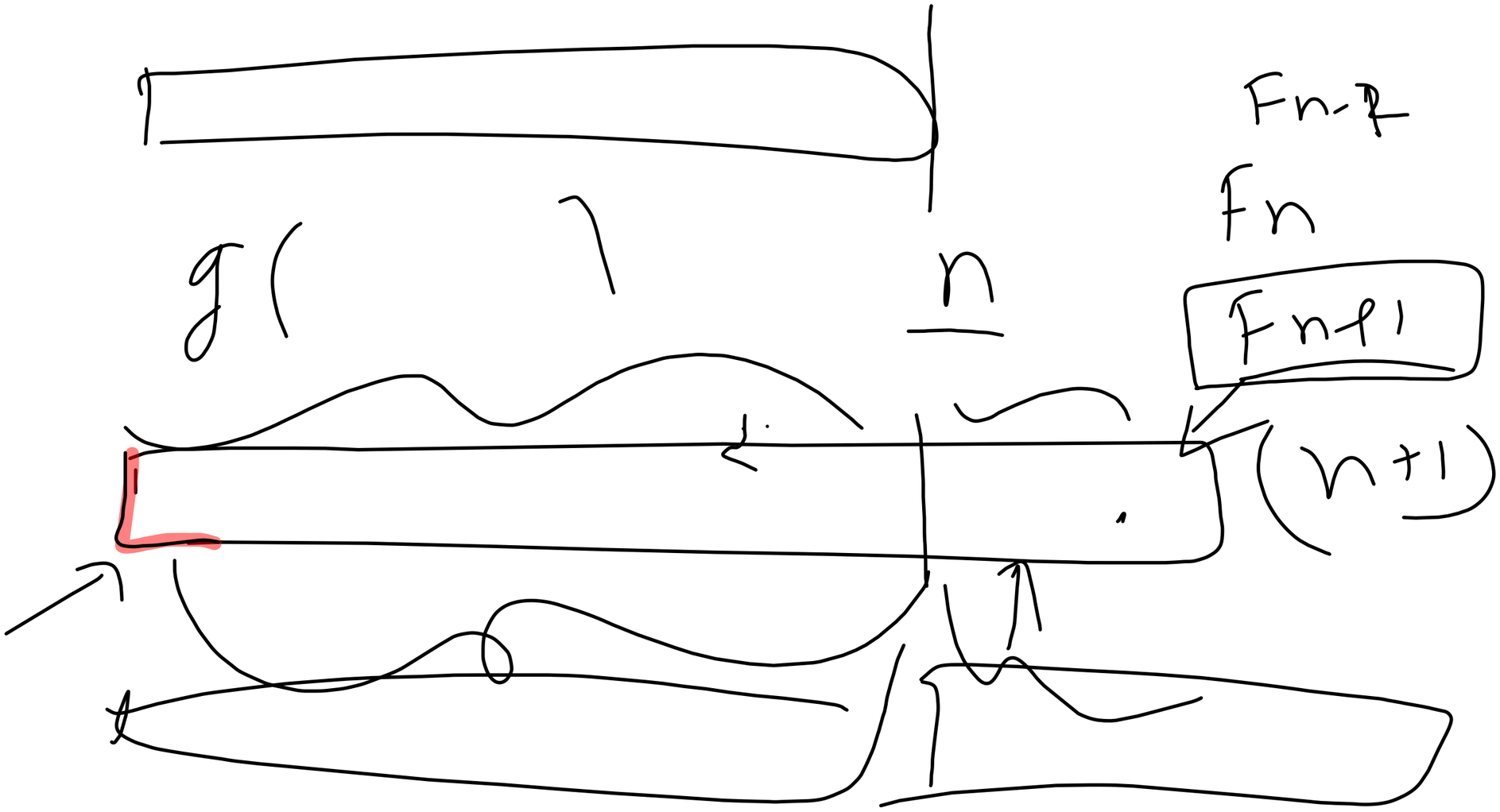
str[3] = '4'

res = 1*10 + 2

(1*10 + 2)*10 + 3

((1*10) + 2)*10 + 3)*10 + 4

O(n) < 1

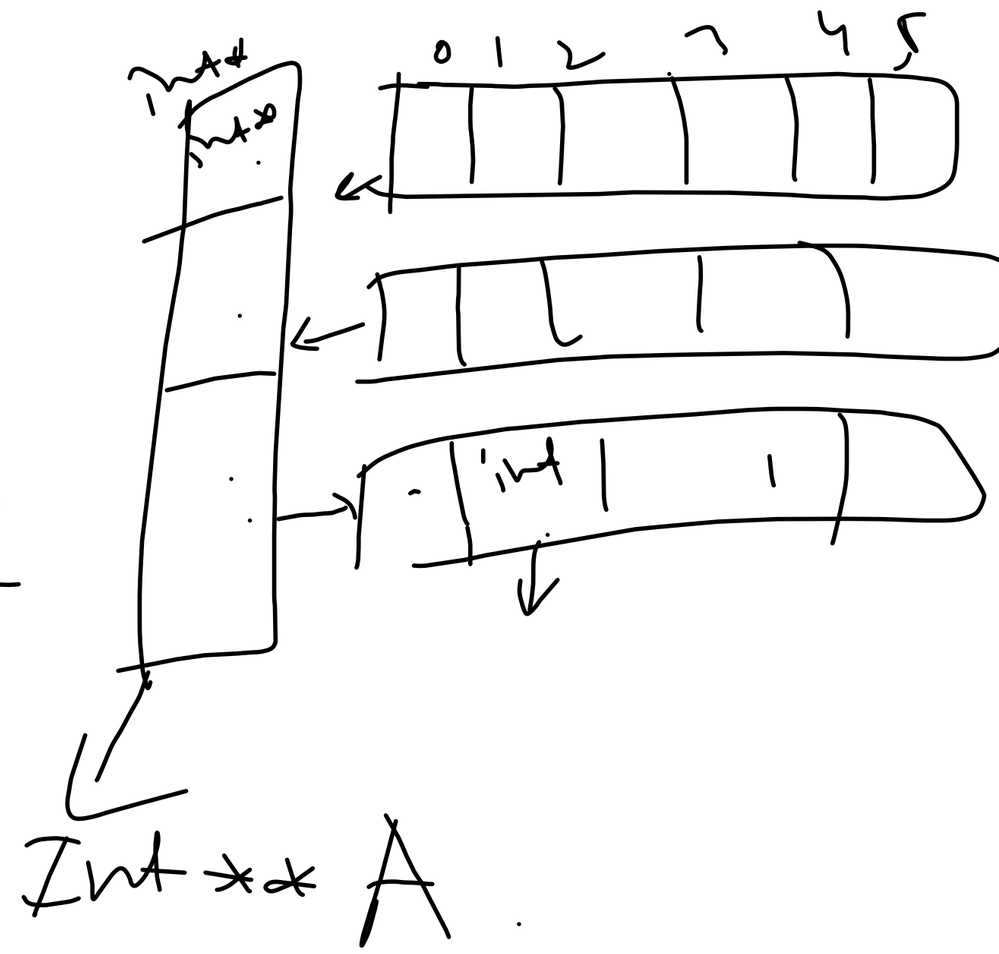

Fn-2
Fn
Fn-1
n
(n+1)

int ** allocate_2D_Matrix(int n , int m)

scan_2D Matrix(

show_2D Matrix(

A[2][1]



int ** allocate (int n ; int m) { n x n

int ** A ; int i ;

A = malloc (n * sizeof (int *))

for (i=0 ; i<n ; i++) {

A[i] = malloc (m * sizeof (int)) m

} }

Return A .

Square & Multiply

a^n

Square & Multiply

1234

a^n

res = res * 10 + r

$((1 * 10 + 2) * 10 + 3) * 10 + 4$

$$\frac{\left(\left(a^{b_0}\right)^2 \cdot a^{b_1}\right)^2 \cdot a^{b_2} a^n}{a^{b_0+2+b_1}}$$

&

n

$30 \rightarrow k = \log_2(n)$ $\quad n = 32 \quad w = 2^k$

$$n = (b_0\ b_1\ \ldots\ b_k)_2$$

$$2^k \cdot b_0 + 2^{k-1} b_1$$

$$d = a^{\left(\left(b_0 * 2 + b_1\right) * 2 + b_2\right) * 2 + \ldots\ b_k}$$