24. Write a C Program to find the maximum size of a file that can be stored in the below file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks.

Test Case:

● Check that the start blocks and the required contiguous blocks are free.

● If free, allocate those blocks to the file.

● If not free, find the next available contiguous blocks.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define BLOCK\_SIZE 8192 // 8 KB

#define POINTERS\_PER\_BLOCK (BLOCK\_SIZE / sizeof(int))

int main() {

int max\_file\_size = 0;

int direct\_blocks = 12;

int single\_indirect\_blocks = POINTERS\_PER\_BLOCK;

int double\_indirect\_blocks = pow(POINTERS\_PER\_BLOCK, 2);

int triple\_indirect\_blocks = pow(POINTERS\_PER\_BLOCK, 3);

int block\_size\_bytes = BLOCK\_SIZE;

// calculate the maximum size of a file in bytes

max\_file\_size = direct\_blocks \* block\_size\_bytes // direct blocks

+ single\_indirect\_blocks \* block\_size\_bytes // single indirect blocks

+ double\_indirect\_blocks \* pow(block\_size\_bytes, 2) // double indirect blocks

+ triple\_indirect\_blocks \* pow(block\_size\_bytes, 3); // triple indirect blocks

printf("The maximum size of a file that can be stored in this file system is %d bytes\n", max\_file\_size);

return 0;

}

