#include <stdio.h>

int contiguousAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int linkedAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int indexedAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int main()

{

int numBlocks = 100;

int blockSize\_a = 1;

int operations\_a\_contiguous = contiguousAllocation(blockSize\_a, numBlocks);

int operations\_a\_linked = linkedAllocation(blockSize\_a, numBlocks);

int operations\_a\_indexed = indexedAllocation(blockSize\_a, numBlocks);

printf("Test Case a:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_a\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_a\_linked);

printf("Indexed Allocation: %d disk I/O operations\n\n", operations\_a\_indexed);

int blockSize\_b = 50;

int operations\_b\_contiguous = contiguousAllocation(blockSize\_b, numBlocks);

int operations\_b\_linked = linkedAllocation(blockSize\_b, numBlocks);

int operations\_b\_indexed = indexedAllocation(blockSize\_b, numBlocks);

printf("Test Case b:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_b\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_b\_linked);

printf("Indexed Allocation: %d disk I/O operations\n\n", operations\_b\_indexed);

int blockSize\_c = 100, operation;

int operations\_c\_contiguous = contiguousAllocation(blockSize\_c, numBlocks);

int operations\_c\_linked = linkedAllocation(blockSize\_c, numBlocks);

int operations\_c\_indexed = indexedAllocation(blockSize\_c, numBlocks);

printf("Test Case c:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_c\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_c\_linked);

printf("Indexed Allocation: %d disk I/O operations\n", operation);

}#include <stdio.h>

int contiguousAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int linkedAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int indexedAllocation(int blockSize, int numBlocks) {

int operations = 0;

if (blockSize == 1 || blockSize == numBlocks)

operations = 1;

else if (blockSize > 1 && blockSize < numBlocks)

operations = 2;

return operations;

}

int main()

{

int numBlocks = 100;

int blockSize\_a = 1;

int operations\_a\_contiguous = contiguousAllocation(blockSize\_a, numBlocks);

int operations\_a\_linked = linkedAllocation(blockSize\_a, numBlocks);

int operations\_a\_indexed = indexedAllocation(blockSize\_a, numBlocks);

printf("Test Case a:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_a\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_a\_linked);

printf("Indexed Allocation: %d disk I/O operations\n\n", operations\_a\_indexed);

int blockSize\_b = 50;

int operations\_b\_contiguous = contiguousAllocation(blockSize\_b, numBlocks);

int operations\_b\_linked = linkedAllocation(blockSize\_b, numBlocks);

int operations\_b\_indexed = indexedAllocation(blockSize\_b, numBlocks);

printf("Test Case b:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_b\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_b\_linked);

printf("Indexed Allocation: %d disk I/O operations\n\n", operations\_b\_indexed);

int blockSize\_c = 100, operation;

int operations\_c\_contiguous = contiguousAllocation(blockSize\_c, numBlocks);

int operations\_c\_linked = linkedAllocation(blockSize\_c, numBlocks);

int operations\_c\_indexed = indexedAllocation(blockSize\_c, numBlocks);

printf("Test Case c:\n");

printf("Contiguous Allocation: %d disk I/O operations\n", operations\_c\_contiguous);

printf("Linked Allocation: %d disk I/O operations\n", operations\_c\_linked);

printf("Indexed Allocation: %d disk I/O operations\n", operation);

}

OUTPUT

Test Case a:

Contiguous Allocation: 1 disk I/O operations

Linked Allocation: 1 disk I/O operations

Indexed Allocation: 1 disk I/O operations

Test Case b:

Contiguous Allocation: 2 disk I/O operations

Linked Allocation: 2 disk I/O operations

Indexed Allocation: 2 disk I/O operations

Test Case c:

Contiguous Allocation: 1 disk I/O operations

Linked Allocation: 1 disk I/O operations

Indexed Allocation: 33 disk I/O operations

--------------------------------

Process exited after 0.03745 seconds with return value 43

Press any key to continue . . .