Programiz
C Online Compiler

Interactive C Course

main.c

Run

```c
#include <stdio.h>
#define CONTIGUOUS 0
#define LINKED 1
#define INDEXED 2
#define FILE_SIZE 100
int calculateContiguousIO(int position) {
    if (position < 0 || position > FILE_SIZE) {
        printf("Invalid position!\n");
        return -1;
    }
    int operations = 0;
    if (position <= FILE_SIZE / 2) {
        operations = position;
    } else {
        operations = FILE_SIZE - position;
    }
    return operations;
}
int calculateLinkedIO(int position) {
    if (position < 0 || position > FILE_SIZE) {
        printf("Invalid position!\n");
        return -1;
    }
    return position + 1;
}
int calculateIndexedIO(int position) {
    if (position < 0 || position > FILE_SIZE) {
        printf("Invalid position!\n");
```

Output

Clear

```
/tmp/2hOR8qqmty.o
Enter the allocation strategy (0 - contiguous, 1 - linked, 2 - indexed): 1
Enter the position to add the block (0 - 100): 54
Number of disk I/O operations required: 55
```

Programiz
C Online Compiler

Interactive C Course

main.c

Run

Output

Clear

```c
    if (position < 0 || position > FILE_SIZE) {
        printf("Invalid position!\n");
        return -1;
    }
    int operations = 1;
    if (position > 0 && position % (FILE_SIZE / 10) == 0) {
        operations++; // Additional operation for accessing the indexed block
    }
    return operations;
}
int main() {
    int allocation_strategy;
    int position;
    printf("Enter the allocation strategy (0 - contiguous, 1 - linked, 2 -
        indexed): ");
    scanf("%d", &allocation_strategy);
    printf("Enter the position to add the block (0 - %d): ", FILE_SIZE);
    scanf("%d", &position);
    int disk_io_operations;
    switch (allocation_strategy) {
        case CONTIGUOUS:
            disk_io_operations = calculateContiguousIO(position);
            break;
        case LINKED:
            disk_io_operations = calculateLinkedIO(position);
            break;
        case INDEXED:
```

```
/tmp/2hOR8qqmty.o
Enter the allocation strategy (0 - contiguous, 1 - linked, 2 - indexed): 1
Enter the position to add the block (0 - 100): 54
Number of disk I/O operations required: 55
```

Programiz
C Online Compiler

Interactive C Course

main.c

Run

```c
    int position;
    printf("Enter the allocation strategy (0 - contiguous, 1 - linked, 2 - indexed): ");
    scanf("%d", &allocation_strategy);
    printf("Enter the position to add the block (0 - %d): ", FILE_SIZE);
    scanf("%d", &position);
    int disk_io_operations;
    switch (allocation_strategy) {
        case CONTIGUOUS:
            disk_io_operations = calculateContiguousIO(position);
            break;
        case LINKED:
            disk_io_operations = calculateLinkedIO(position);
            break;
        case INDEXED:
            disk_io_operations = calculateIndexedIO(position);
            break;
        default:
            printf("Invalid allocation strategy!\n");
            return -1;
    }
    if (disk_io_operations >= 0) {
        printf("Number of disk I/O operations required: %d\n", disk_io_operations);
    }
    return 0;
}
```

Output

Clear

```
/tmp/2hOR8qqmty.o
Enter the allocation strategy (0 - contiguous, 1 - linked, 2 - indexed): 1
Enter the position to add the block (0 - 100): 54
Number of disk I/O operations required: 55
```