

**Experiment-34: Consider a file system where the records of the file are stored one after another both physically and logically. A record of the file can only be accessed by reading all the previous records. Design a C program to simulate the file allocation strategy.**

**Aim:**

To simulate the file allocation strategy where records of the file are stored one after another both physically and logically, and a record can only be accessed by reading all the previous records.

**Procedure:**

1. Take the number of records in the file as input.
2. Store the records sequentially in memory (using an array).
3. Access a record by sequentially reading all the previous records (simulating the behavior of the allocation strategy).
4. Display the records as they are accessed.

**C Program:**

```
#include <stdio.h>
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter the number of records in the file: ");
```

```
    scanf("%d", &n);
```

```
    int file[n];
```

```
    printf("Enter the records: \n");
```

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

```
        printf("Record %d: ", i + 1);
```

```
        scanf("%d", &file[i]);
```

```
    }
```

```
    int record;
```

```
    printf("Enter the record number to access (1 to %d): ", n);
```

```
scanf("%d", &record);

if (record < 1 || record > n) {
    printf("Invalid record number.\n");
} else {
    printf("Accessing records sequentially:\n");
    for (int i = 0; i < record; i++) {
        printf("Record %d: %d\n", i + 1, file[i]);
    }
}

return 0;
}
```

Output:

## Output

```
Enter the number of records in the file: 5
Enter the records:
Record 1: 4
Record 2: 5
Record 3: 8
Record 4: 6
Record 5: 4
Enter the record number to access (1 to 5): 5
Accessing records sequentially:
Record 1: 4
Record 2: 5
Record 3: 8
Record 4: 6
Record 5: 4

=== Code Execution Successful ===192372048|
```