

To implement indexed file allocation strategy

Program:

```
#include<stdio.h>

int main()

{

int n,m[20],i,j,index[20],s[20],b[20][20],x;

printf("enter no of files");

scanf("%d",&n);

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

{

printf("enter index block of file %d:",i+1);

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

printf("enter no of blocks occupied by file %d ",i+1);

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

printf("enter blocks of file %d:",i+1);

for(j=0;j<m[i];j++)

scanf("%d",&b[i][j]);

}

printf("\nfile\tindex\tlength\n");

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

{

printf("%d\t%d\t%d\n",i+1,index[i],m[i]);

}

printf("\nenter file name");

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

```

printf("file name is:%d\n",x);

i=x-1;

printf("index is:%d\n",index[i]);

printf("block occupied are:\n");

for(j=0;j<m[i];j++)

printf("%3d->%d\n",index[i],b[i][j]);

return 0;

}

```

Output:

```

20A91A05B6@Linux ~
[20A91A05B6@Linux ~]$ cc index.c
[20A91A05B6@Linux ~]$ ./a.out
enter no of files3
enter index block of file 1:4
enter no of blocks occupied by file 1 2
enter blocks of file 1:1
2
enter index block of file 2:5
enter no of blocks occupied by file 2 2
enter blocks of file 2:6
8
enter index block of file 3:9
enter no of blocks occupied by file 3 2
enter blocks of file 3:9
0

file   index   length
1       4       2
2       5       2
3       9       2

enter file name2
file name is:2
index is:5
block occupied are:
5->6
5->8

```