# **35. Consider a file system that brings all the file pointers together into an index block. The ith entry in the index block points to the ith block of the file. Design a C program to simulate the file allocation strategy.**

**Aim:** To write a C program to simulate the file allocation strategy.

# **ALGORITHM:**

* 1. Define the structure of a block that will be stored in the file.
  2. Create a file to represent the indexed file.
  3. Initialize an index block that contains pointers to data blocks.
  4. To write a new block:
  5. Prompt the user for the block number and the data to be written to theblock.
  6. Update the corresponding entry in the index block to point to the new datablock.
  7. Write the data block to the file.
  8. To read a specific block:
  9. Prompt the user for the block number they want to access.
  10. Use the index block to find the pointer to the requested data block.11.Read and display the data in the requested data block.
  11. Continue this process until the user decides to exit.

**Program:**

#include<stdio.h> #include<conio.h> #include<stdlib.h> int main()

{

int f[50], index[50],i, n, st, len, j, c, k, ind,count=0; for(i=0;i<50;i++)

f[i]=0;

x:printf("Enter the index block: "); scanf("%d",&ind);

if(f[ind]!=1)

{

printf("Enter no of blocks needed and no of files for the index %d on the disk :

\n", ind);

scanf("%d",&n);

}

else

{

printf("%d index is already allocated \n",ind); goto x;

}

y: count=0; for(i=0;i<n;i++)

{

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

if(f[index[i]]==0) count++;

}

if(count==n)

{

for(j=0;j<n;j++) f[index[j]]=1; printf("Allocated\n"); printf("File Indexed\n"); for(k=0;k<n;k++)

printf("%d >%d : %d\n",ind,index[k],f[index[k]]);

}

else

{

printf("File in the index is already allocated \n"); printf("Enter another file indexed");

goto y;

}

printf("Do you want to enter more file(Yes - 1/No - 0)"); scanf("%d", &c);

if(c==1) goto x; else exit(0);

getch();

}**Output:**

