**36. With linked allocation, each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. Each block contains a pointer to the next block. Design a C program to simulate the file allocation strategy.**

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

# **ALGORITHM:**

* 1. Define the structure of a block that will be stored in the file. Each blockcontains a pointer to the next block.
  2. Create a file to represent the linked allocation.
  3. Create a directory entry for the file containing a pointer to the first and lastblocks.
  4. To write a new block:
  5. Prompt the user for the block data.
  6. Allocate a new block in the file.
  7. If it's the first block, update the directory entry to point to it as both the firstand last block.
  8. If it's not the first block, update the previous block to point to the newblock.
  9. Update the new block's pointer to the next block (usually NULL for the lastblock).

# **Program:**

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

{

int f[50], p,i, st, len, j, c, k, a; for(i=0;i<50;i++)

f[i]=0;

printf("Enter how many blocks already allocated: "); scanf("%d",&p);

printf("Enter blocks already allocated: "); for(i=0;i<p;i++)

{

scanf("%d",&a); f[a]=1;

}

x: printf("Enter index starting block and length: ");

scanf("%d%d", &st,&len); k=len;

if(f[st]==0)

{

for(j=st;j<(st+k);j++)

{

if(f[j]==0)

{ f[j]=1;

printf("%d >%d\n",j,f[j]);

}

else

{

printf("%d Block is already allocated \n",j); k++;

}

}

}

else

printf("%d starting block is already allocated \n",st); printf("Do you want to enter more file(Yes - 1/No - 0)"); scanf("%d", &c);

if(c==1)

goto x; else exit(0);

}

# **Output:**

