**Implement Contiguous file allocation technique**

**Algorithm:**

* + - 1. Start.

1. Read the total number of files that need to be stored contiguously in the memory.
2. I) Read the capacity or the number of memory locations needed to store the file.

II) Read the starting address of the file.

Repeat this step until the total number of files are checked.

1. Assign the value 1 to all the memory locations to ensure that at the beginning all the memory locations are available or free.
2. Take the starting address of file and check if that address is free or not i.e if it has value 1

If yes,

1. Go from starting address to last address of the file. At each memory location check if it is free if yes increment the count variable.
2. Compare count variable with the capacity of the file if both are equal then assign value 0 to all the memory locations starting from starting address of file to last address to ensure that no other files will be allocated these memory locations.

Display the message that the file is allocated with the desired contiguous memory.

If No,

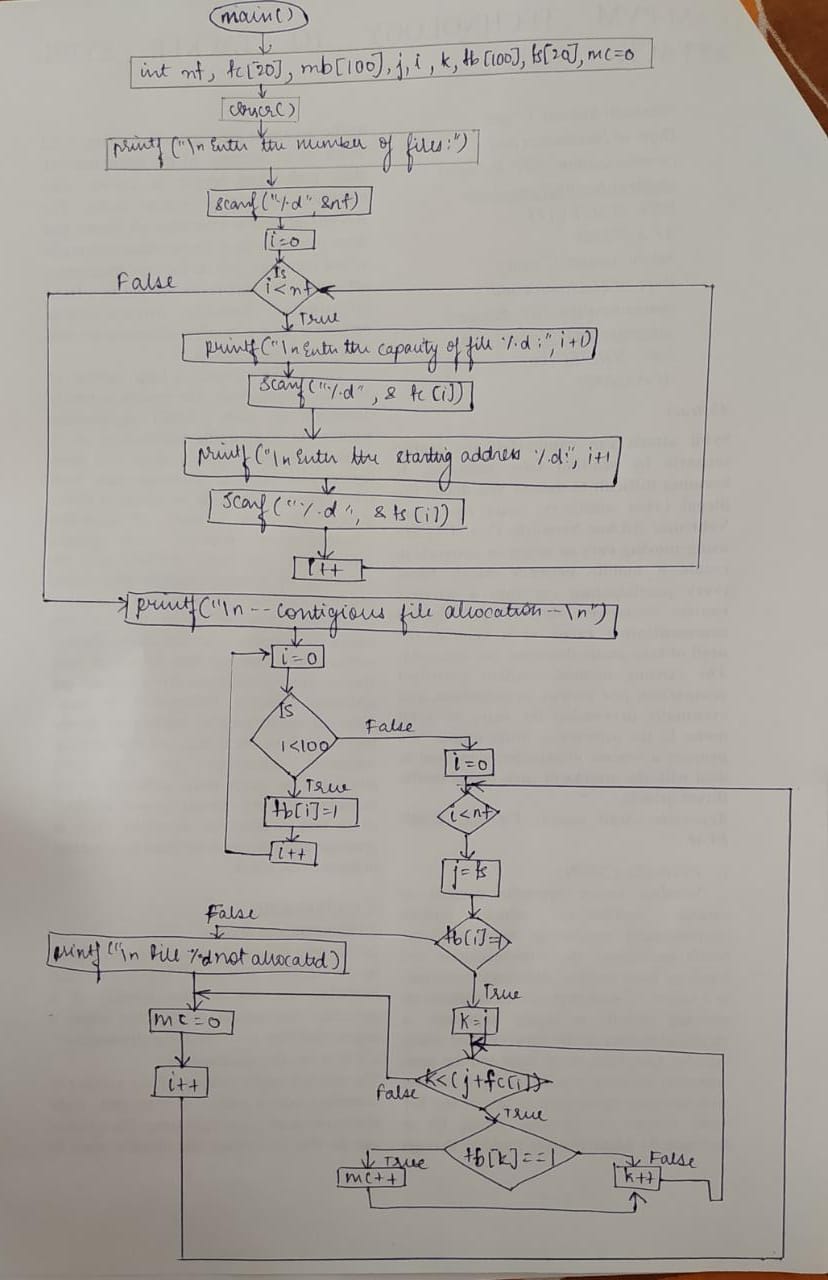
Display the message that the file is not allocated with the required memory locations

Assign value 0 to count variable to process the new file.

Repeat this step until all the files are processed.

1. Stop.

**Flow Chart:**

****

**Code:**

#include<stdio.h>

main()

{

int nf, fc[20], mb[100], i, j, k, fb[100], fs[20], mc=0;

clrscr();

printf("\nEnter the number of files: ");

scanf("%d",&nf);

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

{

printf("\nEnter the capacity of file %d: ",i+1);

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

printf("\nEnter the starting address of file %d: ",i+1);

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

}

printf("\n---CONTIGUOUS FILE ALLOCATION---\n");

for(i=0;i<100;i++)

fb[i]=1;

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

{

j=fs[i];

{

if(fb[j]==1)

{

for(k=j;k<(j+fc[i]);k++)

{

if(fb[k]==1)

mc++;

}

if(mc==fc[i])

{

for(k=fs[i];k<(fs[i]+fc[i]);k++)

{

fb[k]=0;

}

printf("\nFile %d allocated in memory %d to %d...",i+1,fs[i],fs[i]+fc[i]-1);

}

}

else

printf("\nFile %d not allocated since %d contiguous memory not available from %d...",i+1,fc[i],fs[i]);

}

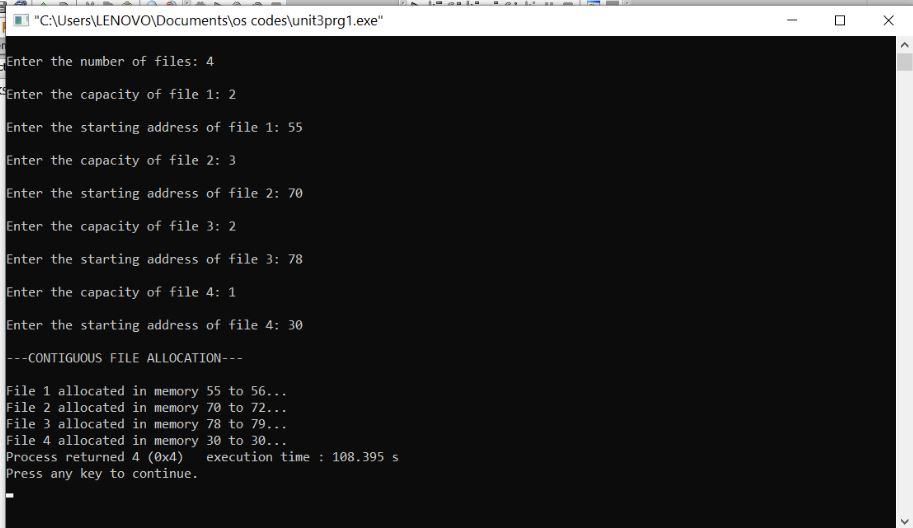
mc=0;

}

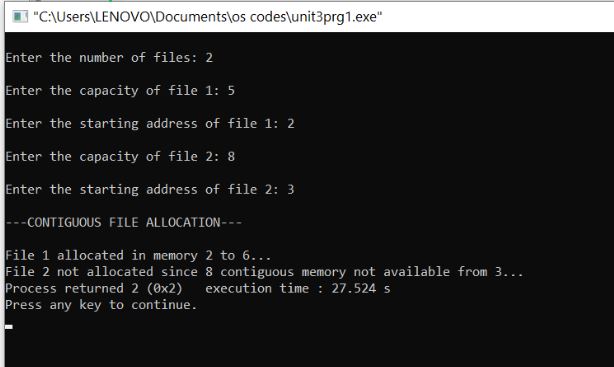
}

**Input/Output:**

**Case 1:**

****

**Case 2:**

****