**9. Simulate the following File allocation strategies (a) Sequenced (b) Indexed (c) Linked**

**(b)Linked file allocation**

**Source code:**

#include<stdio.h>

#include<conio.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++)

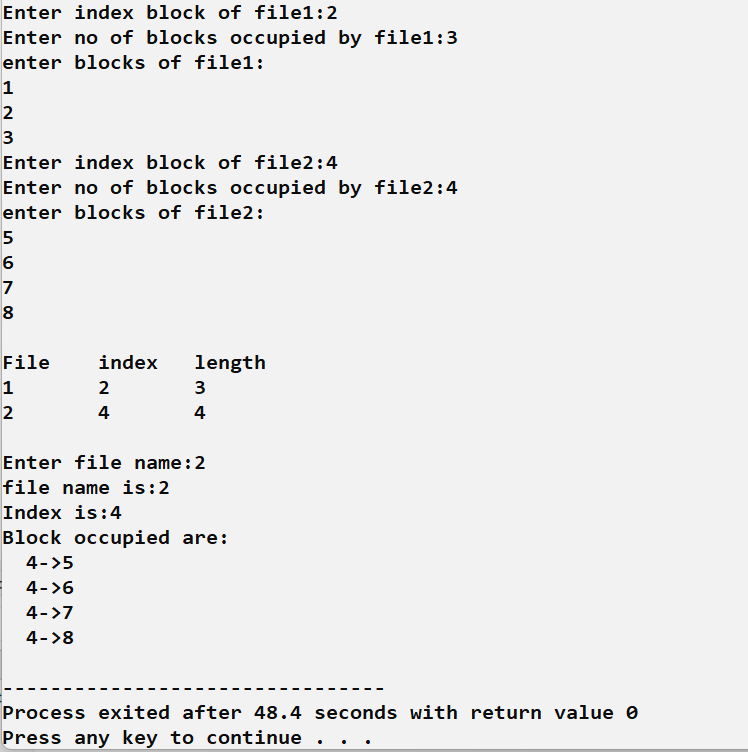
printf("%3d->%d \n",index[i],b[i][j]);

return 0;

getch();

}

**Output:**

****

**9. Simulate the following File allocation strategies (a) Sequenced (b) Indexed (c) Linked**

**(c)Linked file allocation:**

**Source code:**

#include <stdio.h>

#include<string.h>

struct file

{

char fname[10];

int start,size,block[10];

}f[10];

int main()

{

int i,j,n;

printf("Enter no. of files:");

scanf("%d",&n);

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

{

printf("Enter file name:");

scanf("%s",f[i].fname);

printf("Enter starting block:");

scanf("%d",&f[i].start);

f[i].block[0]=f[i].start;

printf("Enter no.of blocks:");

scanf("%d",&f[i].size);

printf("Enter block numbers:");

for(j=1;j<=f[i].size;j++)

{

scanf("%d",&f[i].block[j]);

}

printf("\n");

}

printf("File\tstart\tsize\tblock\n");

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

{

printf("%s\t%d\t%d\t",f[i].fname,f[i].start,f[i].size);

for(j=1;j<=f[i].size-1;j++)

printf("%d--->",f[i].block[j]);

printf("%d",f[i].block[j]);

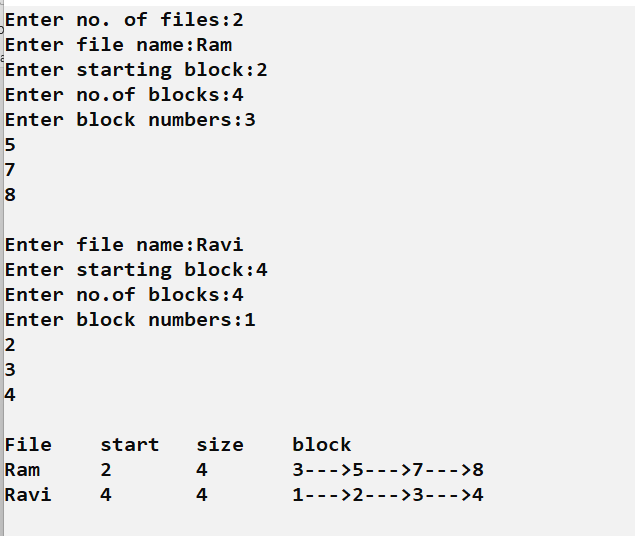
printf("\n");

}

return 0;

}

**Output:**

****