Sequential.c

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

void recurse(int files[])
{
   int flag = 0, startBlock, len, j, k, ch;
   printf("Enter the starting block and the length of the files: ");
   scanf("%d%d", &startBlock, &len);
   for (j=startBlock; j<(startBlock+len); j++)
         {
      if (files[j] == 0)
         flag++;
   }
   if(len == flag)
         {
      for (k=startBlock; k<(startBlock+len); k++)
                  {
         if (files[k] == 0)
                           {
            files[k] = 1;
            printf("%d\t%d\n", k, files[k]);
         }
      }
      if (k != (startBlock+len-1))
         printf("The file is allocated to the disk\n");
   }
   else
      printf("The file is not allocated to the disk\n");

   printf("Do you want to enter more files?\n");
   printf("Press 1 for YES, 0 for NO: ");
   scanf("%d", &ch);
   if (ch == 1)
      recurse(files);
   else
      exit(0);
   return;
}
```

```
int main()
{
int files[50],i;
for(i=0;i<50;i++)
files[i]=0;
recurse(files);
getch();
return 0;
}
```

Linked.c

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

void recursivePart(int pages[])
{
   int st, len, k, c, j;
   printf("Enter the index of the starting block and its length: ");
   scanf("%d%d", &st, &len);
   k = len;
   if (pages[st] == 0)
         {
      for (j = st; j < (st + k); j++)
                  {
         if (pages[j] == 0)
                        {
            pages[j] = 1;
            printf("%d------>%d\n", j, pages[j]);
         }
         else
                        {
            printf("The block %d is already allocated \n", j);
            k++;
         }
      }
   }
   else
      printf("The block %d is already allocated \n", st);
```

```
    printf("Do you want to enter more files? \n");
    printf("Enter 1 for Yes, Enter 0 for No: ");
    scanf("%d", &c);
    if (c==1)
        recursivePart(pages);
    else
        exit(0);
    return;
}

int main()
{
    int pages[50], p, a, i;

    for (i = 0; i < 50; i++)
        pages[i] = 0;
    printf("Enter the number of blocks already allocated: ");
    scanf("%d", &p);
    printf("Enter the blocks already allocated: ");
    for (i = 0; i < p; i++)
            {
        scanf("%d", &a);
        pages[a] = 1;
    }

    recursivePart(pages);
    getch();
    return 0;
}
```

Indexed.c

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int files[50], indexBlock[50], indBlock, n;
void recurse1();
void recurse2();
```

```
void recurse1()
{
   printf("Enter the index block: ");
   scanf("%d", &indBlock);
   if (files[indBlock] != 1)
         {
      printf("Enter the number of blocks and files needed for the index %d on the disk: ",
indBlock);
      scanf("%d", &n);
   }
   else
         {
      printf("%d is already allocated\n", indBlock);
      recurse1();
   }
   recurse2();
}

void recurse2()
{
   int ch, i, j, k;
   int flag = 0;
   for (i=0; i<n; i++)
         {
      scanf("%d", &indexBlock[i]);
      if (files[indexBlock[i]] == 0)
         flag++;
   }
   if (flag == n)
         {
      for (j=0; j<n; j++)
                  {
         files[indexBlock[j]] = 1;
      }
      printf("Allocated\n");
      printf("File Indexed\n");
      for (k=0; k<n; k++)
                  {
         printf("%d ------> %d : %d\n", indBlock, indexBlock[k], files[indexBlock[k]]);
      }
   }
```

```
 else
          {
       printf("File in the index is already allocated\n");
       printf("Enter another indexed file\n");
       recurse2();
    }
    printf("Do you want to enter more files?\n");
    printf("Enter 1 for Yes, Enter 0 for No: ");
    scanf("%d", &ch);
    if (ch == 1)
       recurse1();
    else
       exit(0);
    return;
}

int main()
{
          int i;
    for(i=0;i<50;i++)
       files[i]=0;

    recurse1();
    return 0;
}
```