```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<malloc.h>

typedef struct fat  //file allocation table
{
 char fname[30];
 int start,length,flag;
 struct fat *next;
}fat;
fat *fatfirst,*fatlast,*fattemp;

typedef struct fbl //free block
{
 int start,length;
 struct fbl *next;
}fbl;
fbl *fblfirst,*fbllast,*fbltemp;

void displayfat()
{
 fat* p;
 printf("\nDisk List:");
 printf("\nName\tStart\tLength");

 p=fatfirst;
 while(p!=NULL)
     {
      if(p->flag==0)
            printf("\n%s\t%d\t%d",p->fname,p->start,p->length);
      p=p->next;
     }
}

void displayfbl()
{
 fbl *p;
 printf("\n\nFree Block-space List:");
 printf("\nStart\tLength");

 p=fblfirst;
 while(p!=NULL)
      {
       printf("\n%d\t%d",p->start,p->length);
       p=p->next;
      }
}

void appendfat()
{
 if(fatfirst==NULL)
   {
    fatfirst=fattemp;
    fatlast=fattemp;
   }
 else
```

```
    {
     fatlast->next=fattemp;
     fatlast=fatlast->next;
    }
}

void appendfbl()
{
 if(fblfirst==NULL)
    {
     fblfirst=fbltemp;
     fbllast=fbltemp;
    }
 else
    {
     fbllast->next=fbltemp;
     fbllast=fbllast->next;
    }
}

void main()
{
 int ch,bno,f,blocks;
 char fn[30];

 fbl* fblptr;
 fat* fatptr;

 printf("\nEnter maximum number of blocks to allocate: ");
 scanf("%d",&blocks);
 fbltemp=(fbl*)malloc(sizeof(fbl));
 fbltemp->start=0;
 fbltemp->length=blocks;
 fbltemp->next=NULL;
 appendfbl();

  do
    {
     f=1;
     printf("\n\n\n\nMENU:\n1.INSERTION\n2.DELETION\n3.DISPLAY\n4.EXIT\nEnter your choice: "
     );
     scanf("%d",&ch);
     switch(ch)
        {
        case 1: printf("\n\nEnter the file name:");
                scanf("%s",fn);
                printf("Enter the file size in number of blocks:");
                scanf("%d",&bno);
                for(fblptr=fblfirst;fblptr!=NULL;fblptr=fblptr->next)
                   {
                    if(fblptr->length>=bno)
                      break;
                   }
                if(fblptr==NULL)
                    printf("\nNot enough space...");
                else
                   {
```

```
                    fattemp=(fat*)malloc(sizeof(fat));
                    strcpy(fattemp->fname,fn);
                    fattemp->length=bno;
                    fattemp->start=fblptr->start;
                    fattemp->flag=0;
                    fattemp->next=NULL;

                    appendfat();

                    fblptr->start+=bno;
                    fblptr->length-=bno;
                    printf("\nFile is placed in free space...");
                   }
                break;

        case 2: printf("\n\nEnter the file name to be deleted:");
                scanf("%s",fn);
                for(fatptr=fatfirst;fatptr!=NULL;fatptr=fatptr->next)
                   {
                    if(strcmp(fatptr->fname,fn)==0 && fatptr->flag==0)
                      {
                       f=0;
                       break;
                      }
                   }
                if(f==1)
                   printf("\nThe given file doesn't exist...");
                else
                   {
                    fatptr->flag=1;

                    fbltemp=(fbl*)malloc(sizeof(fbl));
                    fbltemp->start=fatptr->start;
                    fbltemp->length=fatptr->length;
                    fbltemp->next=NULL;

                    appendfbl();

                    printf("\nThe given file is deleted from the disk...");
                   }
                break;

        case 3: displayfat();
                displayfbl();
                break;
        }
    }while(ch!=4);
}
```