

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 1

Title: File Handling Operations

```
#include<stdio.h>
#include<stdlib.h>
int main()
{ int ch1;
  char msg[1000],ch;
  char
old_name[20],new_name[20],source_file[20],target_file[20];
  FILE *fptr,*source,*target; printf("\n1. Create
  File and Write data\n2.Read the
data\n3.Rename File\n4.Copy Data of File to another \n Enter
The Choice:");
  scanf("%d",&ch1);
  //do{
switch(ch1) {
  case 1:
    fptr =fopen("program2.txt","w"); if(fptr
    ==NULL)
    {
      printf("Error!")
      ; exit(1);
    }//if close
    printf("Enter message:");
    scanf("%s",msg);
    fprintf(fptr,"%s",msg);
    break;
  case 2:
    if((fptr = fopen("program2.txt","r")) == NULL)
      { printf("Error! opening file");
        exit(1);
      }//if close
    fscanf(fptr,"%s",msg);
    printf("Messege Is:=%s",msg);
    break;

  case 3:
    printf("\n Enter old progam2.txt");
    scanf("%s",old_name);
    printf("\n Enter new prog2.txt");
    scanf("%s",new_name);
    if(rename(old_name,new_name) ==0)
      { printf("File renamed successfully.\n");
        }//if close
    else
      { printf("unable to rename files.Please check files
        exist and you have permission to modify files.\n");
```

```

        }
        break;
case 4:
    printf("Enter name of file to copy\n");
    scanf("%s",source_file);
    source = fopen(source_file,"r");
    if(source==NULL)
        { printf("Press any key to exit...\n");
          exit(EXIT_FAILURE);
        } //if close
    printf("Enter name of target file\n");
    scanf("%s",target_file); target =
    fopen(target_file,"w");

    if(target == NULL)
        { printf("Press any key to exit...\n");
          exit(EXIT_FAILURE);
        } //if close
    while((ch=fgetc(source))!=EOF)
        fputc(ch,target);
    printf("File copied successfully.\n");
    break;
} //switch close
//}while(ch !=5); //do close
//fclose(source);
//fclose(target);
//fclose(fp);
return 0;
} //main close

```

Output :-

```

1. Create File and Write data
2. Read the data
3. Rename File
4. Copy Data of File to another
Enter The Choice:1
Enter message: Hello

```

```

1. Create File and Write data
2. Read the data
3. Rename File
4. Copy Data of File to another
Enter The Choice:2
Messege Is:=Hello

```

```

1. Create File and Write data
2. Read the data
3. Rename File
4. Copy Data of File to another Enter The Choice:3

```

Enter old program2.txt program2.txt
Enter new prog2.txt
NewProgram.txt
File renamed successfully.

1. Create File and Write data
 2. Read the data
 3. Rename File
 4. Copy Data of File to another
- Enter The Choice:4 Enter
name of file to copy
program2.txt Enter name
of target file
NewProgram.txt
File copied successfully.

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 2

Title: FCFS CPU Scheduling Algorithm

```
#include<stdio.h> #define MAX 100 int main(){
    intArrival_time[MAX], Burst_time[MAX], Completion_time[MAX],
    Turn_Around_time[MAX], Waiting_time[MAX],
    Average_Turn_Around_time = 0, Average_Waiting_time = 0, i, j;
    printf("Enter Process U Want: "); scanf("%d",&j);
    printf("Enter Arrival Time: ");
    for(i=0;i<j;i++){
        scanf("%d",&Arrival_time[i]);
    }
    printf("Enter Burst Time: ");
    for(i=0;i<j;i++){
        scanf("%d",&Burst_time[i]);
    }

    Completion_time[0] = Burst_time[0];
    for(i=1;i<j;i++){
        Completion_time[i] = Completion_time[i-1] + Burst_time[i];
    }
    for(i=0;i<j;i++){
        Turn_Around_time[i] = Completion_time[i] - Arrival_time[i];
        Waiting_time[i] = Turn_Around_time[i] - Burst_time[i];
        Average_Waiting_time += Waiting_time[i];
        Average_Turn_Around_time += Turn_Around_time[i];
    }
    printf("\nProcess Arrival(T) Burst(T) Completion(T) Turn-
    Around(T) Waiting(T)"); for(i=0;i<j;i++){
        printf("\nP[%d]\t %d\t \t%d\t\t %d\t \t%d\t\t \t%d", i+1
        ,Arrival_time[i], Burst_time[i], Completion_time[i],
        Turn_Around_time[i], Waiting_time[i]);
    } printf("\n\nAverage Turn Around
    Time: %d", Average_Turn_Around_time/j);
    printf("\nAverage Waiting Time: %d\n",
    Average_Waiting_time/j);
    return 0;
}
```

Output :-

Enter Process U Want: 4
Enter Arrival Time: 10 9
12
19 4
28
Enter Burst Time: 20

12 34

5

Process	Arrival(T)	Burst(T)	Completion(T)	Turn-Around(T)	Waiting(T)
P[1]	19	20	20	1	-19
P[2]	12	12	32	20	8
P[3]	14	34	66	52	18
P[4]	28	5	71	43	38

Average Turn Around Time: 29

Average Waiting Time: 11

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 3

Title: Producer-Consumer problem using semaphores

```
#include<stdio.h>
#include<conio.h>
void main()
{ int buffer[10], bufsize, in, out, produce, consume,
choice=0; in = 0; out
  = 0; bufsize = 10;
  while(choice !=3)
    { printf("\n 1. Produce \t 2. Consume \t 3. Exit");
      printf("\n Enter your choice: "); scanf("%d",
        &choice);
    switch(choice) {
      case 1: if((in+1)%bufsize==out)
        printf("\nBuffer is Full");
        else
        {
          printf("\nEnter the value:");
          scanf("%d", &produce);
          buffer[in] = produce; in =
            (in+1)%bufsize;
        }
        break;
      case 2: if(in == out)
        printf("\nBuffer is Empty");
        else
        {
          consume = buffer[out];
          printf("\nThe consumed value is %d", consume); out
            = (out+1)%bufsize;
        }
        break;
    }
  }
```

Output:-

1. Produce 2. Consume 3. Exit
Enter your choice: 1
Enter the value: 10

1. Produce 2. Consume 3. Exit
Enter your choice: 2
The consumed value is 10

1. Produce 2. Consume 3. Exit

Enter your choice: 1

Enter the value: 30

1. Produce 2. Consume 3. Exit

Enter your choice: 2

The consumed value is 30

1. Produce 2. Consume 3. Exit

Enter your choice: 3

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 4

Title: Memory Allocation Technique

```
#include<stdio.h>
#include<conio.h>
#define max 25 void
main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp; static int
bf[max],ff[max]; printf("\n\tMemory Management Scheme -
First Fit"); printf("\nEnter the number of blocks:");
scanf("%d",&nb); printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files :-\n"); for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{ if(bf[j]!=1)
{ temp=b[j]-f[i]; if(temp>=0)
{ ff[i]=j;
break;
} } } frag[i]=temp;
bf[ff[i]]=1;}
printf("\nFile_no:\tFile_size
:\tBlock_no:\tBlock_size:\tFragement"); for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],f
rag[i]); getch();
}
```

Output:

Memory Management Scheme - First Fit Enter the number of blocks:5

Enter the number of files:3

Enter the size of the blocks:- Block

1:2

Block 2:1

Block 3:1

Block 4:1

Block 5:2

Enter the size of the files :-

File 1:2

File 2:1

File 3:3

File_no:	File_size :	Block_no:	Block_size:	Fragement
1	2	1	2	0
2	1	2	1	0
3	3	0	1	-1

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38 Practical 5

Title: Sequential File Allocation Strategies

```
#include<stdio.h>
#include<conio.h>
struct fileTable
{ char
name[20]; int
sb, nob;
}ft[30]; void
main() { int
i, j, n; char
s[20];
printf("Enter no of files :");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("\nEnter file name %d :",i+1); scanf("%s",ft[i].name);
printf("Enter starting block of file %d :",i+1);
scanf("%d",&ft[i].sb); printf("Enter no of blocks in file %d
:",i+1); scanf("%d",&ft[i].nob);
}
printf("\nEnter the file name to be searched-- ");
scanf("%s",s); for(i=0;i<n;i++) if(strcmp(s,
ft[i].name)==0) break; if(i==n) printf("\nFile Not
Found"); else
{ printf("\nFILE NAME START BLOCK NO OF BLOCKS
BLOCKS OCCUPIED\n");
printf("\n%s\t\t%d\t\t%d\t\t",ft[i].name,ft[i].sb,ft[i].nob)
; for(j=0;j<ft[i].nob;j++) printf("%d, ",ft[i].sb+j);
}
getch();
}
```

Output :-

INPUT: Enter no of files :3

Enter file name 1 :A

Enter starting block of file 1 :85 Enter no of blocks in file 1 :6

Enter file name 2 :B

Enter starting block of file 2 :102

Enter no of blocks in file 2 :4

Enter file name 3 :C

Enter starting block of file 3 :60 Enter no of blocks in file 3 :4

Enter the file name to be searched -- B OUTPUT:

FILE NAME START BLOCK NO OF BLOCKS BLOCKS OCCUPIED

B 102 4 102, 103, 104, 105

Name: Unnati Rajendra Satav

Class: TE

Div: C

Roll No: 38

Practical 6

Title: Single level directory file organization technique

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
struct
{
char dname[10],fname[10][10]; int fcnt;
}dir; void main() { int i,ch; char f[30]; dir.fcnt = 0;
printf("\nEnter name of directory -- "); scanf("%s",
dir.dname); while(1) { printf("\n\n1. Create File\t2. Delete
File\t3. Search File \n4. Display Files\t5. Exit\nEnter your
choice -- "); scanf("%d",&ch); switch(ch) {
case 1: printf("\nEnter the name of the file -- ");
scanf("%s",dir.fname[dir.fcnt]); dir.fcnt++; break;
case 2: printf("\nEnter the name of the file -- ");
scanf("%s",f); for(i=0;i<dir.fcnt;i++) {
if(strcmp(f, dir.fname[i])==0)
{
printf("File %s is deleted ",f);
strcpy(dir.fname[i],dir.fname[dir.fcnt-1]); break;
}} if(i==dir.fcnt)
printf("File %s not found",f);
else dir.fcnt--; break;
case 3: printf("\nEnter the name of the file -- ");
scanf("%s",f); for(i=0;i<dir.fcnt;i++)
{
if(strcmp(f, dir.fname[i])==0)
{ printf("File %s is found ",
f); break;
} } if(i==dir.fcnt) printf("File
%s not found",f); break; case 4:
if(dir.fcnt==0)
printf("\nDirectory Empty"); else
{ printf("\nThe Files are --
"); for(i=0;i<dir.fcnt;i++)
printf("\t%s",dir.fname[i]); }
break; default: exit(0);
} }
getch()
;
}
```

Output:

Enter name of directory -- CSE

1. Create File 2. Delete File 3. Search File
4. Display Files 5.
Exit Enter your choice 1
Enter the name of the file -- A
1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 1
Enter the name of the file -- B
1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 1
Enter the name of the file -- C
1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 4
The Files are -- A B C 1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 3
Enter the name of the file ABC File ABC not found
1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 2
Enter the name of the file B
File B is deleted
1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit
Enter your choice 5 ...Program finished with exit code 0

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 7

Title: FIFO page replacement algorithm

```
#include<stdio.h>
#include<conio.h>
main()
{ int i, j, k, f, pf=0, count=0, rs[25], m[10], n;
printf("\n Enter the length of reference string -- ");
scanf("%d",&n); printf("\n Enter the reference string
-- "); for(i=0;i<n;i++) scanf("%d",&rs[i]);
printf("\n Enter no. of frames -- ");
scanf("%d",&f); for(i=0;i<f;i++)
m[i]=-1;
printf("\n The Page Replacement Process is -- \n");
for(i=0;i<n;i++)
{
for(k=0;k<f;k++)
{
if(m[k]==rs[i]
) break; }
if(k==f) {
m[count++]=rs[i]; pf++; }
for(j=0;j<f;j++)
printf("\t%d",m[j]); if(k==f)
printf("\tPF No. %d",pf); printf("\n"); if(count==f)
count=0; } printf("\n The number of Page Faults using FIFO
are %d",pf); getch();
}
```

Output:

Enter the length of reference string -- 5 Enter the reference string -- 1 2 3 4 5 Enter no. of frames -- 4 The Page Replacement Process is --
1 -1 -1 -1 PF No. 1 1 2 -1 -1 PF No. 2 1 2 3 -1 PF No. 3 1 2 3 4 PF No. 4 5 2 3 4 PF No. 5 The number of Page Faults using FIFO are 5

Name: Unnati Rajendra Satav

Class: TE Div: C Roll No: 38

Practical 8

Title: FCFS disk scheduling algorithm

```
#include<stdio.h>
> main() {
int t[20], n, I, j, tohm[20], tot=0;
float avhm; clrscr();
printf("enter the no.of tracks"); scanf("%d",&n);
printf("enter the tracks to be traversed");
for(i=2;i<n+2;i++)
scanf("%d",&t*i+); for(i=1;i<n+1;i++)
{
tohm[i]=t[i+1]-t[i]; if(tohm[i]<0) tohm[i]=tohm[i]*(-1); }
for(i=1;i<n+1;i++) tot+=tohm[i]; avhm=(float)tot/n;
printf("Tracks traversed\tDifference between tracks\n");
for(i=1;i<n+1;i++)
printf("%d\t\t\t\t\t%d\n",t*i+,tohm*i+); printf("\nAverage header
movements:%f",avhm); getch(); }
```

Output :-

INPUT

Enter no.of tracks:9

Enter track position:55 58 60 70 18 90 150 160 184 OUTPUT

Tracks traversed Difference between tracks

55
58
60
70
18
90
150
160
184
45
3
2
10
52
72
60
10
24

Average header movements: 30.888889