Ex.No 11

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

void fun(int);

struct emp
{
int no,sal,age;
}e[10];
FILE *p;
void main()
{
int i,n;
clrscr();
p=fopen("a.txt","w");
if(p==NULL)
printf("\n cant open the file");
printf("\n no of employees:");
scanf("%d",&n);
printf("\n enter the employee no salary age");
for(i=0;i<n;i++)
{
fscanf(stdin,"%d %d
%d",&e[i].no,&e[i].sal,&e[i].age);
fprintf(p,"%d %d %d \n",e[i].no,e[i].sal,e[i].age);
}
fclose(p);
fun(n);
getch();
}

void fun(int n)
{
int i,x,y,w;
printf("\nlist of people age > 35 & salary < 4500
:");
p=fopen("a.txt","r");
for(i=0;i<n;i++)
```

```
{
fscanf(p,"%d %d %d",&x,&y,&w);
if(y<4500&&w>35)
{
printf("\n employee id is %d",x);
}
}}
```

Ex.No 11 (Prelab)

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
FILE *mfile,*odd,*even;
int n,a[100],i;

clrscr();

printf("no of elements\n");
scanf("%d",&n);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
mfile=fopen("mainfile.txt","w");
for(i=0;i<n;i++)
fprintf(mfile,"\n%d\n",a[i]);
fclose(mfile);
mfile=fopen("mainfile.txt","r");
odd=fopen("oddfile.txt","w");
even=fopen("evenfile.txt","w");
while(mfile!=NULL)
{fscanf(mfile,"%d",a[i]);
if(a[i]%2==0)
fprintf(even,"\n%d\n",a[i]);
else
fprintf(odd,"\n%d\n",a[i]);
}
fclose(mfile);
```

```
fclose(odd);
fclose(even);
getch();
}
```

Ex.No.9 (Matrix Multiplication using dynamic memory allocation)

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

/* Main Function */
int main()
{

/* Declaring pointer fo matrix multiplication.*/
int **ptr1, **ptr2, **ptr3;

/* Declaring integer variables for row and columns
of two matrices.*/
int row1, col1, row2, col2;

/* Declaring indexes. */
int i, j, k;

/* Request the user to input number of columns of
the matrices.*/
printf("\nEnter number of rows for first matrix :
");
scanf("%d", &row1);
printf("\nEnter number of columns for first matrix
: ");
scanf("%d", &col1);

printf("\nEnter number of rows for second matrix :
");
scanf("%d", &row2);
```

```
printf("\nEnter number of columns for second matrix
: ");
scanf("%d", &col2);

if(col1 != row2)
{
printf("\nCannot multiply two matrices.");
return(0);
}

/* Allocating memory for three matrix rows. */
ptr1 = (int **) malloc(sizeof(int *) * row1);
ptr2 = (int **) malloc(sizeof(int *) * row2);
ptr3 = (int **) malloc(sizeof(int *) * row1);

/* Allocating memeory for the cloumns of three
matrices. */
for(i=0; i<row1; i++)
{
ptr1[i] = (int *)malloc(sizeof(int) * col1);
}
for(i=0; i<row2; i++)
{
ptr2[i] = (int *)malloc(sizeof(int) * col2);
}

for(i=0; i<row1; i++)
{
ptr3[i] = (int *)malloc(sizeof(int) * col2);
}


/* Request the user to input members of first
matrix. */
printf("\nEnter elements of first matrix :\n");
for(i=0; i< row1; i++)
{
for(j=0; j< col1; j++)
{
printf("\tA[%d][%d] = ",i, j);
```

```
scanf("%d", &ptr1[i][j]);
}
}

/* request to user to input mebmbers of first
matrix. */
printf("\nEnter elements of second matrix :\n");
for(i=0; i< row2; i++)
{
for(j=0; j< col2; j++)
{
printf("\tB[%d][%d] = ",i, j);
scanf("%d", &ptr2[i][j]);
}
}

/* Calculation begins for the resultant matrix. */
for(i=0; i < row1; i++)
{
for(j=0; j < col1; j++)
{
ptr3[i][j] = 0;
for(k=0; k<col2; k++)
{
ptr3[i][j] = ptr3[i][j] + ptr1[i][k] * ptr2[k][j];
}
}
}

/* Printing the contents of first matrix. */
printf("\n\nFirst matrix :");
for(i=0; i< row1; i++)
{
printf("\n\t\t\t");
for(j=0; j< col1; j++)
{
printf("%4d", ptr1[i][j]);
}
}
/* Printing the contents of second matrix. */
```

```
printf("\n\nSecond matrix :");
for(i=0; i< row2; i++)
{
printf("\n\t\t\t");
for(j=0; j < col2; j++)
{
printf("%4d", ptr2[i][j]);
}
}


/* Printing the contents of third matrix. */

printf("\n\nResultant matrix :");
for(i=0; i< row1; i++)
{
printf("\n\t\t\t");
for(j=0; j < col2; j++)
{
printf("%4d", ptr3[i][j]);
}
}
return(0);
}
```