```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#define n1 13
char p[20];
float a4,b4,c4,d4,r1,r2;
int d3,on,n,rn;
int a3,b3,n,i,c3=0,p3;
int dd,mm,yy,valid=1;
int g1[10][10],h1[10][10],f1[10][10],i,j,m,n,p2,q2,k;
int pos,k,m,t,lar;
int a3,b3,c3,d3,g3;
int n,num,d1,sum,b3;
int len,a1,b2,r,s1[n1],i,top=-1,y,fact=1;
int i,b1,n,ch,w,f=1,w2;
int q[n1];
char stack[25];
char infix[25],postfix[25];
float w1;
int f,r,f2[10];
char *q1,ipstr[17];
int p1,i;
char a[50],b[20];
int j,e;
char fname[15];
FILE *fp;

struct student
{
char name[15];
int usn,m1,m2,m3;
}s;

void insert()
{
fp=fopen(fname,"w");

for(i=0;i<n;i++)
{
printf("enter the records of the student %d\n",i+1);
printf("enter the name\t");
scanf("%s",s.name);
printf("enter the usn\t");
scanf("%d",&s.usn);
printf("enter the m1\t");
scanf("%d",&s.m1);
printf("enter the m2\t");
scanf("%d",&s.m2);
printf("enter the m3");
scanf("%d",&s.m3);
fprintf(fp,"%s\t%d\t%d\t%d\t%d",s.name,s.usn,s.m1,s.m2,s.m3);
}
fclose(fp);
}
```

```
void display()
{
if(n==0)printf("no records to display\n");
else
{
fp=fopen(fname,"r");
if(fp==NULL);
printf("error in opening the file\n");
{
for(i=0;i<n;i++)
{
printf("details of the student %d\n",i+1);
fscanf(fp,"%s",s.name);
printf("name \t %s\n",s.name);
fscanf(fp,"%d%d%d%d",&s.usn,&s.m1,&s.m2,&s.m3);

printf("usn \t %d\n",s.usn);
printf("m1\t %d\n",s.m1);
printf("m2 \t %d\n",s.m2);
printf("m3\t %d\n",s.m3);
}
}
fclose(fp);
}
 }

void search()
{
int usn,f=0;

if(n==0)printf("no records to search\n");
else
{
fp=fopen(fname,"r");
if(fp==NULL)printf("error in opening the file\n");
else
{
printf("enter the usn of the student\n");
scanf("%d",&usn);
for(i=0;i<n;i++)
{
fscanf(fp,"%s%d%d%d%d",s.name,&s.usn,&s.m1,&s.m2,&s.m3);
if(usn==s.usn)
{
f=1;
break;
}
}
}
}
if(f==1)
{
 printf("name \t %s\n",s.name);
printf("usn \t %d\n",s.usn);
printf("m1\t %d\n",s.m1);
printf("m2 \t %d",s.m2);
printf("m3\t %d\n",s.m3);
```

```
}
  fclose(fp);
 }

 void c()
{
int i=0,n;
printf("enter the first string\n");
scanf("%s",a);
n=strlen(a);
for(i=0;i<n;i++)
{
b[i]=a[i];
}
a[i]='\0';
printf("the copied string is %s\n",b);
}

void d()
{
int n,m;
printf("enter the first string\n");
scanf("%s",a);
printf("enter the second string\n");
scanf("%s",b);
n=strlen(a);
m=strlen(b);

for(i=0;i<m;i++)
a[m+i]=b[i];
a[m+n]='/0';
printf("combined string is %s\n",a);
}

void push(int a1)
{
s1[++top]=a1;
}
int pop()
{
return(s1[top--]);
}

void rem()
{
if(r<f)
printf("under flow\n");
else
printf("the element removed is %d\n",q[f++]);
}

void insert1()
{
if(r==n1-1)
printf("over flow\n");
else
{
```

```
printf("enter the element to insert\n");
scanf("%d",&q[++r]);
}
}

void disp()
{
if(r<f)
printf("empty\n");
else
{
int i;
printf("the elements in the queue are\n");
for(i=0;i<=r;i++)
printf("%d\n",q[i]);
printf("total number of elements in the queue is %d\n",i);
}
}

union
{
unsigned long int ip;
unsigned char a[4];
}x;

void push2()
{
if(top==n1-1)
printf("\nover flow\n");
else
{
printf("\nenter the element\n");
scanf("%d",&s1[++top]);
}
}

void pop1()
{
if(top==-1)
printf("\nunder flow\n");
else
printf("\nthe popped element is %d",s1[top--]);
}

void display3()
{
if(top==-1)
printf("\nstack is empty\n");
else
{
printf("\nthe steck elememts are\n");
for(i=top;i>-1;i--)
printf("%d\n",s1[i]);
printf("\nnumber of elements are %d\n",top+1);
 }
 }
```

```
void convert()
{
int i=0,j=0;
while(infix[i]!=0)
{
char ch=infix[i++];
if(isalpha(ch))
postfix[j++]=ch;
else
{
while(top!=-1&&precedence(stack[top],ch))
postfix[j++]=stack[top--];
if(top==-1||ch!=')')
stack[++top]=ch;
}
}
 while(top!=-1)
{
 char c;
 c=stack[top--];
 if(c!='(')
 postfix[j++]=c;
 }
 postfix[j]='\0';
}

int precedence(char top, int cur)
{
if(top!='('&&cur==')')
return 1;
else if((top=='$'||top=='*'||top=='/')&&(cur=='*'||cur=='+'||cur=='-'))
return 1;
else if((top=='+'||top=='-')&&(cur=='+'||cur=='-' ))
return 1;
else
return 0;
}

int sea(int l,int h)
{
if(l>h)
return -1;
else
{

m=(l+h)/2;
if(k>s1[m])
return(m+1,n-1);
else
return(l,m-1);
}
}

void tower(int n,char f, char a,char t)
{
if(n==1)
printf("disk is moved from %c to %c\n",f,t);
```

```
else
{
tower(n-1,f,t,a);
printf("disk is moved from %c to %c\n",f,t);
tower(n-1,a,f,t);
}
}

void sort()
{

      for(i=1;i<n;i++)
      {
      for(j=0;j<n-i;j++)
      {
      if(s1[j]>s1[j+1])
      {
      t=s1[j];
      s1[j]=s1[j+1];
      s1[j+1]=t;
      }
      }
      }
      printf("the sorted array is \n");
      for(i=0;i<n;i++)
      printf("%d\n",s1[i]);
}


void read(int g1[10][10],int m,int n)
{
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&g1[i][j]);
}

void write(int g1[10][10],int m,int n)
{
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf("%d\t",g1[i][j]);
}
printf("\n");
}
}

void mul()
{
for(i=0;i<m;i++)
{
for(j=0;j<q2;j++)
{
f1[i][j]=0;
for(k=0;k<n;k++)
{
```

```
f1[i][j]+=g1[i][k]*h1[k][j];
}
}
}
}

int gcd(int a3,int b3)
{
while(a3!=b3)
{
if(a3>b3)
a3-=b3;
else
b3-=a3;
}
return b3;
}


void main()
{
int w;
clrscr();
man:printf("\n1:lab programs\n 2:sorting programms\n3:searching
programs\n4:matrix programs\n5:conversion\n6:applications\n 7:exit\n");
scanf("%d",&w);
switch(w)
{

case 1:     man1: printf("1:files programs\n2: string
programs\n3:evaluate the valid postfix expression");
      printf("\n4:queue program\n5:ip address program\n6:stack
program\n");
      printf("7:conversion\n8:binary search using recursion 9: towers of
hanaio\n");
      printf("10:to main menu\n11:exit\n");
      scanf("%d",&w);

      switch(w)
      {
            case 1:
                  printf("enter the file name\n");
                  scanf("%s",fname);
                  printf("enter the number of records \n");
                  scanf("%d",&n);
          while(1)
          {
          printf("1:insert record\t 2: display\t 3:search\t 4: for sub
menu\n");
          scanf("%d",&b1);

          switch(b1)
          {
          case 1: insert();
                  break;
          case 2: display();
                  break;
```

```
            case 3: search();
                    break;
            case 4:
                    goto man1;
            default : printf("enter the valid input\n");
            }
            }

case 2:
       while(1)
        {
       printf("1:copy\t2:catinate\t3:to sub menuu\n");
       scanf("%d",&e);

       switch(e)
       {
       case 1:c();
       break;
       case 2:d();
       break;
       case 3: goto man1;
       default : printf("enter the valid input\n");
       }
       }

case 3:
       printf("enter the string\n");
       scanf("%s",p);
       len=strlen(p);
       for(i=0;i<len;i++)

       switch(p[i])
       {
       case '+' : b2=pop();
             a1=pop();
             push(a1+b2);
             break;
       case '-' :b2=pop();
             a1=pop();
             push(a1-b2);
             break;
       case '*' :b2=pop();
             a1=pop();
             push(a1*b2);
             break;
       case '/' :b2=pop();
             a1=pop();
             push(a1/b2);
             break;
       case '^' :b2=pop();
             a1=pop();
             push(pow(a1,b2));
             break;
       default:push(p[i]-'0');
       }
       printf("%d\n",pop());
       goto man1;
```

```
case 4:
     r=-1;
     f=0;
     while(1)
     {
     printf("1:insert 2:remove 3:display 4: exit\n");
     scanf("%d",&ch);

     switch(ch)

     {
           case 1: insert1();
                 break;
           case 2: rem();
                 break;
           case 3: disp();
                 break;
           case 4: goto man1;

           default : printf("invalid\n");
     }
     }

case 5:
     printf("enter the ip address\n");
     scanf("%s",ipstr);
     q1=strtok(ipstr,".");
     p1=atoi(q1);
     x.a[3]=p1;
     for(i=2;i>=0;i--)
     {
     q1=strtok(NULL,".");
     p1=atoi(q1);
     x.a[i]=p1;
     }
     printf("long integer is %lu\n",x.ip);
     goto man1;

case 6:while(1)
     {
     printf("\n1:push\n 2:pop\n3:display\n4:to the sum menu\n");
     scanf("%d",&w);

     switch(w)
     {
     case 1:push2();
           break;
     case 2: pop1();
           break;
     case 3: display3();
           break;
     case 4:goto man1;
     default:printf("enter the valid inout\n");
     }
     }
```

```
case 7:     printf("enter the valid infix expression\n");
      scanf("%s",infix);
.......
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#define n1 13
char p[20];
float a4,b4,c4,d4,r1,r2;
int d3,on,n,rn;
int a3,b3,n,i,c3=0,p3;
int dd,mm,yy,valid=1;
int g1[10][10],h1[10][10],f1[10][10],i,j,m,n,p2,q2,k;
int pos,k,m,t,lar;
int a3,b3,c3,d3,g3;
int n,num,d1,sum,b3;
int len,a1,b2,r,s1[n1],i,top=-1,y,fact=1;
int i,b1,n,ch,w,f=1,w2;
int q[n1];
char stack[25];
char infix[25],postfix[25];
float w1;
int f,r,f2[10];
char *q1,ipstr[17];
int p1,i;
char a[50],b[20];
int j,e;
char fname[15];
FILE *fp;

struct student
{
char name[15];
int usn,m1,m2,m3;
}s;

void insert()
{
fp=fopen(fname,"w");

for(i=0;i<n;i++)
{
printf("enter the records of the student %d\n",i+1);
printf("enter the name\t");
scanf("%s",s.name);
printf("enter the usn\t");
scanf("%d",&s.usn);
printf("enter the m1\t");
scanf("%d",&s.m1);
printf("enter the m2\t");
scanf("%d",&s.m2);
printf("enter the m3");
scanf("%d",&s.m3);
fprintf(fp,"%s\t%d\t%d\t%d\t%d",s.name,s.usn,s.m1,s.m2,s.m3);
```

```
}
fclose(fp);
}
void display()
{
if(n==0)printf("no records to display\n");
else
{
fp=fopen(fname,"r");
if(fp==NULL);
printf("error in opening the file\n");
{
for(i=0;i<n;i++)
{
printf("details of the student %d\n",i+1);
fscanf(fp,"%s",s.name);
printf("name \t %s\n",s.name);
fscanf(fp,"%d%d%d%d",&s.usn,&s.m1,&s.m2,&s.m3);

printf("usn \t %d\n",s.usn);
printf("m1\t %d\n",s.m1);
printf("m2 \t %d\n",s.m2);
printf("m3\t %d\n",s.m3);
}
}
fclose(fp);
}
 }

void search()
{
int usn,f=0;

if(n==0)printf("no records to search\n");
else
{
fp=fopen(fname,"r");
if(fp==NULL)printf("error in opening the file\n");
else
{
printf("enter the usn of the student\n");
scanf("%d",&usn);
for(i=0;i<n;i++)
{
fscanf(fp,"%s%d%d%d%d",s.name,&s.usn,&s.m1,&s.m2,&s.m3);
if(usn==s.usn)
{
f=1;
break;
}
}
}
}
if(f==1)
{
 printf("name \t %s\n",s.name);
printf("usn \t %d\n",s.usn);
```

```
printf("m1\t %d\n",s.m1);
printf("m2 \t %d",s.m2);
printf("m3\t %d\n",s.m3); }
  fclose(fp);
 }

 void c()
{
int i=0,n;
printf("enter the first string\n");
scanf("%s",a);
n=strlen(a);
for(i=0;i<n;i++)
{
b[i]=a[i];
}
a[i]='\0';
printf("the copied string is %s\n",b);
}

void d()
{
int n,m;
printf("enter the first string\n");
scanf("%s",a);
printf("enter the second string\n");
scanf("%s",b);
n=strlen(a);
m=strlen(b);

for(i=0;i<m;i++)
a[m+i]=b[i];
a[m+n]='/0';
printf("combined string is %s\n",a);
}

void push(int a1)
{
s1[++top]=a1;
}
int pop()
{
return(s1[top--]);
}

void rem()
{
if(r<f)
printf("under flow\n");
else
printf("the element removed is %d\n",q[f++]);
}

void insert1()
{
if(r==n1-1)
printf("over flow\n");
```

```
else
{printf("enter the element to insert\n");
scanf("%d",&q[++r]);
}
}

void disp()
{
if(r<f)
printf("empty\n");
else
{
int i;
printf("the elements in the queue are\n");
for(i=0;i<=r;i++)
printf("%d\n",q[i]);
printf("total number of elements in the queue is %d\n",i);
}
}

union
{
unsigned long int ip;
unsigned char a[4];
}x;

void push2()
{
if(top==n1-1)
printf("\nover flow\n");
else
{
printf("\nenter the element\n");
scanf("%d",&s1[++top]);
}
}

void pop1()
{
if(top==-1)
printf("\nunder flow\n");
else
printf("\nthe popped element is %d",s1[top--]);
}

void display3()
{
if(top==-1)
printf("\nstack is empty\n");
else
{
printf("\nthe steck elememts are\n");
for(i=top;i>-1;i--)
printf("%d\n",s1[i]);
printf("\nnumber of elements are %d\n",top+1);
 }
 }
```

```
void convert()
{
int i=0,j=0;
while(infix[i]!=0)
{
char ch=infix[i++];
if(isalpha(ch))
postfix[j++]=ch;
else
{
while(top!=-1&&precedence(stack[top],ch))
postfix[j++]=stack[top--];
if(top==-1||ch!=')')
stack[++top]=ch;
}
}
 while(top!=-1)
{
 char c;
 c=stack[top--];
 if(c!='(')
 postfix[j++]=c;
 }
 postfix[j]='\0';
}

int precedence(char top, int cur)
{
if(top!='('&&cur==')')
return 1;
else if((top=='$'||top=='*'||top=='/')&&(cur=='*'||cur=='+'||cur=='-'))
return 1;
else if((top=='+'||top=='-')&&(cur=='+'||cur=='-' ))
return 1;
else
return 0;
}

int sea(int l,int h)
{
if(l>h)
return -1;
else
{

m=(l+h)/2;
if(k>s1[m])
return(m+1,n-1);
else
return(l,m-1);
}
}

void tower(int n,char f, char a,char t)
{
if(n==1)
printf("disk is moved from %c to %c\n",f,t); else
```

```
{
tower(n-1,f,t,a);
printf("disk is moved from %c to %c\n",f,t);
tower(n-1,a,f,t);
}
}

void sort()
{

      for(i=1;i<n;i++)
      {
      for(j=0;j<n-i;j++)
      {
      if(s1[j]>s1[j+1])
      {
      t=s1[j];
      s1[j]=s1[j+1];
      s1[j+1]=t;
      }
      }
      }
      printf("the sorted array is \n");
      for(i=0;i<n;i++)
      printf("%d\n",s1[i]);
}


void read(int g1[10][10],int m,int n)
{
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&g1[i][j]);
}

void write(int g1[10][10],int m,int n)
{
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf("%d\t",g1[i][j]);
}
printf("\n");
}
}

void mul()
{
for(i=0;i<m;i++)
{
for(j=0;j<q2;j++)
{
f1[i][j]=0;
for(k=0;k<n;k++)
{f1[i][j]+=g1[i][k]*h1[k][j];
}
```

```
}
}
}

int gcd(int a3,int b3)
{
while(a3!=b3)
{
if(a3>b3)
a3-=b3;
else
b3-=a3;
}
return b3;
}


void main()
{
int w;
clrscr();
man:printf("\n1:lab programs\n 2:sorting programms\n3:searching
programs\n4:matrix programs\n5:conversion\n6:applications\n 7:exit\n");
scanf("%d",&w);
switch(w)
{

case 1:     man1: printf("1:files programs\n2: string
programs\n3:evaluate the valid postfix expression");
      printf("\n4:queue program\n5:ip address program\n6:stack
program\n");
      printf("7:conversion\n8:binary search using recursion 9: towers of
hanaio\n");
      printf("10:to main menu\n11:exit\n");
      scanf("%d",&w);

      switch(w)
      {
            case 1:
                  printf("enter the file name\n");
                  scanf("%s",fname);
                  printf("enter the number of records \n");
                  scanf("%d",&n);
          while(1)
          {
          printf("1:insert record\t 2: display\t 3:search\t 4: for sub
menu\n");
          scanf("%d",&b1);

          switch(b1)
          {
          case 1: insert();
                  break;
          case 2: display();
                  break;
          case 3: search();
                  break;          case 4:
```

```
                goto man1;
          default : printf("enter the valid input\n");
          }
          }

case 2:
      while(1)
       {
      printf("1:copy\t2:catinate\t3:to sub menuu\n");
      scanf("%d",&e);

      switch(e)
      {
      case 1:c();
      break;
      case 2:d();
      break;
      case 3: goto man1;
      default : printf("enter the valid input\n");
      }
      }

case 3:
      printf("enter the string\n");
      scanf("%s",p);
      len=strlen(p);
      for(i=0;i<len;i++)

      switch(p[i])
      {
      case '+' : b2=pop();
            a1=pop();
            push(a1+b2);
            break;
      case '-' :b2=pop();
            a1=pop();
            push(a1-b2);
            break;
      case '*' :b2=pop();
            a1=pop();
            push(a1*b2);
            break;
      case '/' :b2=pop();
            a1=pop();
            push(a1/b2);
            break;
      case '^' :b2=pop();
            a1=pop();
            push(pow(a1,b2));
            break;
      default:push(p[i]-'0');
      }
      printf("%d\n",pop());
      goto man1;

 case 4:    r=-1;
      f=0;
```

```
      while(1)
      {
      printf("1:insert 2:remove 3:display 4: exit\n");
      scanf("%d",&ch);

      switch(ch)

      {
            case 1: insert1();
                  break;
            case 2: rem();
                  break;
            case 3: disp();
                  break;
            case 4: goto man1;

            default : printf("invalid\n");
      }
      }

 case 5:
      printf("enter the ip address\n");
      scanf("%s",ipstr);
      q1=strtok(ipstr,".");
      p1=atoi(q1);
      x.a[3]=p1;
      for(i=2;i>=0;i--)
      {
      q1=strtok(NULL,".");
      p1=atoi(q1);
      x.a[i]=p1;
      }
      printf("long integer is %lu\n",x.ip);
      goto man1;

 case 6:while(1)
      {
      printf("\n1:push\n 2:pop\n3:display\n4:to the sum menu\n");
      scanf("%d",&w);

      switch(w)
      {
      case 1:push2();
            break;
      case 2: pop1();
            break;
      case 3: display3();
            break;
      case 4:goto man1;
      default:printf("enter the valid inout\n");
      }
      }

case 7:     printf("enter the valid infix expression\n");
      scanf("%s",infix);      convert();
      printf("the infix exp is %s\n",infix);
      printf("the postfix exp is %s\n",postfix);
```

```
      goto man1;

case 8 :printf("enter the number of elements \n");
      scanf("%d",&n);
      printf("enter the elements in the ascending order\n");
      for(i=0;i<n;i++)
      scanf("%d",&s1[i]);
      printf("enter the elememt to be searched\n");
      scanf("%d",&k);
      pos=sea(0,n-1);
      if(pos<0)
      printf("does not found\n");
      else
      printf("element is found is the position %d\n",pos+1);
      goto man1;

case 9:printf("enter the number of disks\n");
      scanf("%d",&n);
      if(n<1)
      printf("invalid\n");
      else
      {
      tower(n,'A','B','C');
      printf("total number of moves are %d\n",(int)pow(2,n)-1);
      }
      goto man1;


case 10: goto man;

case 11:exit(0);

default:printf("enter the valid input\n");
      goto man;
      }
case 2:  man2:printf("1:bubble sort\n2:selection sort\n3:main menu\n");
      scanf("%d",&w);

      switch(w)
      {
      case 1:
            printf("enter the number of elements in the array\n");
            scanf("%d",&n);
            printf("enter the elements of the array\n");
            for(i=0;i<n;i++)
            scanf("%d",&s1[i]);
            for(i=1;i<n;i++)
            {
            for(j=0;j<n-i;j++)
            {
            if(s1[j]>s1[j+1])
            {
            t=s1[j];
            s1[j]=s1[j+1];
.............
```

```
            s1[j+1]=t;
            }
            }
            }
            printf("the sorted array is \n");
            for(i=0;i<n;i++)
            printf("%d\n",s1[i]);
            goto man2;
      case 2:
            printf("enter the number of elements in the array\n");
            scanf("%d",&n);
            printf("enter the elements of the array\n");
            for(i=0;i<n;i++)
            scanf("%d",&s1[i]);
            for(i=0;i<n-1;i++)
            {
            pos=i;
            for(j=1+i;j<n;j++)
            {
            if(s1[j]<s1[pos])
            {
            pos=j;
            t=s1[i];
            s1[i]=s1[pos];
            s1[pos]=t;
            }
            }
              }
            printf("the sorted array is \n");
            for(i=0;i<n;i++)
            printf("%d\n",s1[i]);
            goto man2;
      case 3:goto man;
      default:printf("enter the valid input\n");
            goto man;

      }

case 3: man3: printf("1:binary search\n2:linear search\n3:main menu\n");
      scanf("%d",&w);

      switch(w)
      {

      case 1:   printf("enter the number of elements \n");
            scanf("%d",&n);
            printf("enter the elements in the ascending order\n");
            for(i=0;i<n;i++)
            scanf("%d",&s1[i]);
            sort();
            printf("enter the elememt to be searched\n");
```

```
            scanf("%d",&k);
            pos=sea(0,n-1);
            if(pos<0)
            printf("does not found\n");
            else
            printf("element is found is the position %d\n",pos+1);
            goto man3;
      case 2: printf("enter the number of elements \n");
            scanf("%d",&n);
            printf("enter the elements in the ascending order\n");
            for(i=0;i<n;i++)
            scanf("%d",&s1[i]);
            printf("enter the elememt to be searched\n");
            scanf("%d",&k);
            for(i=0;i<n;i++)
            {
            if(s1[i]==k)
            printf("the element is found in the position %d\n",i+1);
            }
            goto man3;
      case 3:goto man;
      default:printf("enter the valid input\n");
            goto man;

            }
case 4:
man4:printf("1:multiplication\n2:addition\n3:transpose\n4:trace\n5:main
menu\n6:exit\n");
            scanf("%d",&w);

      switch(w)
      {
      case 1:
            printf("enter the order of the first matrix\n");
            scanf("%d%d",&m,&n);
            printf("enter the order of the second matrix\n");
            scanf("%d%d",&p2,&q2);
            printf("enter the elements of the first matrix\n");
            read(g1,m,n);
            printf("first matrix is \n");
            write(g1,m,n);
            printf("enter the elements of the second matrix\n");
            read(h1,p2,q2);
            printf("second matrix is\n");
            write(h1,p2,q2);
            if(n!=p2)
            printf("multiplicatio is not possible\n");
            else
            {
            mul();
            printf("multplication matrix\n");
            write(f1,m,q2);
            }
            goto man4;
      case 2:printf("enter the order of the first matrix\n");
            scanf("%d%d",&m,&n);
            printf("enter the order of the second matrix\n");
```

```
      scanf("%d%d",&p2,&q2);
      printf("enter the elements of the first matrix\n");
      read(g1,m,n);
      printf("first matrix is \n");
      write(g1,m,n);
      printf("enter the elements of the second matrix\n");
      read(h1,p2,q2);
      printf("second matrix is\n");
      write(h1,p2,q2);
      if(m==n&&p2==q2&&m==p2)
      {
      for(i=0;i<m;i++)
      {
      for(j=0;j<m;j++)
      {
      f1[i][j]=g1[i][j]+h1[i][j];
      }
      }
      printf("the added matrix is\n");
      write(f1,m,n);
      }
      else
      printf("addition is not possilbe\n");
      goto man4;
case 3: printf("enter the order of the matrix\n");
      scanf("%d%d",&m,&n);
      printf("enter the elements of the matrix\n");
      for(i=0;i<m;i++)
      for(j=0;j<n;j++)
      scanf("%d",&h1[i][j]);
      printf(" inputted matrix is \n");
      for(i=0;i<m;i++)
      {
      for(j=0;j<n;j++)
      {
      printf("%d\t",h1[i][j]);
      }
      printf("\n");
      }
      for(i=0;i<m;i++)
      for(j=0;j<n;j++)
      {
      g1[j][i]=h1[i][j];
      }
      printf("transpose matrix is\n");
      for(i=0;i<n;i++)
      {
      for(j=0;j<m;j++)
      {
      printf("%d\t",g1[i][j]);
      }
      printf("\n");
      }

      goto man4;
case 4:
      printf("enter the order of the matrix\n");
```

```
            scanf("%d",&m);
            printf("enter the elements of the matrix\n");
            for(i=0;i<m;i++)
            for(j=0;j<m;j++)
            scanf("%d",&h1[i][j]);
            printf(" inputted matrix is \n");
            for(i=0;i<m;i++)
            {
            for(j=0;j<m;j++)
            {
            printf("%d\t",h1[i][j]);
            }
            printf("\n");
            }
            for(i=0;i<m;i++)
            for(j=0;j<m;j++)
            {
            if(i==j)
            sum+=h1[i][j];
            }
            printf("sum value is %d\n",sum);
            goto man4;
      case 5:     goto man;
      case 6:     exit(0);
      default:printf("enter the valid input\n");
      goto man;

      }
case 5:man5:printf("1:binary to decimal\n2:octal to decimal\n3:decimal
to binary\n4:main menu\n5:exit\n");
      scanf("%d",&w);
            switch(w)
      {

      case 1:
            printf("enter binary number\n");
            scanf("%d",&n);
            num=n;
            b3=1;
            sum=0;
            while(num!=0)
            {
            d1=num%10;
            num/=10;
            sum+=d1*b3;
            b3=b3*2;
            }
            printf("the binary number = %d\n",n);
            printf("the decimal number =%d\n",sum);
            goto man5;
      case 2:
            printf("enter octal number\n");
            scanf("%d",&n);
            num=n;
            b3=1;
            sum=0;
            while(num!=0)
```

```
            {
            d1=num%10;
            num/=10;
            sum+=d1*b3;
            b3=b3*8;
            }
            printf("the octal number = %d\n",n);
            printf("the decimal number =%d\n",sum);
            goto man5;
      case 3:
            printf("enter decimal number\n");
            scanf("%d",&n);
            num=n;
            b3=1;
            sum=0;
            while(num!=0)
            {
            d1=num%2;
            num=num/2;
            sum+=d1*b3;
            b3=b3*10;
            }
            printf("the decimal number = %d\n",n);
            printf("the binary number =%d\n",sum);
            goto man5;
      case 4:goto man;
      case 5:exit(0);
      default:printf("enter the valid input\n");
      goto man;
      }
case 6: man6:printf("1:to check for even\n2:largest of two
numbers\n3:largest of three numbers\n");

      printf("4:to find the gcd of three numbers\n5:to check for leap
year\n6:to check for validity of the date\n");
      printf("7:to check for prime\n8:to find the number of prime
numbers in between the given range\n");
      printf("9:to check for palindrome\n10:roots of a quadratic
equation\n11:fibnacci series\n12:to find the factorial of the
number\n13:main menu\n14:exit\n");
      scanf("%d",&w);

      switch(w)
      {
      case 1: printf("enter the number\n");
            scanf("%d",&n);
            if(n%2==0)
            printf("number is even\n");
            else
            printf("number is odd\n");
            goto man6;
      case 2:printf("enter the two number\n");
            scanf("%d%d",&n,&m);
            if(m>n)
            printf("%d is greater\n",m);
            else
            printf("%d is greater\n",n);
```

```
      goto man6;
case 3:printf("enter the three numbers\n");
      scanf("%d%d%d",&m,&n,&t);
      lar=m;
      if(lar<n)
      lar=n;

      if(lar<t)
      lar=t;
      printf("the largest of three numbers is %d\n",lar);
      goto man6;
 case 4:
      printf("enter the three numbers\n");
      scanf("%d%d%d",&a3,&b3,&c3);
      d3=gcd(a3,b3);
      g3=gcd(d3,c3);
      printf(" gcd of three numbers is %d\n",g3);
      goto man6;
 case 5:printf("enter the year\n");
      scanf("%d",&y);
      if(y%4==0||y%400==0)
      printf("leap\n");
      else
      printf("not a leap\n");
      goto man6;
 case 6:
      printf("enter the date in dd mm yy format\n");
      scanf("%d%d%d",&dd,&mm,&yy);
      if(dd<=0||dd>31||mm<=0||mm>12||yy<=0)
      valid=0;
      else
      if ((mm==4||mm==6||mm==9||mm==11)&&dd>30)
      valid=0;
      else
      if (mm==2)
      {
      if(yy%4==0)
      {
      if(dd>29)
      valid=0;
      }
      else
      {
      if(dd>28)
      valid=0;
      }
      }
      if(valid>0)
      printf("%d/%d/%d is a valid date",dd,mm,yy);
      else
      printf("%d/%d/%d is not a valid date\n",dd,mm,yy);
      goto man6;
case 7: printf("enter the number\n");
      scanf("%d",&w2);
      for(i=2;i<w2/2;i++)
      {
      if(w2%i==0)
```

```
            {
            printf("not a prime\n");
            goto man6;
            }
            }
            printf("prime\n");
            goto man6;
      case 8:
            printf(" enter the munber range\n");
            scanf("%d%d",&a3,&b3);
            printf("prime numbers are\n");
            for(i=a3;i<=b3;i++)
            {
            p3=1;
            for(n=2;n<=i/2;n++)
            {
            if(i%n==0)
            {
            p3=0;
            break;
            }
            }
            if(p3==0)
            continue;
            if(p3==1)
            printf("%d\t",i);
            c3++;
            }
            printf("\n total nmber of prime numbers found in the given
range is %d\n",c3);
             goto man6;
      case 9:

            printf("enter the number\n");
            scanf("%d",&on);
            n=on;
            rn=0;
            while(n!=0)
            {
            d3=n%10;
            n=n/10;
            rn=rn*10+d3;
            }
            printf("reverse number is %d\n",rn);
            if(on==rn)
            printf("palindrome\n");
            else
            printf("not a palindrome\n");
            goto man6;
      case 10:

            printf("enter the coefficients\n");
            scanf("%f%f%f",&a4,&b4,&c4);
            if(a4==0)
            {
            printf("invalid\n");
            goto man6;
```

```
            }
            else
            {
            d4=b4*b4-4*a4*c4;
            if(d4>0)
            {
            printf("roots are real and disinct\n");
            r1=(-b4+sqrt(d4))/(2*a4);
            r2=(-b4-sqrt(d4))/(2*a4);
            printf("r1=%f\nr2=%f\n",r1,r2);
            }
            else
            if(d4==0)
            {
            printf("roots are real and equal\n");
            r1=-b4/(2*a4);
            printf("r1=r2=%f\n",r1);
            }
            else
            {
            printf("roots are imaginary\n");
            r1=-b4/(2*a4);
            r2=sqrt(fabs(d4))/(2*a4);
            printf("r1=%f+i(%f)\nr2=%f-i(%f)\n",r1,r2,r1,r2);
            }
            }
                goto man6;
       case 11:printf("enter the n value\n");
            scanf("%d",&n);
            f2[0]=0;
            f2[1]=1;
            for(i=2;i<=n;i++)
            f2[i]=f2[i-1]+f2[i-2];
            printf("the fibonacci series is\n 0\n1\n");
            for(i=2;i<=n;i++)
            printf("%d\n",f2[i]);
            goto man6;
       case 12:printf("enter the n value\n");
            scanf("%d",&n);
            for(i=1;i<=n;i++)
            fact=fact*i;
            printf("the factorial of %d is %d\n",n,fact);
            goto man6;
       case 13:goto man;
       case 14:exit(0);
       default:printf("enter the valid input\n");
      goto man;

      }

case 7:exit(0);
default:printf("enter the valid input\n");
      goto man;
      }

      }
```