GOVERNMENT ARTS COLLEGE FOR WOMEN
RAMANATHAPURAM

PG DEPARTMENT OF COMPUTER SCIENCE

PROGRAMMING IN C LAB – 7BCE1P1

I B.Sc. COMPUTER SCIENCE

I-SEMESTER

2020 - 2021

SIMPLE PROGRAMS

SOURCE CODE: AO.C

```
//Arithemetic Operations
#include<stdio.h>
#include<conio.h>
void main()
{
    int x,y,a,s,m,d,r;
    clrscr();
    printf("ARITHMETIC OPERATIONS\n");
    printf("********** **********\n");

    printf("Enter the two numbers:\n");
    scanf("%d%d",&x,&y);

    a=x+y;
    s=x-y;
    m=x*y;
    d=x/y;
    r=x%y;

    printf("\nX+Y  = %d",a);
    printf("\nX-Y  = %d",s);
    printf("\nX*Y  = %d",m);
    printf("\nX/Y  = %d",d);
    printf("\nX mod Y  = %d",r);

    getch();
}
```

OUTPUT:

```
ARITHMETIC OPERATIONS
********** **********
Enter the two numbers:
12
5

X+Y  = 17
X-Y  = 7
X*Y  = 60
X/Y  = 2
X mod Y  = 2
```

SOURCE CODE: EE.C

```
//Evaluation of an expression
#include<stdio.h>
#include<conio.h>
void main()
{
    int x,y,z;
    clrscr();

    x=2;  y=4;

    z=(x+y) * (x+y);

    printf("\n\t\tEVALUATION OF AN EXPRESSION\n");
    printf("\t\t*************************\n");

    printf("\nX => %d",x);
    printf("\nY => %d",y);
    printf("\n(X+Y) * (X+Y) => %d",z);

    getch();
}
```

OUTPUT:

```
            EVALUATION OF AN EXPRESSION
            *************************

X => 2
Y => 4
(X+Y) * (X+Y) => 36
```

GROUP –A

INDEX

SOURCE CODE: SOD.C

```
//To find the sum of digits of a given integer
#include <stdio.h>
#include <conio.h>
void main()
{
    int n,r,s;
    clrscr();
    printf("\n\t\tSUM OF DIGITS");
    printf("\n\t\t----------------");
    printf("\nEnter an integer:");
    scanf("%d",&n);

    r=s=0;
    while(n>0)
    {
            r=n%10;
            s=s+r;
            n=n/10;
    }

    printf("sum of digits:%d",s);
    getch();
}
```

OUTPUT:

```
            SUM OF DIGITS
            ----------------
Enter an integer:235
sum of digits:10
```

SOURCE CODE: ARM.C

```
/*ARMSTRONG NUMBER CHECKING*/
#include <stdio.h>
#include<conio.h>
void main()
{
    int n,c,r,s=0;
    clrscr();
    printf("\t\tARMSTRONG NUMBER CHECKING\n");
    printf("\t\t************************\n");
    printf("Enter the value:");
    scanf("%d",&n);

    c=n;
    while(n>0)
    {
            r=n%10;
            s=s+(r*r*r);
            n=n/10;
    }

    if(c==s)
            printf("The given number %d is Armstrong number",c);
    else
            printf("The given number %d is not armstrong number",c);
getch();
}
```

OUTPUT 1:

```
            ARMSTRONG NUMBER CHECKING
            ************************
Enter the value:204
The given number 204 is not armstrong number
```

OUTPUT 2:

```
          ARMSTRONG NUMBER CHECKING
          ************************
Enter the value:407
The given number 407 is Armstrong number
```

SOURCE CODE: PRIME.C

```
// Prime number checking
#include <stdio.h>
#include <conio.h>
void main()
{
    int n,i,count;
    clrscr();
    printf("\n\t PRIME NUMBER CHECKING");
    printf("\n\t====================");

    printf("\nENTER YOUR INTEGER: ");
    scanf("%d",&n);

    i=2; count=0;
    while(i<=n-1)
    {
            count= ((n%i)==0)? count+1 : count+0;
            i++;
    }

    if (count==0)
         printf("PRIME NUMBER");
    else
        printf("NOT A PRIMENUMBER");

    getch();
}
```

OUTPUT 1:

```
            PRIME NUMBER CHECKING
            ==================
     ENTER YOUR INTEGER: 6
     NOT A PRIMENUMBER
```

OUTPUT 2:

```
            PRIME NUMBER CHECKING
             ==================
    ENTER YOUR INTEGER: 13
    PRIME NUMBER
```

SOURCE CODE: FIBO.C

```
//Program for generate fibonacci series
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,f1,f2,sum,i;
   clrscr();

   printf("\n\t\tFIBONACCI SERIES");
   printf("\n\t\t================");

   printf("\nHow many numbers do you want in your fibonacci series? ");
   scanf("%d",&n);

   sum=0;
   i=3;
   f1=0;
   f2=1;

   printf("\nf1:%d",f1);
   printf("\nf2:%d",f2);

   while(i<=n)
    {
       sum=f1+f2;
       printf("\nf%d:%d",i,sum);
       f1=f2;
       f2=sum;
       i++;
    }

    getch();
}
```

OUTPUT:

```
            FIBONACCI SERIES
            ==============
How many numbers do you want in your fibonacci series? 8

f1:0
f2:1
f3:1
f4:2
f5:3
f6:5
f7:8
f8:13
```

SOURCE CODE: ADAM.C

```
//program for adam number checking

#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
   int n,sq,sq1,rev1,rev2,rt,rt1,r,sum;
   clrscr();

   printf("\n\t\tADAM NUMBER");
   printf("\n\t\t@@@@@@@@@@");

   printf("\nEnter a number: ");
   scanf("%d",&n);

   sq=n*n;

   sq1=sq;
   sum=0;
   while(sq>0)
   {
      r=sq%10;    sum=(sum*10)+r;        sq=sq/10;
   }

   rev1=sum;

   rt=sqrt((double)rev1);

   rt1=rt;
   sum=0;
   while(rt>0)
   {
      r=rt%10;    sum=(sum*10)+r;        rt=rt/10;
   }

   rev2=sum;

   printf("\nGiven number:%d",n);
   printf("\nSquare:%d",sq1);
   printf("\nReverse:%d",rev1);
   printf("\nRoot:%d",rt1);
   printf("\nReverse:%d",rev2);
```

```
        if(rev2==n)
            printf("\nADAM");
        else
            printf("\nNOT ADAM");

        getch();
    }
```

OUTPUT 1:

```
                ADAM NUMBER
                @@@@@@@@@
    Enter a number: 8

    Given number:8
    Square:64
    Reverse:46
    Root:6
    Reverse:6
    NOT ADAM
```

OUTPUT 2:

```
                ADAM NUMBER
                @@@@@@@@@
    Enter a number: 13

    Given number:13
    Square:169
    Reverse:961
    Root:31
    Reverse:13
    ADAM
```

SOURCE CODE: REVERSE.C

```
//To reverse a number and string
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
     int n,s,r;
     char string[100];
     clrscr();

     printf("\n\t\tReverse of number and string");
     printf("\n\t\t*****************************");

     printf("\nEnter your number: ");
     scanf("%d",&n);

     s=r=0;
     while(n>0)
     {
     r=n%10;     s=(s*10)+r;         n=n/10;
     }

     printf("\nReversed Number:%d",s);

     printf("\n\nEnter your string: ");
     scanf("%s",string);

     printf("\nReversd string:%s",strrev(string));

     getch();
}
```

OUTPUT:

```
                Reverse of number and string
                *****************************
Enter your number: 341

Reversed Number:143

Enter your string: Ramanathapuram

Reversd string:marupahtanamaR
```

SOURCE CODE: MINMAX.C

```
//To find minimum and maximum of given 'n'numbers
#include <stdio.h>
#include <conio.h>
void main()
{
    int n,i,a[100],min,max;
    clrscr();

    printf("\n\t\tMINIMUM AND MAXIMUM OF GIVEN 'N'NUMBERS");
    printf("\n\t\t==================================");

    printf("\nHow many numbers do you need? ");
    scanf("%d",&n);

    printf("\nEnter %d numbers:\n",n);
    for(i=1;i<=n;i++)
    scanf("%d",&a[i]);

    min=a[1];
    for(i=2;i<=n;i++)
    {
            if(min<a[i])
                    continue;
           else
                    min=a[i];
    }

    max=a[1];
    for(i=2;i<=n;i++)
    {
            if(max>a[i])
                    continue;
            else
                    max=a[i];
     }

     printf("\n Minimum: %d",min);
     printf("\n Maximum: %d",max);

     getch();
 }
```

OUTPUT:

```
                MINIMUM AND MAXIMUM OF GIVEN 'N'NUMBERS
                ===============================
    How many numbers do you need? 5

    Enter 5 numbers:
    12
    -9
    8
    0
    87

     Minimum: -9
     Maximum: 87
```

SOURCE CODE: ASCEND.C

```
//Ascending order
#include <stdio.h>
#include <conio.h>
void main()
{
    int n,i,j,a[100],temp;
    clrscr();
    printf("\n\t\tASCENDING ORDERED NUMBER");
    printf("\n\t\t**********************");

    printf("\nHow many numbers do you need? ");
    scanf("%d",&n);

    printf("\nEnter %d numbers\n",n);
    for(i=1;i<=n;i++)
            scanf("%d",&a[i]);

    for(i=1;i<=n;i++)
            for(j=i+1;j<=n;j++)
            if(a[i]>a[j])
            {
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
            }

    printf("\nASCENDING ORDERED NUMBERS:");
    for(i=1;i<=n;i++)
    printf("\n %d",a[i]);

    getch();
}
```

OUTPUT:

```
                ASCENDING ORDERED NUMBER
                **********************
    How many numbers do you need? 5

    Enter 5 numbers
    23
    12
    34
    6
    33

    ASCENDING ORDERED NUMBERS:
     6
     12
     23
     33
     34
```

SOURCE CODE: MADD.C

```
// matrix addition
#include <stdio.h>
#include <conio.h>
void main()
{
    int r,c,i,j,first[10][10],second[10][10],result[10][10];
    clrscr();

    printf ("\n\t\t MARTIX ADDITION");
    printf("\n\t\t  ^^^^^^^^^^^^^^^^");
    printf("\n\t Row-> ");              scanf("%d",&r);
    printf("\n\t Column-> ");          scanf("%d",&c);

    printf("\n\tEnter first matrix elements:\n");
    for(i=1;i<=r;i++)
            for(j=1;j<=c;j++)
            scanf("%d",&first[i][j]);

    printf("\n\tEnter second matrix elements:\n");
    for(i=1;i<=r;i++)
            for(j=1;j<=c;j++)
                    scanf("%d",&second[i][j]);

    for(i=1;i<=r;i++)
            for(j=1;j<=c;j++)
                    result[i][j]=first[i][j]+second[i][j];

    printf("\n FIRST MATRIX:\n");
    for(i=1;i<=r;i++)
    {
            for(j=1;j<=c;j++)
                     printf("\t%d",first[i][j]);
             printf("\n");
    }

    printf("\n SECOND MATRIX:\n");
    for(i=1;i<=r;i++)
    {
            for(j=1;j<=c;j++)
                    printf("\t%d",second[i][j]);
            printf("\n");
    }


    printf("\n RESULT MATRIX:\n");
```

```
    for(i=1;i<=r;i++)
    {
        for(j=1;j<=c;j++)
            printf("\t%d",result[i][j]);
        printf("\n");
    }

    getch();
}
```

OUTPUT:

```
            MARTIX ADDITION
            ^^^^^^^^^^^^^

      Row-> 2

      Column-> 2

      Enter first matrix elemens:t
1
2
3
4

      Enter second matrix elements:
5
6
5
6

FIRST MATRIX:
     1      2
     3      4

SECOND MATRIX:
     5      6
     5      6

RESULT MATRIX:
     6      8
     8      10
```

SOURCE CODE: MMU.C

```
//matrix multiplication
#include <stdio.h>
#include <conio.h>

#define scan(row,column,matrix){\
for(i=1;i<=row;i++)  \
for(j=1;j<=column;j++) \
scanf("%d",&matrix[i][j]); \
}

#define print(row,column,matrix){\
for(i=1;i<=row;i++)    \
{\
for(j=1;j<=column;j++)  \
printf("\t%d",matrix[i][j]); \
printf("\n");\
}\
}

void main()
{
      int r1,c1,r2,c2,a[10][10],b[10][10],c[10][10],i,j,k;
      clrscr();

      printf("\t\tMATRIX MULTIPLICATION \n");
      printf("\t\t==================== \n");

      printf("\n Enter the order of matrix A:\n");
      printf("Row -> ");        scanf("%d",&r1);
      printf("Column -> ");    scanf("%d",&c1);

      printf("\n Enter the order of matrix B:\n");
      printf("Row -> ");        scanf("%d",&r2);
      printf("Column -> ");    scanf("%d",&c2);

      if(c1!=r2)
      {
            printf("Matrix multiplication is not possible");
            goto end;
      }

      printf("Enter the elements of matrix A:\n");       scan(r1,c1,a);
      printf("Enter the elements of matrix B:\n");       scan(r2,c2,b);

    for(i=1;i<=r1;i++)
```

```
            for(j=1;j<=c2;j++)
            {
                  c[i][j]=0;
                  for(k=1;k<=r2;k++)
                        c[i][j] += a[i][k] * b[k][j];
            }

      printf("\n A Matrix:\n");          print(r1,c1,a);
      printf("\n B Matrix:\n");          print(r2,c2,b);
      printf("\n Result matrix:\n");     print(r1,c2,c);

      end:  getch();
   }
```

OUTPUT 1:

```
            MATRIX MULTIPLICATION
            ================

 Enter the order of matrix A:
Row -> 2
Column -> 3

 Enter the order of matrix B:
Row -> 2
Column -> 3
Matrix multiplication is not possible
```

OUTPUT 2:

```
                MATRIX MULTIPLICATION
                =================

 Enter the order of matrix A:
Row -> 2
Column -> 3

 Enter the order of matrix B:
Row -> 3
Column -> 1
Enter the elements of matrix A:
2
3
4
2
1
3
Enter the elements of matrix B:
2
1
3

A Matrix:
       2       3       4
       2       1       3

 B Matrix:
       2
       1
       3

 Result matrix:
       19
       14
```

SOURCE CODE: NCRNPR.C

```
//NCR and NPR calculation
#include <stdio.h>
#include <conio.h>

int fact(int);

int fact(int x)
{
    int i,r;
    for(i=x, r=1;i>=1;i--)
            r=r*i;
    return(r);
}

void main()
{
    int n,r;
    float ncr,npr;

    clrscr();
    printf("\n\t\t NCR AND NPR CALCULATION");
    printf("\n\t\t **********************");

    printf("\nEnter the value of n:");        scanf("%d",&n);
    printf("\nEnter the value of r:");        scanf("%d",&r);

    ncr=fact(n)/ (fact(r)*fact(n-r));
    npr=fact(n)/fact(n-r);

    printf("\n NCR :%f",ncr);
    printf("\n NPR :%f",npr);

    getch();
}
```

OUTPUT:

```
                NCR AND NPR CALCULATION
                **********************
Enter the value of n:5

Enter the value of r:2

 NCR :10.000000
 NPR :20.000000
```

GROUP –B

INDEX

SOURCE CODE: SG.C

```
//Student grade using else if ladder
#include <stdio.h>
#include <conio.h>
void main()
{
    int t,e,c,fc,total;
    float p;

    clrscr();
    printf("\n\tSTUDENT GRADE USING ELSE IF LADDER\n");

    printf("\n Enter the marks:\n");
    printf("\n Tamil=");          scanf("%d",&t);
    printf("\n English=");       scanf("%d",&e);
    printf("\n C=");             scanf("%d",&c);
    printf("\n FC=");            scanf("%d",&fc);

    if(t<40 || e<40 || c<40 || fc<40)
    {
            printf("\nResult: FAIL");
            goto end;
    }

    total=t+e+c+fc;
    printf("\n Total marks= %d/400",total);

    p=total/4.0;
    printf("\n Percentage= %f%",p);

    if(p>=80)
            printf("\n Your Grade is:A+");
    else if(p>=60)
            printf("\n Your Grade is:A");
        else if(p>=50)
                printf("\n Your Grade is:B");
              else
                  printf("\n Your Grade is:C");

    end:  getch();
}
```

OUTPUT 1:

STUDENT GRADE USING ELSE IF LADDER

Enter the marks:

Tamil=43
English=38
C=56
FC=55

Result: FAIL

OUTPUT 2:

STUDENT GRADE USING ELSE IF LADDER

Enter the marks:

Tamil=82
English=83
C=80
FC=82

Total marks= 327/400
Percentage= 81.750000%
Your Grade is:A+

OUTPUT 3:

STUDENT GRADE USING ELSE IF LADDER

Enter the marks:

Tamil=70
English=60
C=62
FC=60

Total marks= 252/400
Percentage= 63.000000%
Your Grade is:A

OUTPUT 4:

STUDENT GRADE USING ELSE IF LADDER

Enter the marks:

Tamil=40
English=43
C=59
FC=60

Total marks= 202/400
Percentage= 50.500000%
Your Grade is:B

OUTPUT 5:

STUDENT GRADE USING ELSE IF LADDER

Enter the marks:

Tamil=40
English=41
C=42
FC=40

Total marks= 163/400
Percentage= 40.750000%
Your Grade is:C

SOURCE CODE: SHFUN.C

```
//String handling functions
#include <stdio.h>
#include <conio.h>
#include <string.h>

void main()
{
    char s1[20],s2[20],s3[20];
    int x,l1,l2,l3;

    clrscr();
    printf("\n\t\tUSE OF STRING HANDLING FUNCTIONS\n");

    printf("\n\nEnter two string constants: \n");
    scanf("%s%s",s1,s2);

    x=strcmp(s1,s2);
    if(x!=0)
    {
        printf("\n\nStrings are NOT EQUAL \n");
        strcat(s1,s2);
    }
    else
        printf("\n\nStrings are EQUAL\n");

    strcpy(s3,s1);

    l1=strlen(s1);
    l2=strlen(s2);
    l3=strlen(s3);

    printf("\ns1=%s\t Length=%d characters\n",s1,l1);
    printf("\ns2=%s\t Length=%d characters\n",s2,l2);
    printf("\ns3=%s\t Length=%d characters\n",s3,l3);

    getch();
}
```

OUTPUT 1:

USE OF STRING HANDLING FUNCTIONS

Enter two string constants:
Aim Achieve

Strings are NOT EQUAL

s1=AimAchieve Length=10 characters

s2=Achieve Length=7 characters

s3=AimAchieve Length=10 characters

OUTPUT 2:

USE OF STRING HANDLING FUNCTIONS

Enter two string constants:
Aim Aim

Strings are EQUAL

s1=Aim Length=3 characters

s2=Aim Length=3 characters

s3=Aim Length=3 characters

SOURCE CODE: QE.C

```
//quadratic equation
#include<stdio.h>
#include<conio.h>
#include<math.h>

void main()
{

    int flag,a,b,c,d;
    double x1,x2;

    clrscr();
    printf("\n\t\tQUDRATIC EQUATION\n");

    printf("Enter the value of a: ");          scanf("%d",&a);
    printf("Enter the value of b: ");          scanf("%d",&b);
    printf("Enter the value of c: ");          scanf("%d",&c);

    if(a==0)
    {
            printf("\nNo solutions");
            goto end;
    }

    d=(b*b)-(4*a*c);
    if(d>0)              flag=1;
    if(d==0)             flag=2;
    if(d<0)              flag=3;

    switch(flag)
    {
    case 1:
            printf("\nRoots are real and different.\n");
            x1=(-b+sqrt((double)d))/(2*a);x2=(-b-sqrt((double)d))/(2*a);
            printf("\n x1=%lf     x2=%lf ",x1,x2);
            break;

    case 2:
            printf("\nRoots are equal.\n");
            x1=-b/(2.0*a);      x2=-b/(2.0*a);
            printf("\n x1=%lf     x2=%lf ",x1,x2);
            break;
```

```
    case 3:
            printf("\nRoots are complex.");
            printf("\n x1=%lf + %lfi ",-b/(2.0*a),sqrt((double)-d)/(2*a));
            printf("\n x2=%lf - %lfi ",-b/(2.0*a),sqrt((double)-d)/(2*a));
            break;
    }

    end:   getch();
}
```

OUTPUT 1:

```
              QUDRATIC EQUATION
Enter the value of a: 0
Enter the value of b: 3
Enter the value of c: 4

No solutions
```

OUTPUT 2:

```
              QUDRATIC EQUATION
Enter the value of a: 1
Enter the value of b: 1
Enter the value of c: 12

Roots are complex.
 x1=-0.500000 + 3.427827i
 x2=-0.500000 - 3.427827i
```

OUTPUT 3:

```
              QUDRATIC EQUATION
Enter the value of a: 4
Enter the value of b: 4
Enter the value of c: 1

Roots are equal.

 x1=-0.500000     x2=-0.500000
```

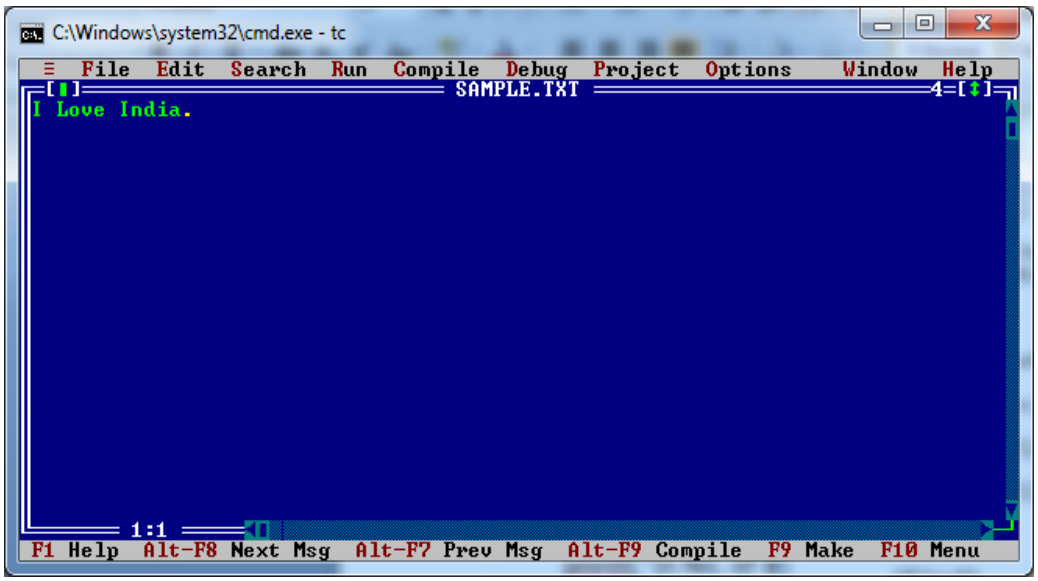
OUTPUT 4:

```
              QUDRATIC EQUATION
Enter the value of a: 1
Enter the value of b: 5
Enter the value of c: 6

Roots are real and different.

 x1=-2.000000     x2=-3.000000
```

SOURCE CODE: FILEPGM.C

```
//To count letters, words and lines in a file.
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *f;
    char ch,fname[30];
    int c,w,l;

    clrscr();
    printf("Enter the file name: ");
    scanf("%s",fname);

    f=fopen(fname,"r");

    if(f==NULL)
    {
            printf("\nUnable to open the file.");
            goto end;
    }

    c=w=l=0;
    while( (ch=fgetc(f)) !=EOF)
    {
            c++;

            if(ch=='\n' || ch=='\0')
                    l++;

            if(ch==' ' || ch=='\t' || ch=='\n' || ch=='\0')
                    w++;
    }

    if(c>0)
    {
            w++;
            l++;
    }

    printf("\n No. of characters: %d",c);
    printf("\n No. of words: %d",w);
    printf("\n No. of lines: %d",l);

    fclose(f);

    end:   getch();
}
```

FILE: SAMPLE.TXT



OUTPUT 1:

Enter the file name: sample.txt

No. of characters: 13
No. of words: 3
No. of lines: 1

OUTPUT 2:

Enter the file name: simple.txt

Unable to open the file.

SOURCE CODE: MLSTRUCT.C

```
//Student marklist preparation using arry of structure
#include <stdio.h>
#include <conio.h>

struct student
{
      char sname [30];
      int t,e,c,fc,total;
      float p;
};

void main()
{
      int n,i;
      struct student s[10];

      clrscr();
      printf("\n\t\t  STUDENT MARKLIST");
      printf("\n\t\t  -----------------");

      printf("\nHow many students are in your class? ");       scanf("%d",&n);

      printf("\nEnter the student details:\n");
      for(i=1;i<=n;i++)
      {
         printf("\n Name: ");                  scanf("%s",s[i].sname);
         printf("\n Tamil Mark: ");            scanf("%d",&s[i].t);
         printf("\n English Mark: ");          scanf("%d",&s[i].e);
         printf("\n C Mark: ");                scanf("%d",&s[i].c);
         printf("\n FC Mark: ");               scanf("%d",&s[i].fc);

         s[i].total=s[i].t+s[i].e+s[i].c+s[i].fc;
         s[i].p=s[i].total/4.0;
      }


printf("\n_____________________________________________________\n");
printf("NAME \t TAMIL \t ENG \t C \t FC \t TOTAL \t PER");
printf("\n _____________________________________________________\n");

for(i=1;i<=n;i++)
  printf("%s \t %d \t %d \t %d \t %d \t %d \t %f \n" , s[i].sname, s[i].t, s[i].e, s[i].c, s[i].fc, s[i].total, s[i].p);

printf("\n_____________________________________________________\n");
getch();
}
```

OUTPUT:

STUDENT MARKLIST

How many students are in your class? 2

Enter the student details:

Name: Raji

Tamil Mark: 67

English Mark: 56

C Mark: 56

FC Mark: 56

Name: Yuva

Tamil Mark: 67

English Mark: 56

C Mark: 56

NAME	TAMIL	ENG	C	FC	TOTAL	PER
Raji	67	56	56	56	235	58.750000
Yuva	67	56	56	51	230	57.500000

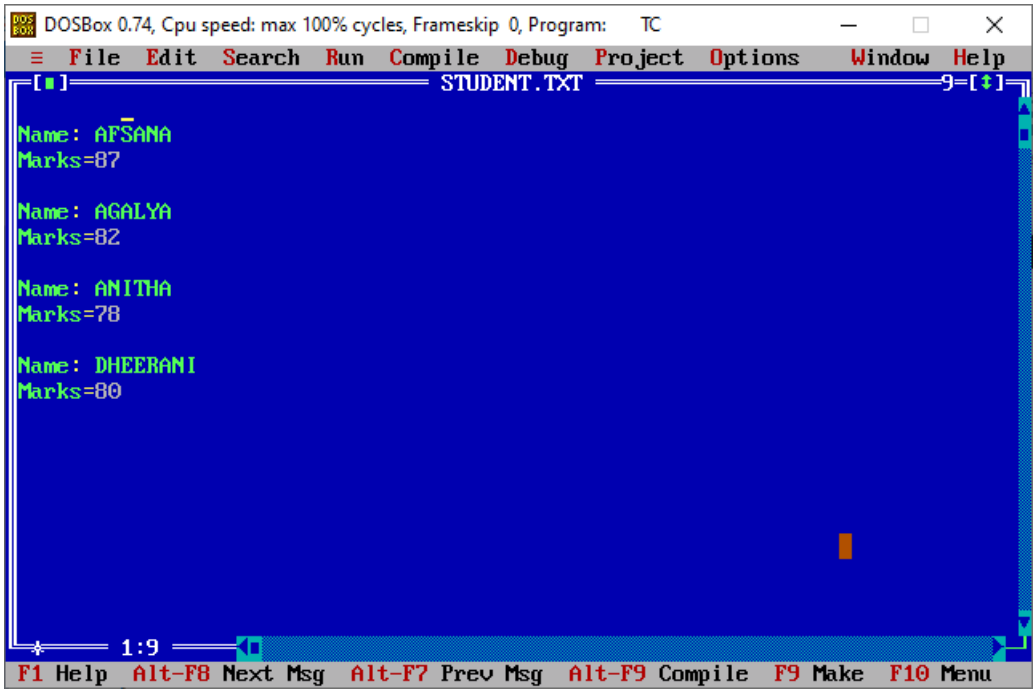
SOURCE CODE: FILEPGMSML.C

```
//student mark list processing using file
#include <stdio.h>
void main()

{

char name[50];
FILE *fptr;
int marks,i,n;
printf("Enter number of students: ");
scanf("%d",&n);

fptr=(fopen("student.txt","w"));
if(fptr==NULL)
{
        printf("Error!");
        exit(1);
}

for (i=0;i<n;++i)
{
        printf("For student%d\nEnter name: ",i+1);
        scanf("%s",name);
        printf("Enter marks: ");
        scanf("%d",&marks);
        fprintf(fptr,"\nName: %s \nMarks=%d \n",name,marks);
}

fclose(fptr);
getch();
}
```

OUTPUT:

```
Enter number of students: 4

For student1
Enter name:AFSANA
Enter marks:87

For student2
Enter name:AGALYA
Enter marks:82

For student3
Enter name:ANITHA
Enter marks:78

For student3
Enter name:DHEERANI
Enter marks:80
```

FILE: STUDENT.TXT



SOURCE CODE: FILEPGMPAYBILLL.C

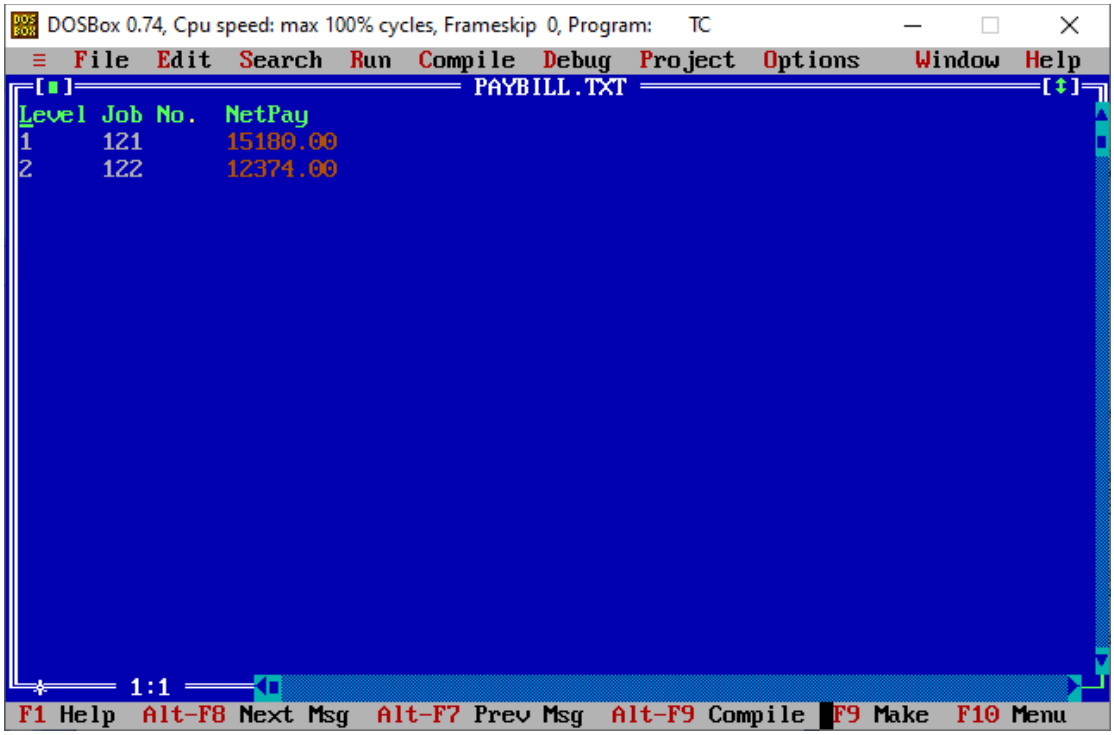
```
//paybill processing using file
  #include<stdio.h>
  #include<conio.h>

  #define    CA1   1000
  #define    CA2    750
  #define    CA3    500
  #define    CA4    250
  #define    EA1    500
  #define    EA2    200
  #define    EA3    100
  #define    EA4      0

  void main()
  {
     int   level, jobnumber;
     float gross,
       basic,
       house_rent,
       perks,
       net,
       incometax;
     FILE *fptr;

     fptr=(fopen("paybill.txt","w"));
   if(fptr==NULL)
   {
          printf("Error!");
          exit(1);
   }

     fprintf(fptr,"Level Job No.  NetPay\n");
     input:
     printf("\nEnter level, job number, and basic pay\n");
     printf("Enter 0 (zero) for level to END\n\n");
     scanf("%d", &level);
     if (level == 0) goto stop;
     scanf("%d %f", &jobnumber, &basic);
     switch (level)
     {
     case 1:
            perks = CA1 + EA1;
            break;
     case 2:
            perks = CA2 + EA2;
            break;
     case 3:
            perks = CA3 + EA3;
            break;
     case 4:
            perks = CA4 + EA4;
            break;
```

```
        default:
                printf("Error in level code\n");
                goto stop;
        }

        house_rent = 0.25 * basic;
        gross = basic + house_rent + perks;

        if (gross <= 2000)
        incometax = 0;
        else if (gross <= 4000)
            incometax = 0.03 * gross;
          else if (gross <= 5000)
                  incometax = 0.05 * gross;
                else
                  incometax = 0.08 * gross;
       net = gross - incometax;
       fprintf(fptr,"%d \t %d \t %.2f\n", level, jobnumber, net);
       goto input;

       stop: printf("\n\nEND OF THE PROGRAM");
       getch();
    }
```

OUTPUT:

```
Enter level, job number, and basic pay
Enter 0 (zero) for level to END

1
121
12000

Enter level, job number, and basic pay
Enter 0 (zero) for level to END

1
122
10000

Enter level, job number, and basic pay
Enter 0 (zero) for level to END

0
END OF THE PROGRAM
```

FILE: PAYBILL.TXT



SOURCE CODE: FILEPGMINVENTORY.C

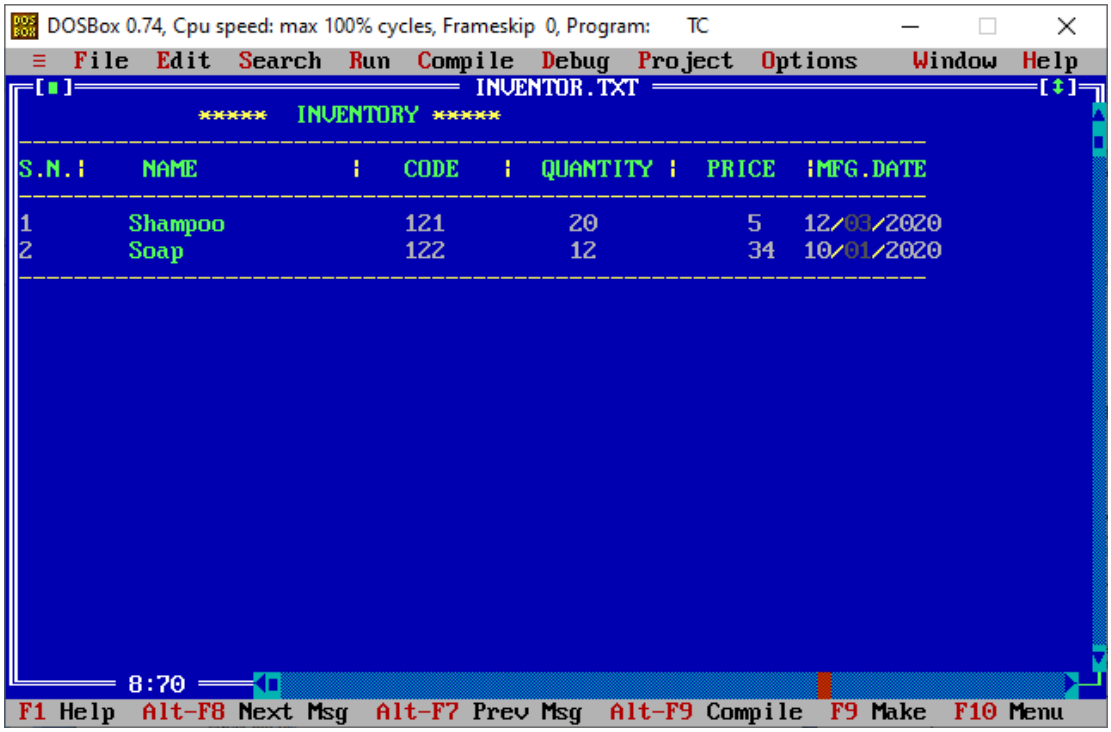
```
//Inventory control processing using file
#include <stdio.h>
#include <conio.h>
void main()
{

    struct date
    {
        int day;
        int month;
        int year;
    };
    struct details
    {
        char name[20];
        int price;
        int code;
        int qty;
        struct date mfg;
    };

    struct details item[50];
    int n,i;
    FILE *fptr;
    fptr=(fopen("inventory.txt","w"));
    if(fptr==NULL)
    {
        printf("Error!");
        exit(1);
    }

    clrscr();
    printf("Enter number of items:");
    scanf("%d",&n);
    fflush(stdin);
    for (i=0;i<n;i++)
    {
        fflush(stdin);
        printf("Item name:");
        scanf("%[^\n]",item[i].name);
        fflush(stdin);
        printf("Item code:");
        scanf("%d",&item[i].code);
        fflush(stdin);
        printf("Quantity:");
        scanf("%d",&item[i].qty);
        fflush(stdin);
        printf("price:");
        scanf("%d",&item[i].price);
        fflush(stdin);
        printf("Manufacturing date(dd-mm-yyyy):");
        scanf("%d-%d-
%d",&item[i].mfg.day,&item[i].mfg.month,&item[i].mfg.year);
```

```
        }
      fprintf(fptr,"          *****  INVENTORY *****\n");
      fprintf(fptr,"-------------------------------------------------------------------\n");
      fprintf(fptr,"S.N.|      NAME              |    CODE    |   QUANTITY  |   PRICE
|MFG.DATE\n");
      fprintf(fptr,"-------------------------------------------------------------------\n");

      for (i=0;i<n;i++)
        fprintf(fptr,"%d              %-15s              %-d          %-5d          %-5d
%d/%d/%d\n",i+1,item[i].name,item[i].code,item[i].qty,item[i].price,item[i].mfg.day
,item[i].mfg.month,item[i].mfg.year);

      fprintf(fptr,"-------------------------------------------------------------------\n");
      getch();
   }
```

OUTPUT:

```
Enter number of items:2
Item name:Shampoo
Item code:123
Quantity:20
price:5
Manufacturing date(dd-mm-yyyy):12-03-2020

Item name:Soap
Item code:122
Quantity:12
price:34
Manufacturing date(dd-mm-yyyy):10-01-2020
```

FILE: INVENTORY.TXT



SOURCE CODE: FILEPGMEBBILL.C

```
//generating electricity bill using structure and file
#include<stdio.h>

struct Bill
{
   char firstName[10];
   char lastName[10];
   char Address[20];
   float previousUnit;
   float presentUnit;
};

float generateBill(struct Bill temp)
{
   float diff;
   diff = temp.presentUnit - temp.previousUnit;
   if(diff > 20)
    return diff*4.75;
   else
    return 20*4.75+(diff-20)*7.75;

}
void main()
{
   struct Bill bill;
   FILE *fptr;
    fptr=(fopen("ebbill.txt","w"));
    if(fptr==NULL)
    {
            printf("Error!");
            exit(1);
    }
   printf("Fill up the following: \n");

   printf("First Name: ");
   gets(bill.firstName);

   printf("Last Name: ");
   gets(bill.lastName);

   printf("Address: ");
   gets(bill.Address);

   printf("Previous Unit: ");
   scanf("%f",&bill.previousUnit);

   printf("Present Unit: ");
   scanf("%f",&bill.presentUnit);

   fprintf(fptr,"\a\n\n**********Electricity Bill**********\n\n\a");
   fprintf(fptr,"Name: %s %s",bill.firstName,bill.lastName);
   fprintf(fptr,"\nAddress: %s",bill.Address);
```

```
        fprintf(fptr,"\nPrevious Unit: %.3f Current Unit: %.3f",bill.previousUnit,bill.presentUnit);
        fprintf(fptr,"\nCost: %.3f\n\n", generateBill(bill));

        getch();
    }
```

OUTPUT:

```
Fillup the following:
First Name: Harshatha
Last Name: Kumar
Address: North Street, Ramanathapuram
Previous Unit: 1605
Present Unit: 1725
```

FILE: EBBILL.TXT