**PSP LAB PROGRAMS**

**1. PRIME OR NOT**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,n,flag=0;

clrscr();

printf("enter the number to check prime or not \n");

scanf("%d" , &n);

for(i=2;i<n;i++)

{

if(n%i==0)

{

flag=1;

break;

}

}

if(flag==0)

{

printf( " \n %d is a prime number " ,n);

}

else

{

printf(" \n %d is not a prime number" , n);

}

getch();

}

**OUTPUT**:

enter the number to check prime or not

7

7 is a prime number

**2. ODD OR EVEN**

#include<stdio.h>

#include<conio.h>

void main()

{

int n;

clrscr();

printf("Enter the number to check odd or even \n ");

scanf("%d",&n);

if(n%2==0)

{

printf("%d is even number");

}

else

{

printf("%d is odd number");

}

getch();

}

**OUTPUT:**

Enter the number to check odd or even

3

3 is odd number

**3. DISPLAY THE GRADES**

#include<stdio.h>

#include<conio.h>

void main()

{

int marks;

clrscr();

printf("\n Enter the mark of student \n");

scanf("%d",&marks);

if(marks>=90)

{

printf("The student passed with A grade");

}

else if(marks<90 &&marks>=80)

{

printf("The student passed with B grade ");

}

else if(marks<80 &&marks>=60)

{

printf("The students passed with C grade");

}

else

{

printf("The student passed with D grade");

}

getch();

}

**OUTPUT**:

Enter the mark of student

90

The student passed with A grade

Enter the mark of student

80

The student passed with B grade

Enter the mark of student

60

The students passed with C grade

Enter the mark of student

50

The students passed with D grade

**4.CALCULATOR**

#include<stdio.h>

#include<conio.h>

void main()

{

int ch,a,b,c;

printf( " \n Enter the two operand");

scanf("%d", "%d", &a , &b);

printf( " \n Enter the operation to be performed \n 1.Addition \n 2.Subtraction \n 3.Multiplication \n 4.Divisiob \n");

scanf( "%d" , &ch);

switch(ch)

{

case 1:

c=a+b;

printf(" \n Addition of %d , %d is %d", a,b,c);

break;

case 2:

c=a-b;

printf(" \n Subtraction of %d , %d is %d", a,b,c);

break;

case 3:

c=a\*b;

printf(" \n Multiplication of %d , %d is %d", a,b,c);

break;

case 4:

c=a/b;

printf( "\n Division of %d , %d is %d", a,b,c);

break;

default:

printf("Operation does not exist");

break;

}

getch();

}

**OUTPUT:**

Enter the two operands

2 3

Enter the operation to be performed

1.Addition

2.Subtraction

3.Multiplication

4.Division

1

Addition of 2,3 is 5

Enter the two operands

2 3

Enter the operation to be performed

1.Addition

2.Subtraction

3.Multiplication

4.Division

2

Subtraction of 2, 3 is -1

Enter the two operands

2 3

Enter the operation to be performed

1.Addition

2.Subtraction

3.Multiplication

4.Division

3

Multiplication of 2,3 is 6

Enter the two operands

4 2

Enter the operation to be performed

1.Addition

2.Subtraction

3.Multiplication

4.Division

4

Division of 4,2 is 2

Enter the two operands

4 2

Enter the operation to be performed

1.Addition

2.Subtraction

3.Multiplication

4.Division

5

Operation does not exists

**5. GREAT COMMON DIVISOR**

#include<stdio.h>

#include<stdio.h>

int main()

{

int n1,n2;

clrscr();

{

printf("Enter the two positive integer");

scanf("%d %d",&n1,&n2);

while(n1!=n2)

{

if(n1>n2)

n1-=n2;

else

n2-=n1;

}

printf("GCD=%d",n1);

getch();

return 0;

}

}

**OUTPUT:**

Enter the two positive integer

4 5

GCD=1

**6. ROOTS OF QUADRATIC EQUATION**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int a,b,c,dis;

float r1,r2,real,img;

clrscr();

printf("Enter the coefficient a b c \n");

scanf("%d %d %d", &a,&b,&c);

dis=b\*b-4\*a\*c;

if(dis>0)

{

r1=(-b+sqrt(dis))/(2\*a);

r2=(-b-sqrt(dis))/(2\*a);

printf("Roots are %f \t %f",r1,r2);

}

else if (dis==0)

{

r1=r2=-b/(2\*a);

printf("Roots are equal that is %f",r1);

}

else

{

real=-b/(2\*a);

img=sqrt(-dis)/(2\*a);

printf("Roots are complex numbers %f + %f and %f - %f ",real,img,real,img);

}

getch();

}

**OUTPUT:**

Enter the coefficient a b c

1

7

10

Roots are -2.00 -5.00

Enter the coefficient a b c

1

6

9

Roots are equal that is -3.00

Enter the coefficient a b c

1

4

5

Roots are complex numbers -2.000000 +1.000000 and -2.000000 -1.000000

**7. FACTORIAL**

#include<stdio.h>

#include<conio.h>

int fact(int n);

void main()

{

int num;

clrscr();

printf("Enter the number factorial");

scanf("%d",&num);

printf("Factorial of given number is %d",fact(num));

getch();

}

int fact(int n)

{

if(n>1)

{

n=n\*fact(n-1);

return n;

}

else

{

return 1;

}

}

**OUTPUT:**

Enter the number factorial

5

Factorial of given number is 120

**8. SWAPPING**

#include<stdio.h>

#include<conio.h>

void swap(int\*x,int\*y);

void main()

{

int a,b;

clrscr();

printf("Enter the two value to swap \n ");

scanf("%d \t %d",&a,&b);

swap(&a,&b);

printf("\n After swaping %d \t %d ",a,b);

getch();

}

void swap(int\*x,int\*y)

{

int temp;

printf("\n Before swaping %d \t %d",\*x,\*y);

temp=\*x;

\*x=\*y;

\*y=temp;

}

**OUTPUT:**

Enter the two value to swap

5 4

Before swaping 5 4

After swaping 4 5

**9.MATRIX**

#include<stdio.h>

#include<conio.h>

int i, j , ans[5][5];

void add(int a[5][5] , int b[5][5] , int r , int c)

{

printf(" Addition of matrix A & B \n");

for(i=0; i<r;i++)

{

for( j=0; j<c;j++)

{

ans[i][j]=a[i][j]+b[i][j];

printf("%d \t", ans[i][j]);

}

printf("\n");

}

}

void sub( int a[5][5] , int b[5][5] , int r , int c)

{

printf( " Subtraction of matrix A & B \n");

for(i=0; i<r;i++)

{

for( j=0; j<c; j++)

{

ans[i][j]=a[i][j]-b[i][j];

printf("%d \t", ans[i][j]);

}

printf("\n");

}

}

void mul(int a[5][5] , int b[5][5] , int r , int c)

{

printf( " Multiplication of matrix A & B \n");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

ans[i][j]=a[i][j]\*b[i][j];

printf("%d \t", ans[i][j]);

}

printf("\n");

}

}

void main()

{

int r,c,ch,a[5][5],b[5][5];

clrscr();

printf( "\n Enter the number of row and column of matrix \n");

scanf( "%d %d" ,&r,&c);

printf( "\n Enter the value of matrix A \n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

scanf("%d",&a[i][j]);

}

}

printf( " \n Enter the value of matrix B \n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

scanf("%d",&b[i][j]);

}

}

printf(" \n 1. Addition \n 2.Subtraction \n 3. Multiplication \n Enter the option \n");

scanf( "%d" , &ch);

switch(ch)

{

case 1:

add(a,b,r,c);

break;

case 2:

sub(a,b,r,c);

break;

case 3:

mul(a,b,r,c);

break;

default:

printf(" Operation does not exist \n");

}

getch();

}

**OUTPUT:**

Enter the number of rows and columns of matrix

2

2

Enter the value of matrix A

2

2

2

2

Enter the value of matrix B

3

3

3

3

1. Addition

2.Subtraction

3. Multiplication

Enter the Option

1

Addition of Matrix A and B

5 5

5 5

Enter the number of rows and columns of matrix

2

2

Enter the value of matrix A

2

2

2

2

Enter the value of matrix B

3

3

3

3

1. Addition

2.Subtraction

3. Multiplication

Enter the option

2

Subraction of matrix A&B

-1 -1

-1 -1

Enter the number of row and column of matrix

2

2

Enter the value of matrix A

2

2

2

2

Enter the value of matrix B

3

3

3

3

1. Addition

2.Subtraction

3. Multiplication

Enter the option

3

Multiplication of matrix A & B

6 6

6 6

**10. AVERAGE AGE**

#include<stdio.h>

#include<conio.h>

int i;

void avg(int a[],int n)

{

int avg,sum=0;

for(i=0;i<n;i++)

{

sum=sum+a[i];

}

avg=(sum/n);

printf("Average age of persons %d",avg);

}

void main()

{

int a[10],n;

clrscr();

printf("Enter the number of person \n ");

scanf("%d", &n);

printf("Enter the age of each person \n ");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

avg(a,n);

getch();

}

**OUTPUT:**

Enter the number of persons

2

Enter the age of each person

20

40

Average age of persons 30

**Ex no: 11 LINEAR SEARCH OPERATIONS**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**void main()**

**{**

**int s,i,a[20],n;**

**clrscr();**

**printf("\n Enter the size oof an array \n");**

**scanf("%d",&s);**

**printf("\n Enter the value of for array \n");**

**for(i=0;i<s;i++)**

**{**

**scanf("%d", &a[i]);**

**}**

**printf("\nEnter the value to search in array ");**

**scanf("%d",&n);**

**for(i = 0;i<=s;i++)**

**{**

**if(a[i]==n)**

**{**

**printf("\n the value is found in %d position \n",i++);**

**break;**

**}**

**else if(i==s)**

**{**

**printf("\n the value %d is not found ",n);**

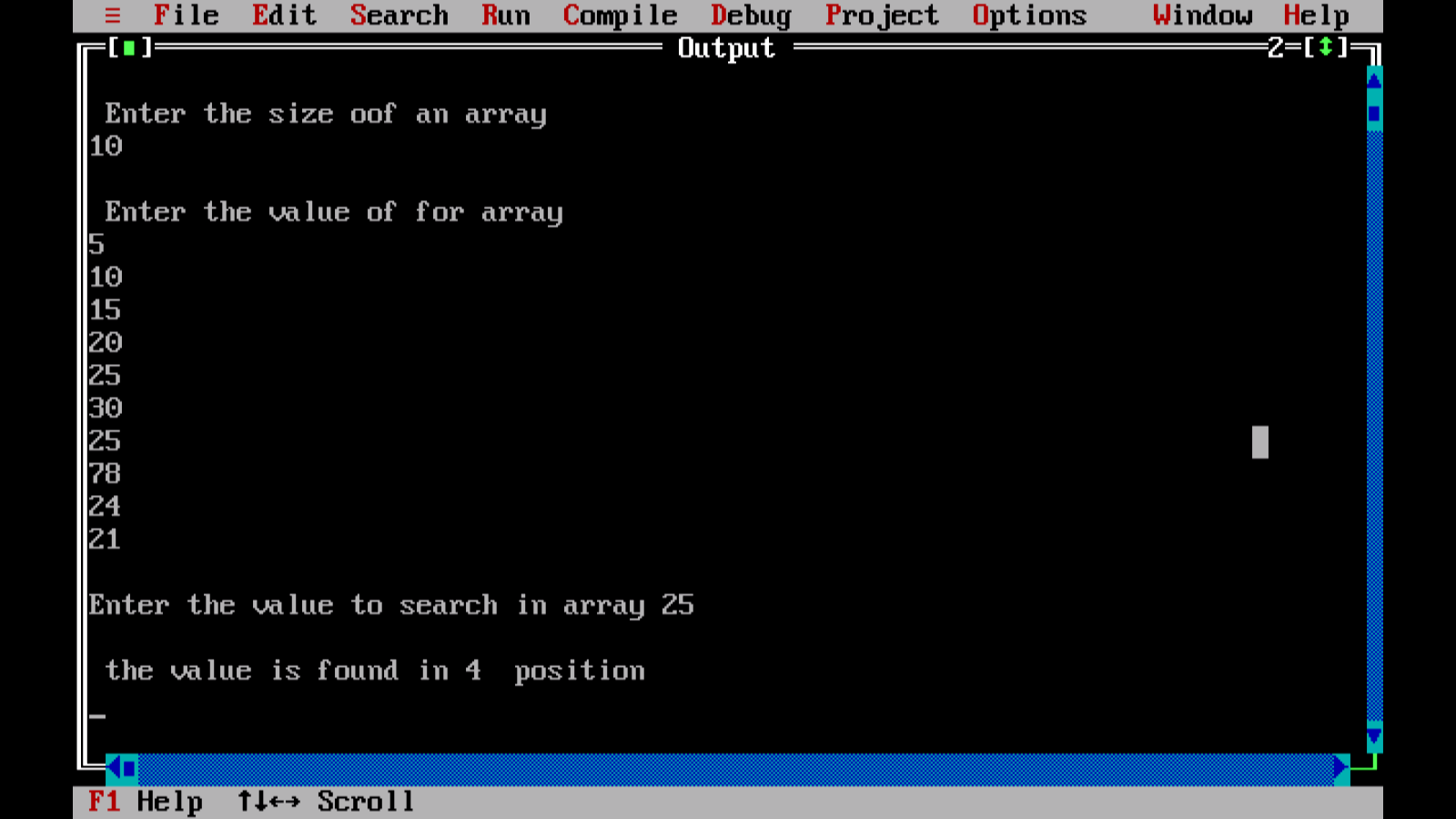
**}**

**getch();**

**}**

**}**

**OUTPUT:**

****

**Ex no: 12 RANK USING STRUCTURE**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**int i,j,n;**

**struct student**

**{**

**char name[20];**

**int s1,s2,s3,total;**

**};**

**struct student st[100],temp;**

**void mark()**

**{**

**for(i=0;i<n;i++)**

**{**

**printf("\n name of the student ");**

**scanf("%s",&st[i].name);**

**printf("marks of 3 subject \n");**

**scanf("%d %d %d",&st[i].s1,&st[i].s2,&st[i].s3);**

**st[i].total = st[i].s1+st[i].s2+st[i].s3;**

**}**

**}**

**void rank()**

**{**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<n;j++)**

**{**

**if(st[j].total < st[i].total){**

**temp = st[i];**

**st[i]=st[j];**

**st[j]=temp;**

**}**

**}**

**}**

**}**

**void display()**

**{**

**for(i=0;i<n;i++)**

**{**

**printf("\n rank: %d",i+1);**

**printf("\n name: %s",st[i].name);**

**printf("\n totalmarks: %d", st[i].total);**

**}**

**}**

**void main()**

**{**

**struct student st[100],temp;**

**clrscr();**

**printf("\n enter the number of students");**

**scanf("%d", &n);**

**mark();**

**rank();**

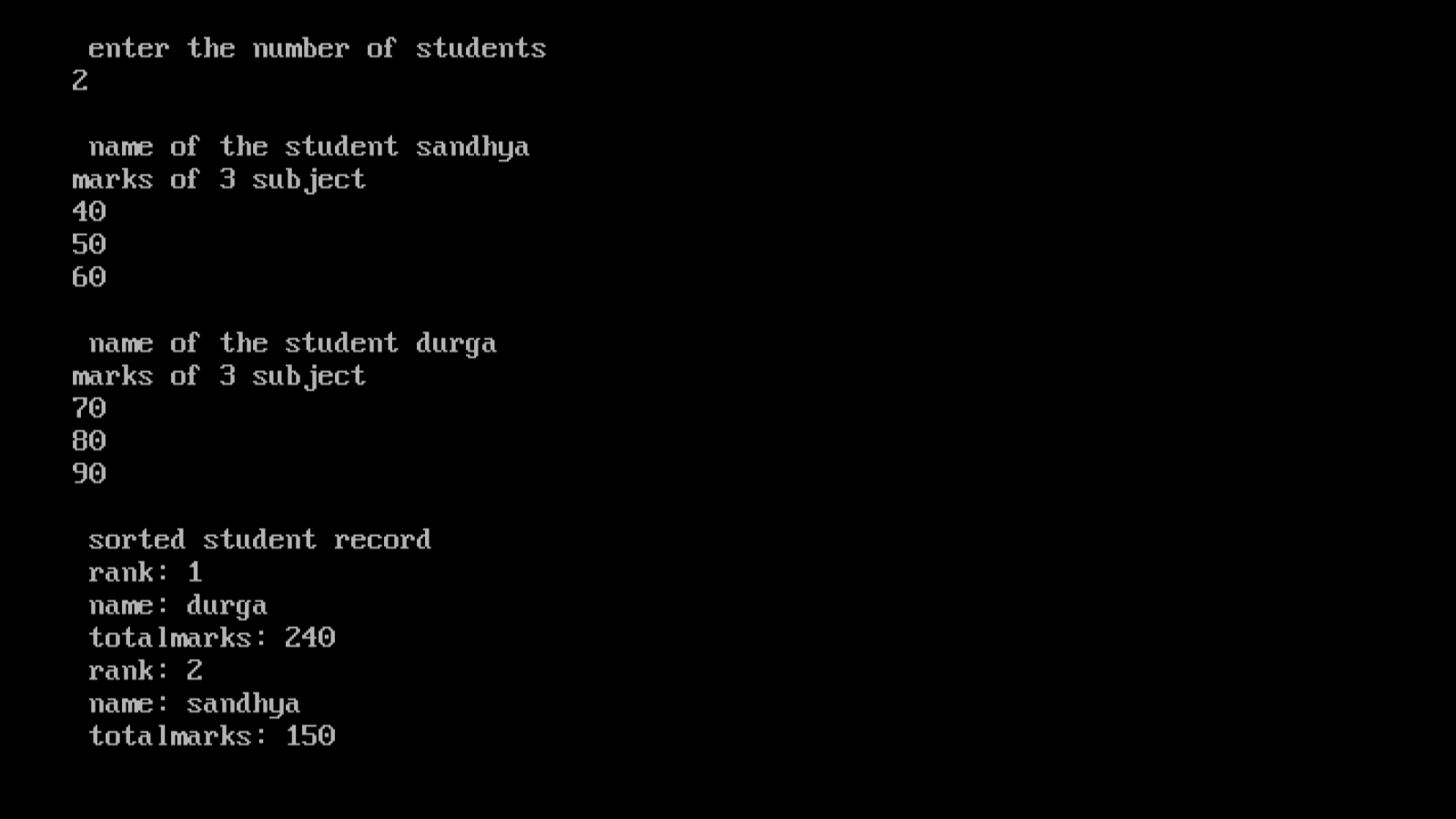
**printf("\n sorted student record");**

**display();**

**getch();**

**}**

**OUTPUT:**

****

**Ex no: 13 VOWELS**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<stdlib.h>**

**void main()**

**{**

**int c=0,count=0;**

**char s[1000];**

**clrscr();**

**printf("\n enter the string ");**

**gets(s);**

**while(s[c]!='\0')**

**{**

**if(s[c]=='a' ||s[c]=='A'||**

**s[c]=='e' ||s[c]=='E'||**

**s[c]=='i' ||s[c]=='I'||**

**s[c]=='o' ||s[c]=='O'||**

**s[c]=='u' ||s[c]=='U')**

**count++;**

**c++;**

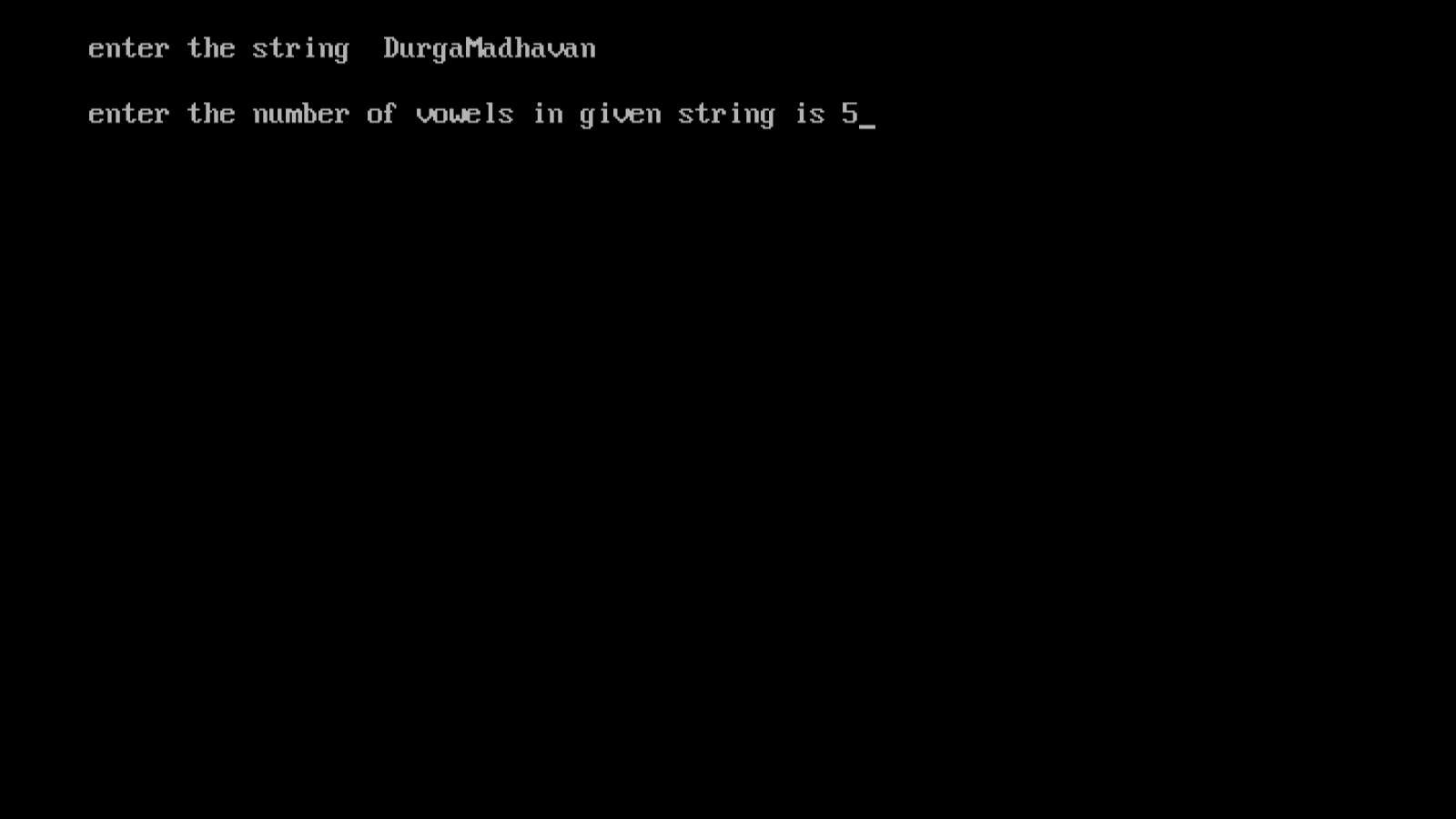
**}**

**printf("\n enter the number of vowels in given string is %d",count);**

**getch();**

**}**

**OUTPUT:**

****

**Ex no: 14 SUBSTRING**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<string.h>**

**void main()**

**{**

**char m[10],s[10],temp[15];**

**int ml=0,sl,i=0,bl,a,x,p=0,t=0;**

**clrscr();**

**puts("\n enter the main string\n");**

**gets(m);**

**puts("\n enter the sub string \n");**

**gets(s);**

**printf("\n enter the position where to be inserted " );**

**scanf("%d", &p);**

**ml = strlen(m);**

**sl= strlen(s);**

**i=0;**

**while(i<ml)**

**{**

**temp[i]=m[i];**

**i++;**

**}**

**bl=ml+sl;**

**a=p+sl;**

**for (i=p;i<bl;i++)**

**{**

**x=temp[i];**

**if(t<sl)**

**{**

**m[i]=s[t];**

**t=t+1;**

**}**

**m[a]=x;**

**a=a+1;**

**}**

**printf("%s",m);**

**getch();**

**}**

**OUTPUT:**

****

**EX NO:15 PALINDROME**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<string.h>**

**void main()**

**{**

**char s[1000];**

**int i,n,c=0;**

**clrscr();**

**printf("\n enter the string ");**

**gets(s);**

**n=strlen(s);**

**for(i=0;i<n/2;i++)**

**{**

**if(s[i] == s[n-i-1])**

**{**

**c++;**

**}**

**}**

**if(c==i)**

**{**

**printf("\n %s is a palindrome ",s);**

**}**

**else**

**{**

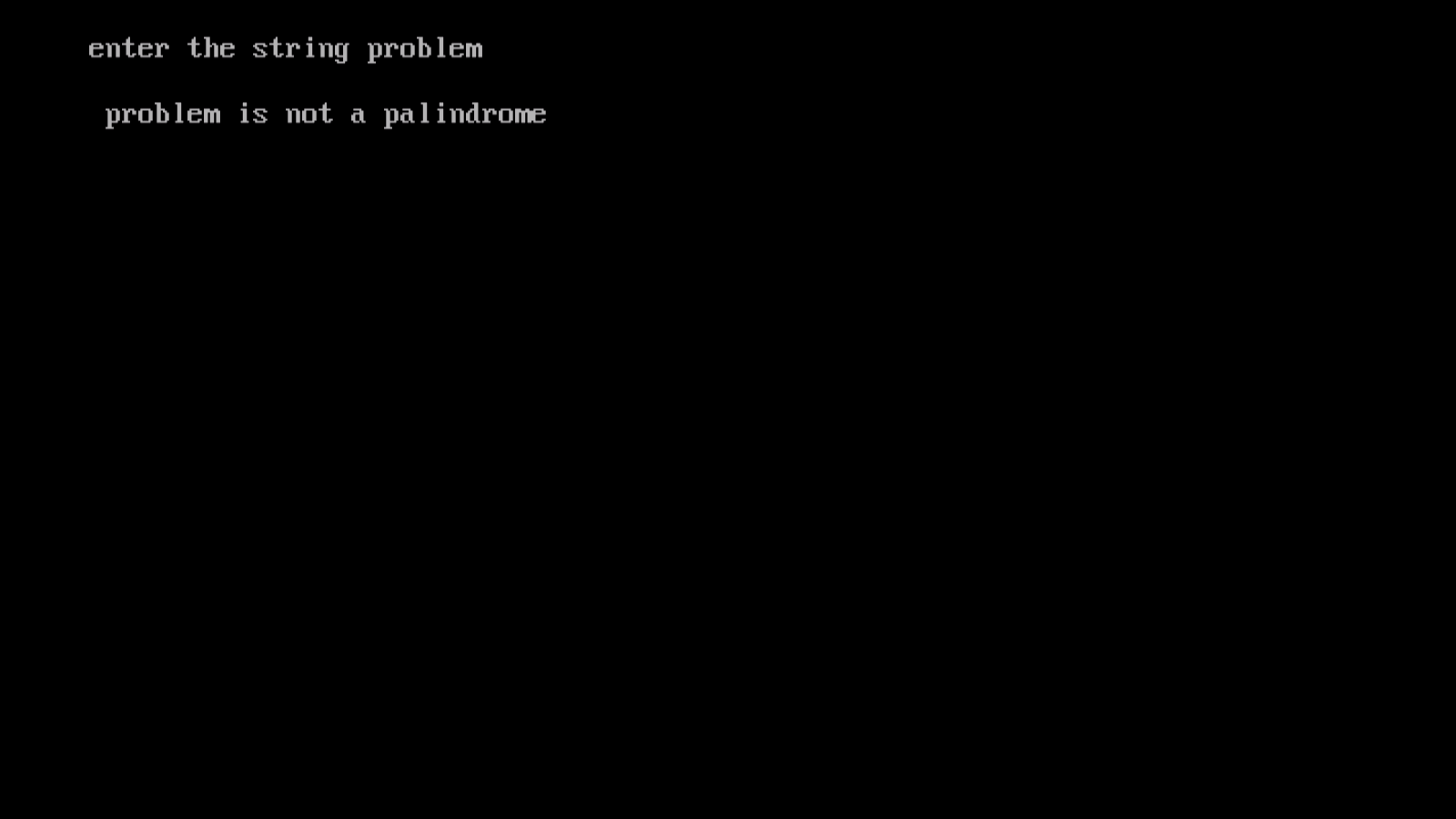
**printf("\n %s is not a palindrome ",s);**

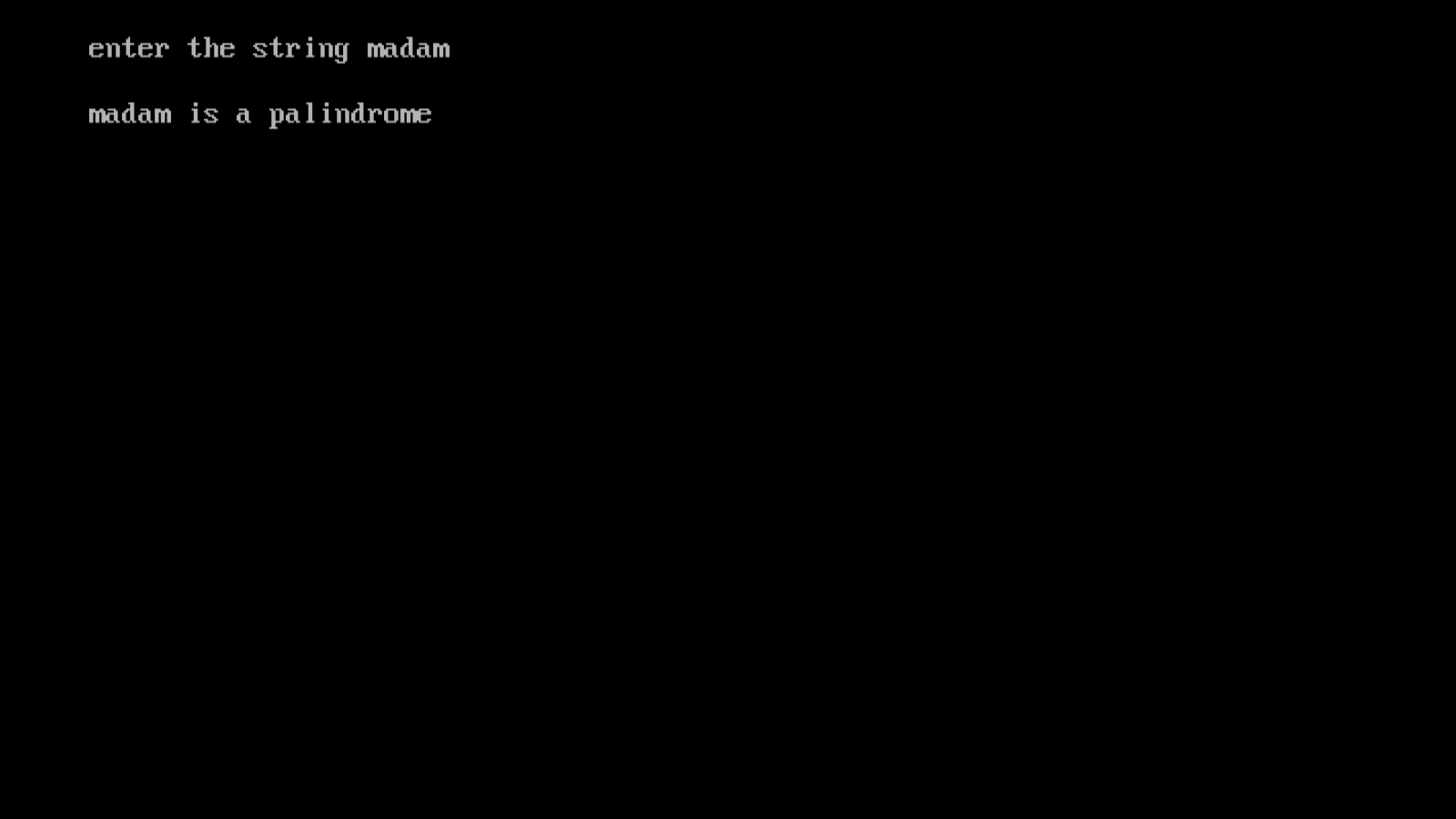
**}**

**getch();**

**}**

**OUTPUT:**

****

****

**Ex no: 16 WRITE IN A FILE**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<stdlib.h>**

**void main()**

**{**

**char c[1000];**

**FILE \*fptr;**

**clrscr();**

**fptr=fopen("FILE1.txt","w");**

**if(fptr == NULL)**

**{**

**printf("error!");**

**exit(1);**

**}**

**printf("\nenter a sentence\n");**

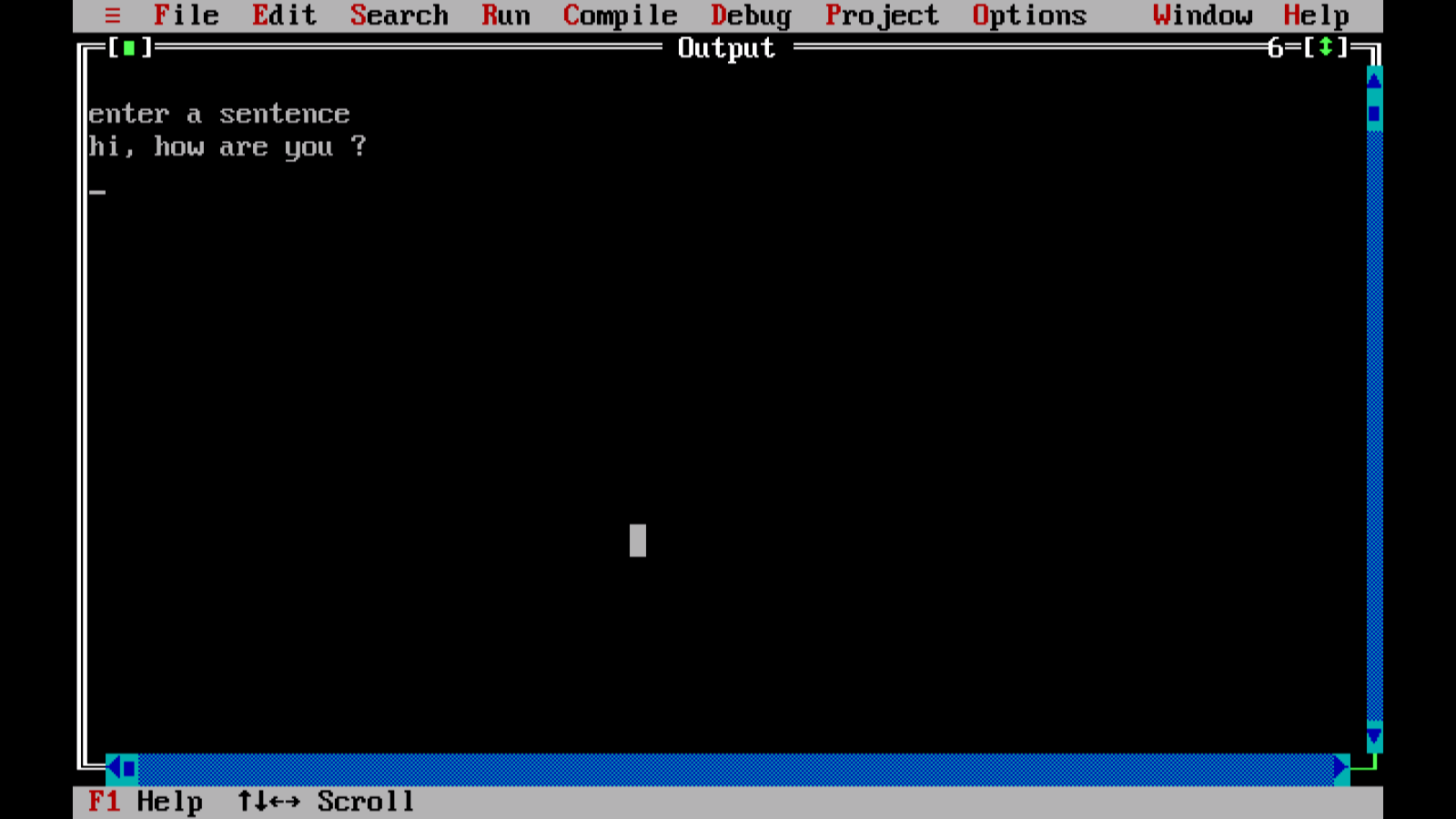
**gets(c);**

**fprintf(fptr, "%s",c);**

**fclose(fptr);**

**}**

**OUTPUT:**

****

****

**EX NO:17 APPEND A FILE**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<stdlib.h>**

**void main()**

**{**

**char name[20];**

**int marks,i,n;**

**FILE \*fptr;**

**clrscr();**

**printf("\enter the number of students\n");**

**scanf("%d",&n);**

**fptr=fopen("FILE2.txt","a");**

**if(fptr == NULL)**

**{**

**printf("error");**

**exit(1);**

**}**

**for(i=0;i<n;i++)**

**{**

**printf("\n enter the name :");**

**scanf("%s",&name);**

**printf("\nmarks:");**

**scanf("%d",&marks);**

**fprintf(fptr,"\nName\t %s \nMarks\t %d \n",name,marks);**

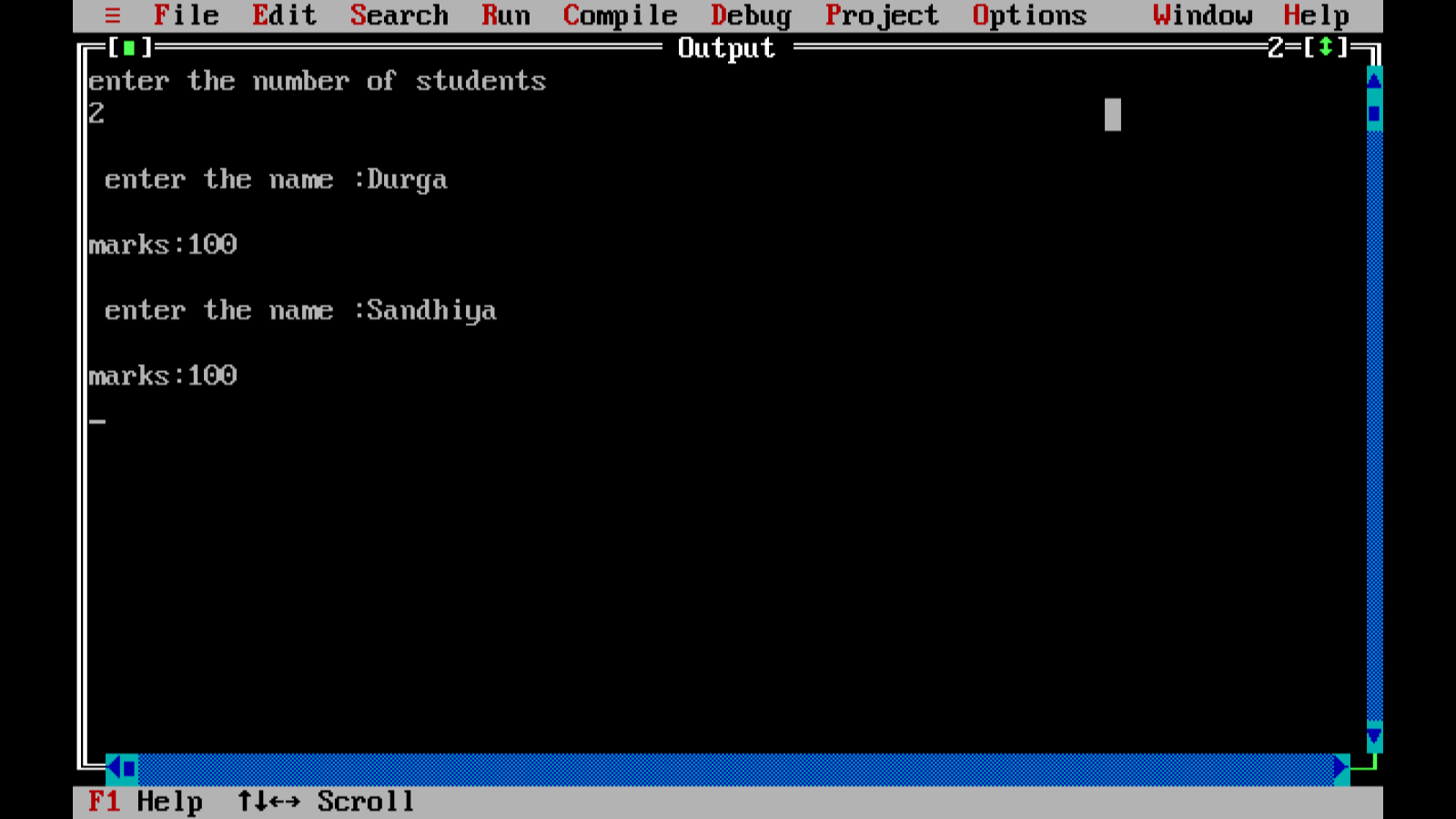
**}**

**getch();**

**fclose(fptr);**

**}**

**OUTPUT:**

****

****

**EX NO:18 PROGRAM USING FWRITE()**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<stdlib.h>**

**struct s**

**{**

**char name[20];**

**int height;**

**};**

**int main()**

**{**

**struct s a[5],b[5];**

**FILE \*fptr;**

**int i;**

**clrscr();**

**fptr = fopen("fwrite.txt","wb");**

**for(i=0;i<2;i++)**

**{**

**fflush(stdin);**

**printf("enter the name ");**

**gets(a[i].name);**

**printf("enter height \t");**

**scanf("%d",&a[i].height);**

**}**

**fwrite(a,sizeof(a),1,fptr);**

**fclose(fptr);**

**fptr=fopen("fwrite.txt","rb");**

**fread(b,sizeof(b),1,fptr);**

**for(i=0;i<2;i++)**

**{**

**printf("\n name: %s\n height:%d",b[i].name,b[i].height);**

**}**

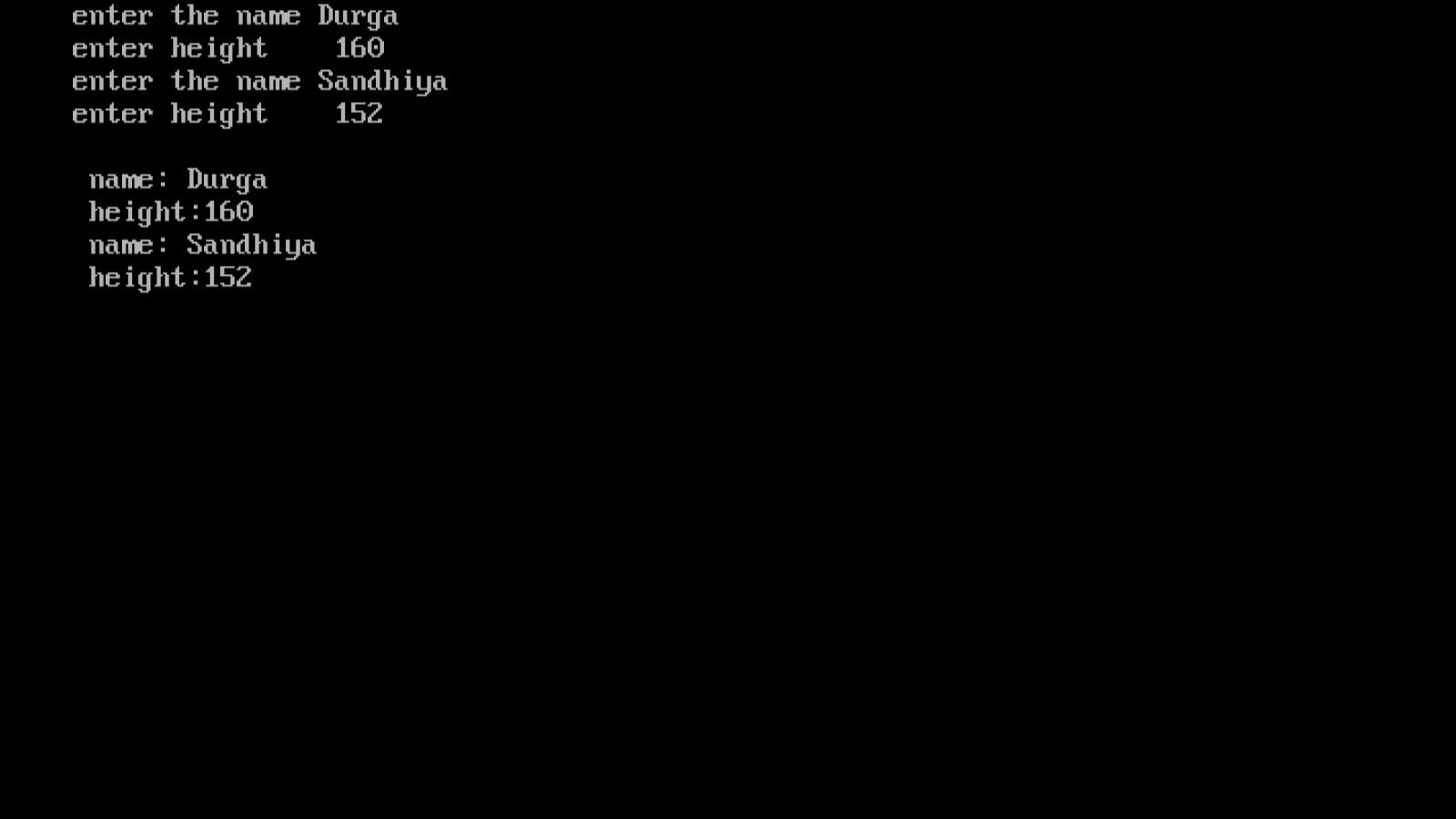
**getch();**

**fclose(fptr);**

**return 0;**

**}**

**OUTPUT:**

****

****

**EXNO: 19 AREA OF CIRCLE**

**PROGRAM CODING :**

**#include<stdio.h>**

**#include<conio.h>**

**#define PI 3.145**

**#define area(r) (PI\*r\*r)**

**int main()**

**{**

**int radius;**

**float area;**

**clrscr();**

**printf("\n enter the radius");**

**scanf("%d", &radius);**

**area= area(radius);**

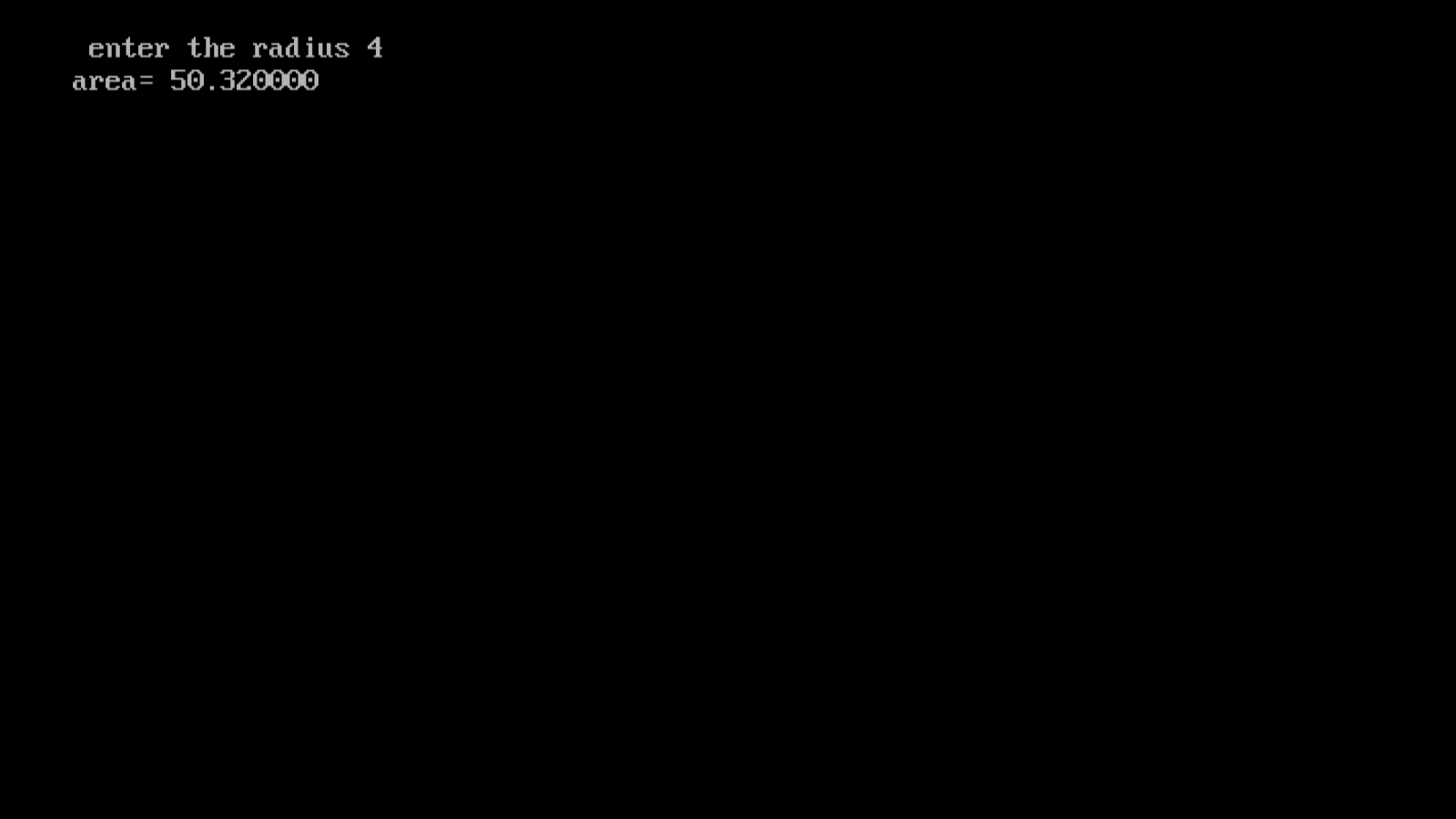
**printf("area= %f",area);**

**getch();**

**return 0;**

**}**

**OUTPUT:**

****

**EX NO:20 SUM AND PRODUCT**

**PROGRAM CODING:**

**#include<stdio.h>**

**#include<conio.h>**

**#define sum(A,B) (A+B)**

**#define prod(A,B) (A \* B)**

**int main()**

**{**

**int n1,n2,s,p;**

**clrscr();**

**printf("\n enter the 2 nos ");**

**scanf("%d %d", &n1,&n2);**

**s=sum(n1,n2);**

**p=prod(n1,n2);**

**printf("\n sum of 2 nos is %d \n ",s);**

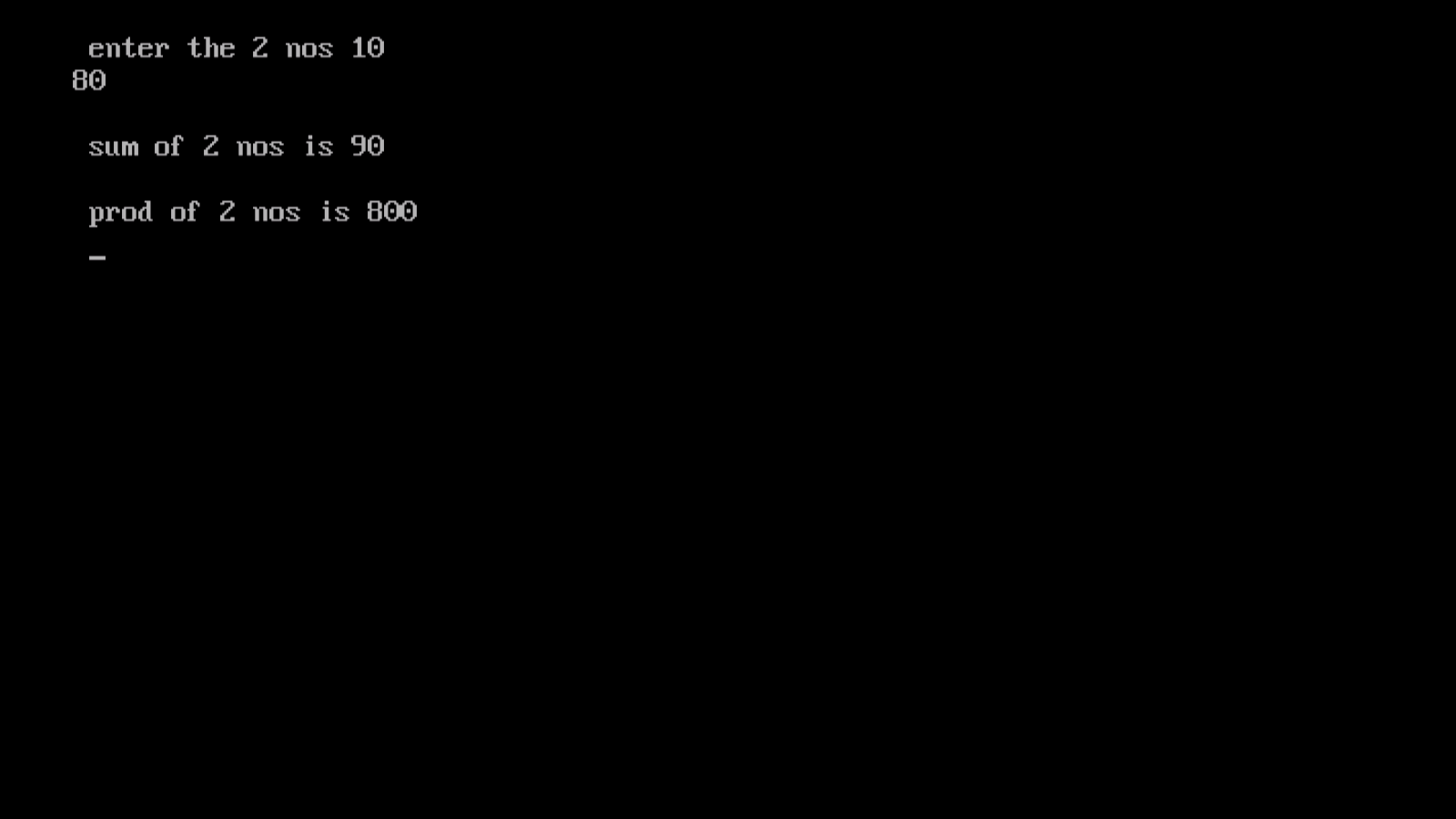
**printf("\n prod of 2 nos is %d \n ",p);**

**getch();**

**return 0;**

**}**

**OUTPUT:**

****