**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE** |
|  | C++ REVISION TOUR |  |
| **1.** | **An array stores details of 25 students( rollno, name, marks in three subjects). Write a program to create such an array and (i) print out a list of students who have failed in more than one subject. Assume 40% as pass marks(ii) sort the array in ascending order based on total marks.** |  |
| **2.** | **Declare two structures one called employee: Name, address, phone number, salary and the second is called perks: Da =20% of salary, Hra=25% of salary, Net =salary+da+hra. Write a program to calculate net salary of an employee.** |  |
| **3.** | **Details of 50 clients of an investment company are stored in an array of structures. Details include customers name, code, date of starting, number of years, interest rate and total amount. Write a program to calculate compound interest for these clients** |  |
| **4.** | **Write a program in C++ using function to display those elements of a two dimensional array M[5][5] which are divisible by 10. Assume the content of the array is already present and the function prototype is as follows: - void display10 (int M [5][5]).** |  |
| **5.** | **Declare a structure distance having feet and inches. Write a C++ program to add two distances.** |  |
| **6.** | **A linear array of size 10 stores following information: name of the country, country’s capital and per capita income of the country. Write a complete program in C++ to do the following:**  **a) To read a country’s name and display capital and per­capita income.**  **b) To read name of the capital city and display country’s name and per capital income. Display an error message in case of an incorrect input.** |  |
|  | CLASSES AND OBJECTS |  |
|  | **Define a class student with the following data members:**  **Admno integer, sname 20 character, eng, maths, science float, total float, ctotal() a function to calculate eng + maths + science**  **Public member function of class student, Takedata() function to accept values for admno, sname, eng, maths, science and invoke ctotal() to calculate total,**  **Showdata() function to display all the data members on the screen.**  **Write a C++ program to create a class and invoke all the member functions.** |  |
|  | **Define a class worker with the following specification**  **Roll\_no integer, wno integer, wname 25 character, hrwrk float, wgrate float, totwage float and function calcwg() to find totwage=hrwrk\*wgrate with float return type.**  **Public member - In\_data() a function to accept values of wno, wname, hrwrk, wgrate and invoke calcwg() to calculate totpay.**  **Out\_data() a function to display all the data members on the screen you should give definitions of functions**  **Write a C++ program to create a class and invoke all the member functions.** |  |
|  | **A class CLOCK has the following members:**  **Data member: hour of type integer, minute of type integer**  **Member functions: readtime(int h, int m); showtime() to display data member, addtime(time T1, time T2).**  **Write a program to input two different objects FT and ST, print their sum (assume 24 hr. clock time)**  **e.g. input FT=6 hrs. 35mins, ST=3hrs 45 min then output T=FT+ST=10hrs 20min** |  |
|  | **A class serial has the following data members**  **Scode integer, title 20 character, duration float, noofepisodes integer**  **and members functions**  **init() to initialize duration as 30 and noofepisodes as 10**  **Newserial() function to accept values for serialcode and title**  **Otherentries() function to assign the values of all data members with the help of corresponding values passed as parameters to this function.**  **Dispdata() function to display all the data members on the screen.**  **Write a C++ program to create a class and invoke all the member functions.** |  |
|  | **A class student has three data members name, roll number, marks of 5 subjects and two member functions to accept data and to assign streams on the basis of table given below. Develop a C++ program to accept the data and print the stream:**  **Average Marks Stream**  **96% or more Computer Science**  **91% - 95% Electronics**  **86% to 90% Mechanical**  **81% to 85% Electrical**  **76% to 80% Chemical**  **71% to 75% Civil** |  |
|  | **CONSTRUCTORS AND DESTRUCTORS** |  |
|  | **Write C++ program to generate Fibonacci series using a class fib, which have the following data members and member functions:**  **Data members: first, second (i.e first 2 terms of the series) and n(no. of terms)**  **Constructor to initialize first, second and n**  **Parameterised Constructor to take the value of first, second and n from the user.**  **Function gen\_fib() to generate the series.** |  |
|  | **Write a C++ program to evaluate ab, by creating a class power where a and b are integer variables**  **Constructor to initialize a and b**  **Parameterized constructor to accept the values**  **Function disp() to display the result values** |  |
|  | **Write a C++ program to evaluate ab, by creating a class power where a and b are integer variables**  **Constructor to initialize a and b**  **Parameterized constructor to accept the values**  **Function disp() to display the result values** |  |
| **4.** | **Develop a program with the given fields and function:**  **Display a class play in C++ with the following**   * **Playcode integer** * **Playtitle 25 character** * **Duration float** * **Noofscenes integer**   **Public member function of class play**   * **A constructor function to initialise duration as 45 and Noofscenes as 5.** * **Newplay() function to accept values for Playcode and Playtitle.** * **Moreinfo() function to assign the values of duration and Noofscenes with the help of corresponding values passed as parameters to this function.** * **Showplay() function to display all the data members on the screen.** |  |
| **5.** | **Develop a program with the given fields and function:**  **Create a class box whose constructor function passes three values, each of which represents the length of one side of a box. From the box class compute the volume of the box and store the result in a double variable. Include a member function called vol () that displays the volume of each box object.** |  |
|  | **INHERITANCE** |  |
|  | **A publisher company markets both books and CDs. Create a class publication that stores the name (string) and price (float) of books and CDs. From this class derive two classes book which adds a page counts (type int), and CD which adds bytes (type int). Each of these classes should have a function getdata() to get data from the user and a function putdata() to display its data. Write a main() function to test the classes book and CD by creating instances of them, asking the user to input their data using the function getdata(), and then displaying the data with the function putdata().** |  |
|  | **A publisher company markets both books and CDs. Create a class publication that stores the name (string) and price (float) of books and CDs. From this class derive two classes book which adds a page counts (type int), and CD which adds bytes (type int). Each of these classes should have a function getdata() to get data from the user and a function putdata() to display its data. Write a main() function to test the classes book and CD by creating instances of them, asking the user to input their data using the function getdata(), and then displaying the data with the function putdata().** |  |
|  | **Write a program that reads the data of a student and computes its grade using single inheritance.** |  |
|  | **A college maintains a list of its students graduating every year. At the end of the year, the college produces a report that lists the following:**  **Number of working Graduates:**  **Number of non-working graduates:**  **Name:**  **Age:**  **Subject:**  **Average Marks:**  **X % of the graduates this year are non-working and n% are first divisioners.**  **Write a C++ program for it that uses the following inheritance path:**  **Person  Student  Graduate**  **(name, age) (rollno, avg marks) (student, employed)** |  |
|  | **Write a C++ program to read and display information about employee and managers. Employee is a class that contains employee number, name, address and department. Manager class contains all information of the employee class and a list of employees working under a manager.** |  |
|  | **FILE HANDLING** |  |
|  | **Write a program in C++ to count the number of uppercase alphabets and number of vowels in a text file “abc.txt”.** |  |
|  | **Write a C++ program to read and write structure emp(eno, ename, edesig, esal) using read() and write() function in a binary file.** |  |
|  | **Write a program to delete the record from file having records maintained through classes.** |  |
|  | **Write a program to search a record based on rollno in a file that has records maintained through class (rollno, name, marks, average and grade) and member function to assign grade on the basis of table given below:**  **Average Marks Grade**  **90% or more A1**  **89% - 80% A2**  **70% to 70% B1**  **69% to 60% B2**  **59% to 50% C1**  **59% to 40% C2**  **Below 40% FAIL** |  |
| **5.** | **Write a program to append data in a file having records maintained through classes(rollno, name, marks, average and grade) and member function to assign grade on the basis of table given below:**  **Average Marks Grade**  **90% or more A1**  **89% - 80% A2**  **70% to 70% B1**  **69% to 60% B2**  **59% to 50% C1**  **59% to 40% C2**  **Below 40%**  **FAIL** |  |
| **6.** | **Write a program to copy all the lines that do not begin with a capital letter to a new file “ABC.txt” from "XYZ.txt"** |  |

**PROGRAMS**

**C++ REVISION TOUR**

1. **An array stores details of 25 students ( rollno, name, marks in three subjects). Write a program to create such an array and (i) print out a list of students who have failed in more than one subject. Assume 40% as pass marks(ii) sort the array in ascending order based on total marks.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

struct student

{

int rollno;

char name[20];

int marks[3];

int ctr=0;

int sums;

} stud[3], temp;

void main()

{

clrscr();

int i, j;

float sum=0;

cout<<"\nEnter students' details: ";

for(i=0; i<3; i++)

{

cout<<"\nStudent name: ";

cin>>stud[i].name;

cout<<"\nStudent roll no.: ";

cin>>stud[i].rollno;

cout<<"\nStudent marks (out of 100): ";

for(j=0; j<3; j++)

{

cout<<"\n\tMarks "<<j+1<<": ";

cin>>stud[i].marks[j];

sum+=stud[i].marks[j];

}

stud[i].sums=sum;

sum=0;

}

for(i=0; i<3; i++)

{

for(j=0; j<3; j++)

{

if(stud[i].marks[j]<40)

{

stud[i].ctr++;

}

}

if(stud[i].ctr>=2)

{

puts(stud[i].name);

cout<<endl;

}

stud[i].ctr=0;

}

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

{

for(int k=i+1; k<3; k++)

{

if(stud[i].sums>stud[k].sums)

{

temp=stud[i];

stud[i]=stud[k];

stud[k]=temp;

}

}

}

cout<<"\nSORTED ARRAY: ";

for(i=0; i<3; i++)

{

cout<<"\nName: ";

cout<<stud[i].name;

cout<<"\nRoll No.: ";

cout<<stud[i].rollno;

cout<<"\nTotal marks: ";

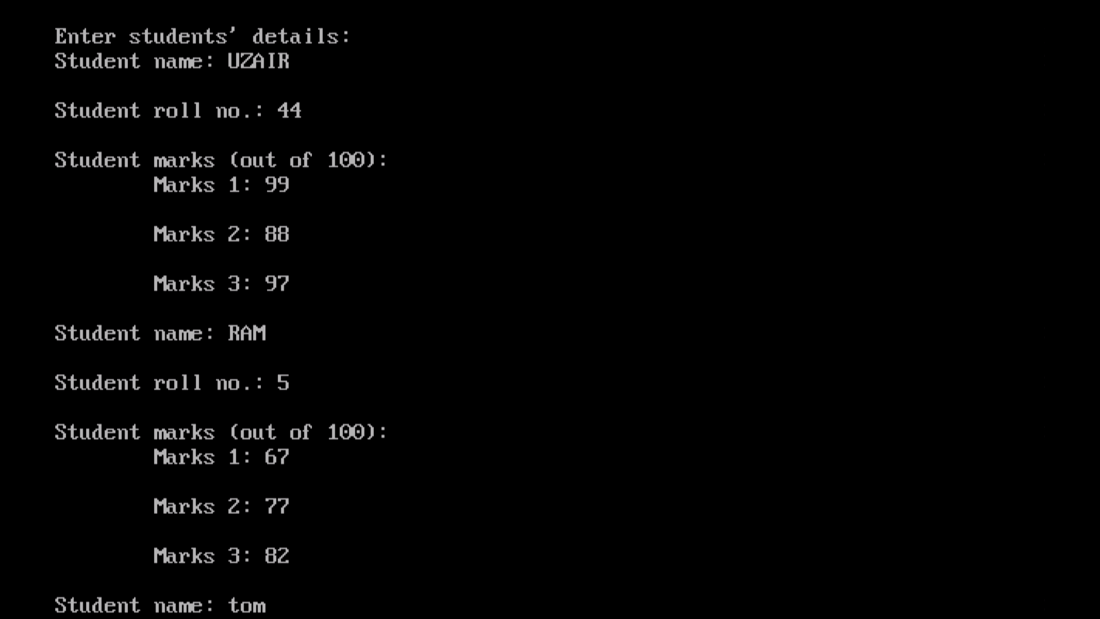
cout<<stud[i].sums;

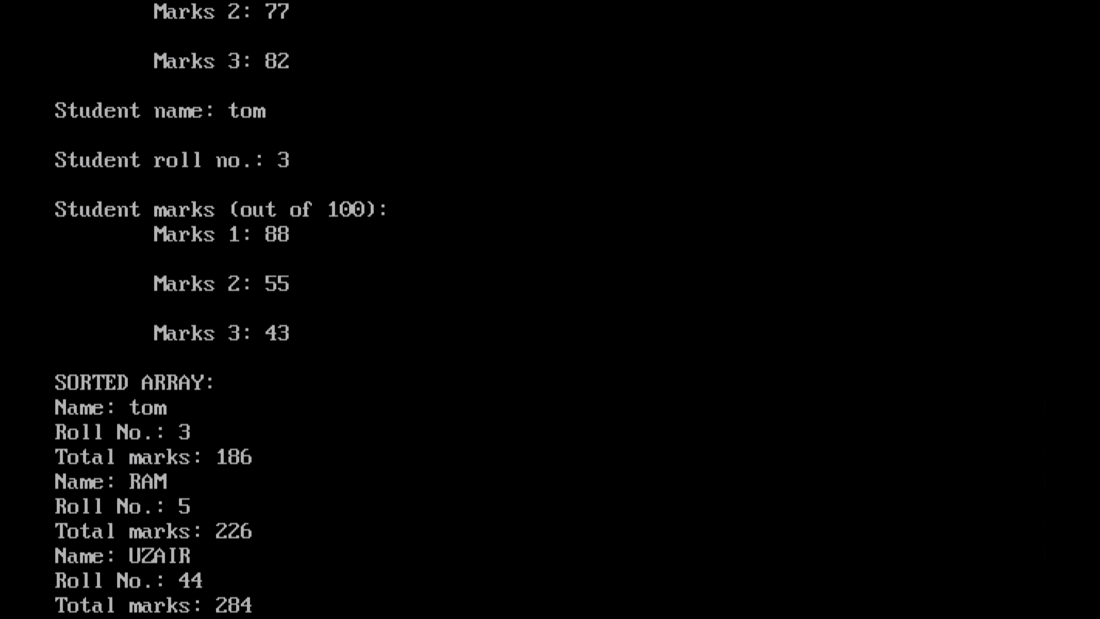
}

getch();

}

**OUTPUT:**





1. **Declare two structures one called employee: Name, address, phone number, salary and the second is called perks: Da =20% of salary, Hra=25% of salary, Net =salary+da+hra. Write a program to calculate net salary of an employee.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

struct employee

{

char name[30];

char address[30];

long int phno;

long int sal;

} e;

struct perks

{

long int da;

long int hra;

long int net;

} p;

void main()

{

clrscr();

cout<<"\nEMPLOYEE DETAILS";

cout<<"\nEnter employee name: ";

gets(e.name);

cout<<"\nEnter employee address: ";

gets(e.address);

cout<<"\nEnter phone no.: ";

cin>>e.phno;

cout<<"\nEnter salary: ";

cin>>e.sal;

p.da=0.2\*e.sal;

p.hra=0.25\*e.sal;

p.net=p.da+p.hra+e.sal;

cout<<"\nDaily Allowance: ";

cout<<p.da;

cout<<"\nHouse Rent Allowance: ";

cout<<p.hra;

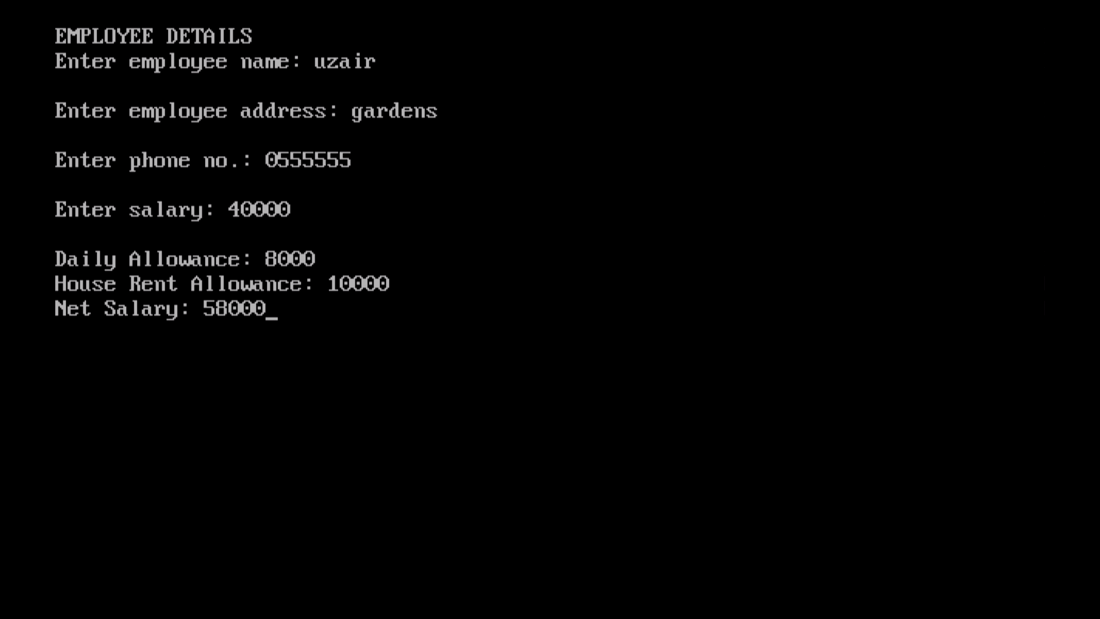
cout<<"\nNet Salary: ";

cout<<p.net;

getch();

}

**OUTPUT:**



1. **Details of 50 clients of an investment company are stored in an array of structures. Details include customers name, code, date of starting, number of years, interest rate and total amount. Write a program to calculate compound interest for these clients**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<math.h>

#include<stdio.h>

struct client

{

char cname[30];

int code;

int dos;

int years;

float intrate;

long int princ;

double total;

} c[3];

void main()

{

clrscr();

double comp;

cout<<"\nClient Details: ";

for(int i=0; i<3; i++)

{

cout<<"\nEnter client name: ";

cin>>c[i].cname;

cout<<"\nEnter client code: ";

cin>>c[i].code;

cout<<"\nEnter date of starting: ";

cin>>c[i].dos;

cout<<"\nEnter initial amount: ";

cin>>c[i].princ;

cout<<"\Enter no. of years: ";

cin>>c[i].years;

cout<<"\nEnter interest rate: ";

cin>>c[i].intrate;

c[i].total=c[i].princ\*pow((1+(c[i].intrate/100)),c[i].years);

comp=c[i].total-c[i].princ;

}

for(i=0; i<3; i++)

{

cout<<"\nEmployee Name: ";

cout<<c[i].cname;

cout<<"\nAmount after compounded interest: ";

cout<<c[i].total;

cout<<"\nCompound Interest: ";

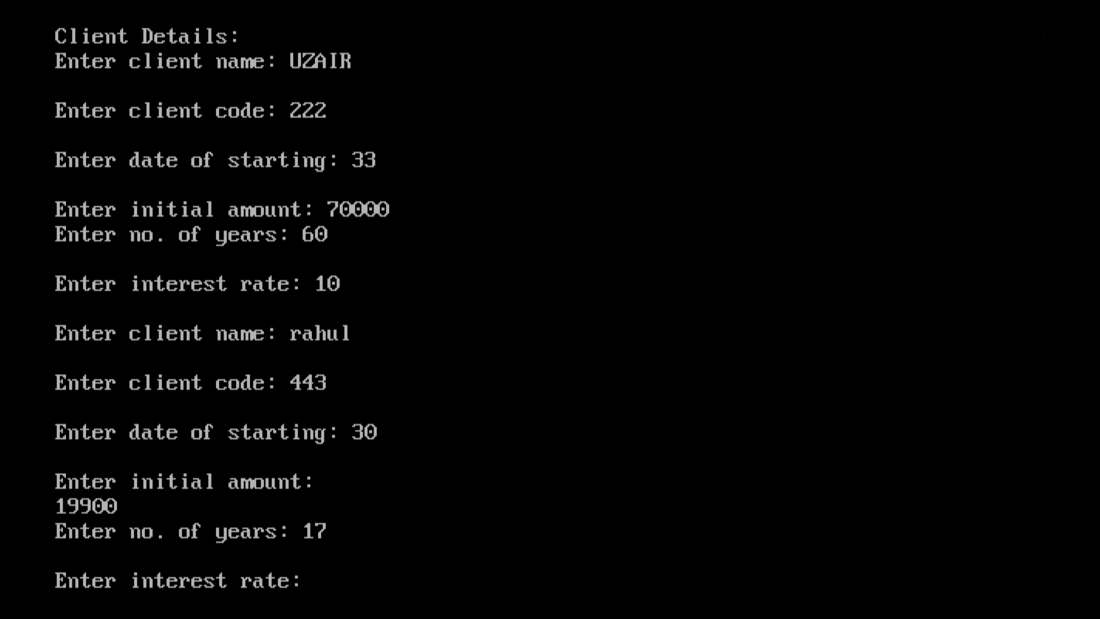
cout<<comp;

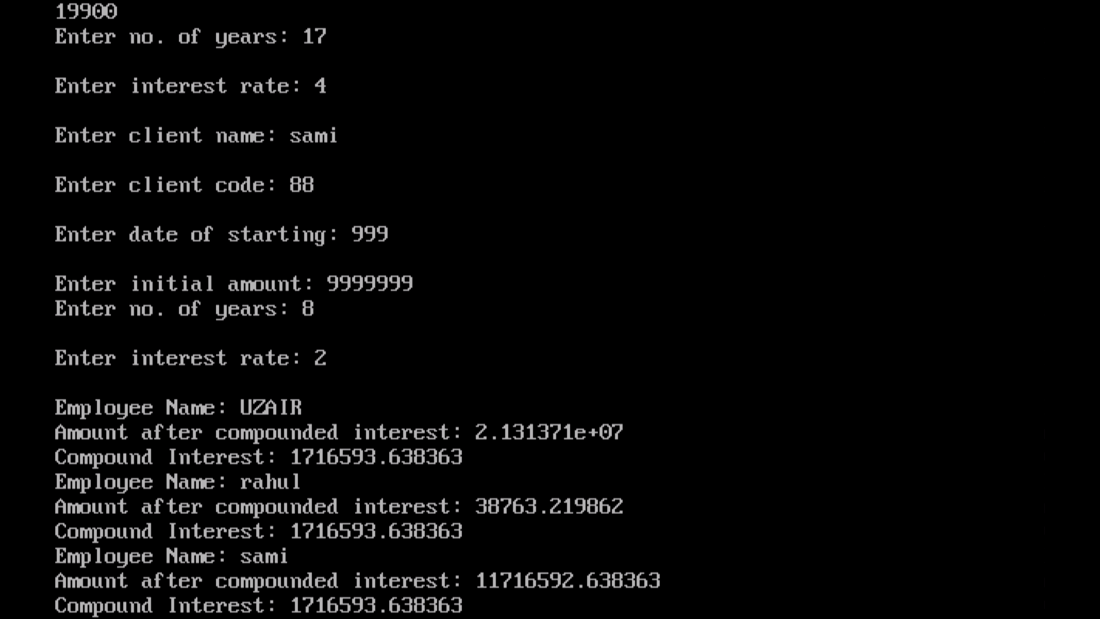
}

getch();

}

**OUTPUT:**





1. **Write a program in C++ using function to display those elements of a two dimensional array M[5][5] which are divisible by 10. Assume the content of the array is already present and the function prototype is as follows: - void display10 (int M [5][5]).**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

void display10(int [5][5]);

void main()

{

clrscr();

int i, j;

int arr[5][5];

for(i=0; i<5; i++)

{

for(j=0; j<5; j++)

{

cout<<"Enter element "<<i+1<<","<<j+1<<": ";

cin>>arr[i][j];

}

cout<<"\n";

}

display10(arr);

getche();

}

void display10(int M[5][5])

{

int x, y;

for(x=0; x<5; x++)

{

for(y=0; y<5; y++)

{

if(M[x][y]%10==0)

{

cout<<M[x][y];

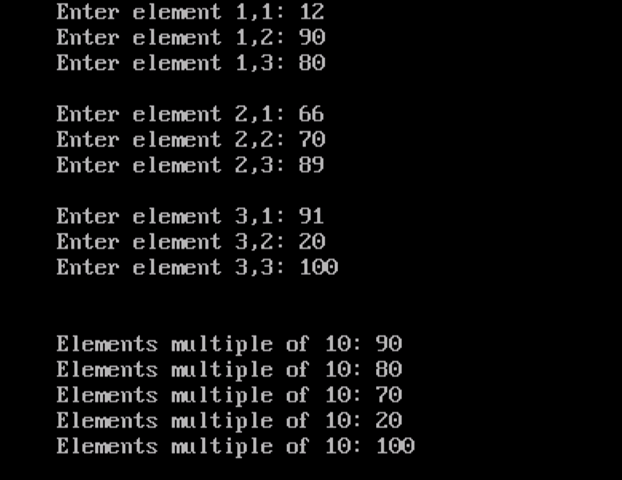
}

}

}

}

**OUTPUT:**



1. **Declare a structure distance having feet and inches. Write a C++ program to add two distances.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

struct distance

{

int feet;

int inches;

};

void main()

{

clrscr();

distance d[2];

distance tot\_dist;

tot\_dist.feet=0;

tot\_dist.inches=0;

int i;

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

{

cout<<"Enter distance in feet: ";

cin>>d[i].feet;

cout<<"\nEnter distance in inches: ";

cin>>d[i].inches;

tot\_dist.feet+=d[i].feet;

tot\_dist.inches+=d[i].inches;

}

if(tot\_dist.inches>=12)

{

tot\_dist.feet+=(tot\_dist.inches/12);

tot\_dist.inches=tot\_dist.inches%12;

}

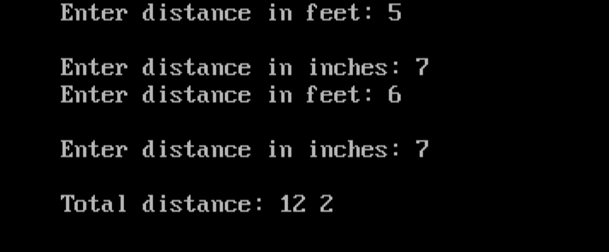
cout<<"\nTotal distance: ";

cout<<tot\_dist.feet<<" "<<tot\_dist.inches;

getch();

}

**OUTPUT:**



1. **A linear array of size 10 stores following information: name of the country, country’s capital and per capita income of the country. Write a complete program in C++ to do the following:**

**a) To read a country’s name and display capital and per-capita income.**

**b) To read name of the capital city and display country’s name and per capital income. Display an error message in case of an incorrect input.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<string.h>

#include<stdio.h>

struct country

{

char cntryname[30];

char cntrycapt[30];

long int captinc;

};

void main()

{

clrscr();

country c[3];

int i, ctr=0, a;

char captname[30];

cout<<"\nInput Country Details: ";

for(i=0; i<3; i++)

{

cout<<"\nEnter country name: ";

gets(c[i].cntryname);

cout<<"\nEnter country capital: ";

gets(c[i].cntrycapt);

cout<<"\nEnter per-capita income: ";

cin>>c[i].captinc;

}

cout<<"\n\t\t\t\tSEARCH RECORDS ";

cout<<"\n1. Search record by country name";

cout<<"\n2. Search record by country's capital";

cout<<"\nEnter here: ";

cin>>a;

switch(a)

{

case 1:

{

char contname[30];

cout<<"\nEnter country name's record to be searched: ";

gets(contname);

for(i=0; i<3; i++)

{

if(strcmp(c[i].cntryname, contname)==0 )

{

ctr=1;

cout<<"\nCountry Capital: ";

puts(c[i].cntrycapt);

cout<<"\nPer-Capita Income: ";

cout<<c[i].captinc;

break;

}

}

if(ctr==0)

{

cout<<"\nRecord Not Found";

}

break;

}

case 2:

{

cout<<"\nEnter capital city's record to be searched: ";

gets(captname);

for(i=0; i<3; i++)

{

if(strcmp(c[i].cntrycapt, captname)==0)

{

ctr=1;

cout<<"\nCountry Name: ";

puts(c[i].cntryname);

cout<<"\nPer-Capita Income: ";

cout<<c[i].captinc;

break;

}

}

if(ctr==0)

{

cout<<"\nRecord Not Found";

}

break;

}

default:

{

cout<<"\nInvalid Entry";

break;

}

}

getch();

}

**OUTPUT:**





**CLASSES AND OBJECTS**

1. **Define a class student with the following data members:**

**Admno integer, sname 20 character, eng, maths, science float, total float, ctotal() a function to calculate eng + maths + science**

**Public member function of class student, Takedata() function to accept values for admno, sname, eng, maths, science and invoke ctotal() to calculate total,**

**Showdata() function to display all the data members on the screen.**

**Write a C++ program to create a class and invoke all the member functions.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class student

{

private:

int admno;

char sname[20];

float eng, maths, science;

float total;

void ctotal()

{

total=eng+maths+science;

}

public:

void Takedata();

void Showdata();

};

void student::Takedata()

{

cout<<"\nEnter student details: ";

cout<<"\Enter name: ";

gets(sname);

cout<<"\nEnter admission no.: ";

cin>>admno;

cout<<"\nEnter English marks: ";

cin>>eng;

cout<<"\nEnter Math marks: ";

cin>>maths;

cout<<"\nEnter Science marks: ";

cin>>science;

ctotal();

}

void student::Showdata()

{

cout<<"\Name: ";

puts(sname);

cout<<"\nAdmission no.: ";

cout<<admno;

cout<<"\nEnglish marks: ";

cout<<eng;

cout<<"\nMath marks: ";

cout<<maths;

cout<<"\nScience marks: ";

cout<<science;

cout<<"\nTotal marks: ";

cout<<total;

}

void main()

{

clrscr();

student s1;

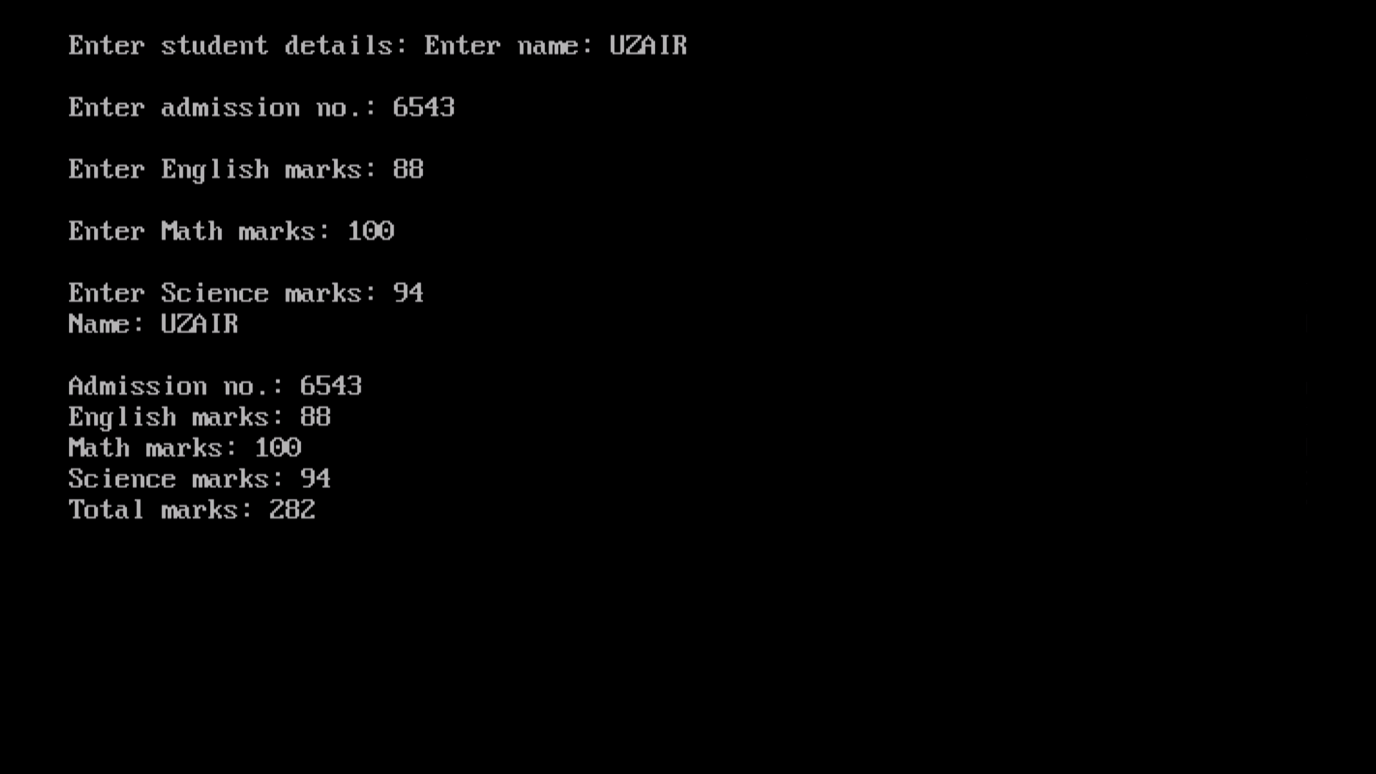
s1.Takedata();

s1.Showdata();

getch();

}

**OUTPUT:**



1. **Define a class worker with the following specification**

**Roll\_no integer, wno integer, wname 25 character, hrwrk float, wgrate float, totwage float and function calcwg() to find totwage=hrwrk\*wgrate with float return type.**

**Public member - In\_data() a function to accept values of wno, wname, hrwrk, wgrate and invoke calcwg() to calculate totpay.**

**Out\_data() a function to display all the data members on the screen you should give definitions of functions**

**Write a C++ program to create a class and invoke all the member functions.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class worker

{

int roll\_no;

int wno;

char wname[25];

float hrwrk;

float wgrate;

float totwage;

float calcwg()

{

float totpay;

totpay=hrwrk\*wgrate;

return totpay;

}

public:

void In\_data();

void Out\_data();

};

void worker::In\_data()

{

cout<<"\nEnter worker details: ";

cout<<"\Enter name: ";

gets(wname);

cout<<"\nEnter roll no.: ";

cin>>roll\_no;

cout<<"\nEnter worker no.: ";

cin>>wno;

cout<<"\nEnter work hours: ";

cin>>hrwrk;

cout<<"\nEnter wage rate: ";

cin>>wgrate;

totwage=calcwg();

}

void worker::Out\_data()

{

cout<<"\Name: ";

puts(wname);

cout<<"\nRoll no.: ";

cout<<roll\_no;

cout<<"\nWorker no.: ";

cout<<wno;

cout<<"\nWork Hours: ";

cout<<hrwrk;

cout<<"\nWage rate: ";

cout<<wgrate;

cout<<"\nTotal wage: ";

cout<<totwage;

}

void main()

{

clrscr();

worker w1;

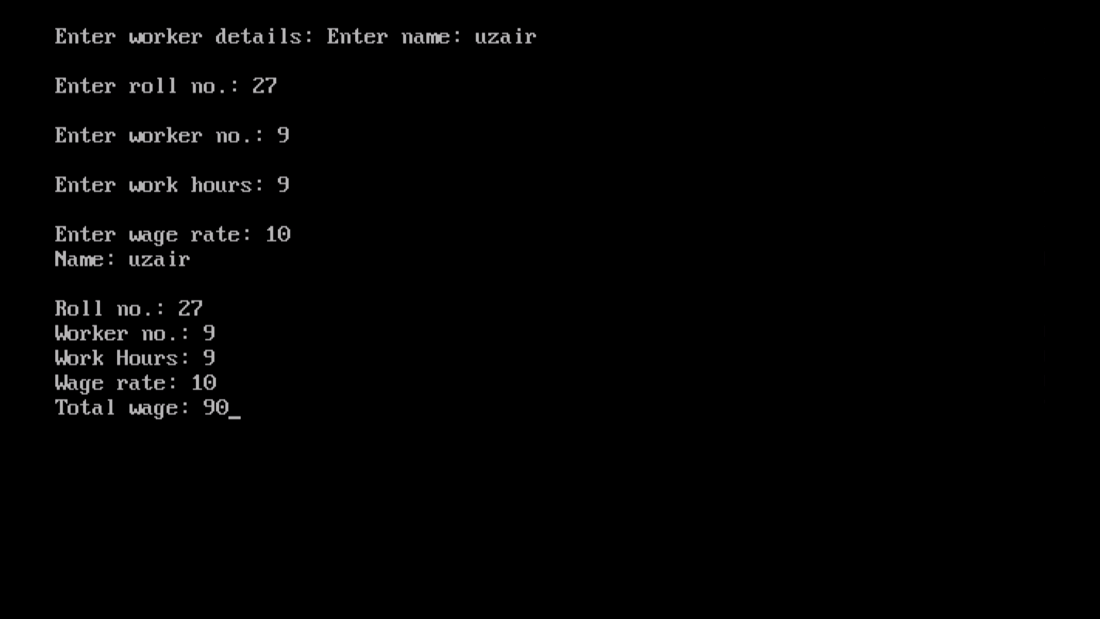
w1.In\_data();

w1.Out\_data();

getch();

}

**OUTPUT:**



1. **A class CLOCK has the following members:**

**Data member: hour of type integer, minute of type integer**

**Member functions: readtime(int h, int m); showtime() to display data member, addtime(time T1, time T2).**

**Write a program to input two different objects FT and ST, print their sum (assume 24 hr. clock time)**

**e.g. input FT=6 hrs. 35mins, ST=3hrs 45 min then output T=FT+ST=10hrs 20min**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

class CLOCK

{

int hour;

int minute;

public:

void readtime(int, int);

void showtime();

void addtime(CLOCK, CLOCK);

} z;

void CLOCK::readtime(int a, int b)

{

hour= a;

minute= b;

}

void CLOCK::showtime()

{

cout<<"\nTotal Time: ";

cout<<hour<<" hrs "<<minute<<" mins ";

}

void CLOCK::addtime(CLOCK x, CLOCK y)

{

hour=0;

minute=0;

hour=x.hour+y.hour;

minute=x.minute+y.minute;

if(minute>=60)

{

hour+=minute/60;

minute=minute%60;

}

}

void main()

{

clrscr();

CLOCK c1, c2, z;

int f, g, i, j;

cout<<"\nEnter time 1 in hours: ";

cin>>f;

cout<<"\nEnter time 1 in minutes: ";

cin>>g;

cout<<"\nEnter time 2 in hours: ";

cin>>i;

cout<<"\nEnter time 2 in minutes: ";

cin>>j;

c1.readtime(f, g);

c2.readtime(i, j);

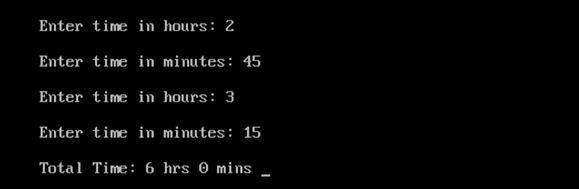
z.addtime(c1, c2);

z.showtime();

getch();

}

**OUTPUT:**



1. **A class serial has the following data members**

**Scode integer, title 20 character, duration float, noofepisodes integer**

**and members functions**

**init() to initialize duration as 30 and noofepisodes as 10**

**Newserial() function to accept values for serialcode and title**

**Otherentries() function to assign the values of all data members with the help of corresponding values passed as parameters to this function.**

**Dispdata() function to display all the data members on the screen.**

**Write a C++ program to create a class and invoke all the member functions.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class serial

{

int Scode;

char title[30];

float duration;

int noofepisodes;

public:

void init()

{

duration=30;

noofepisodes=10;

}

void Newserial();

void Otherentries(int, char[]);

void Dispdata();

};

void serial::Newserial()

{

cout<<"\nNew Serial Details: ";

cout<<"\nEnter serial code: ";

cin>>Scode;

cout<<"\nEnter serial title: ";

gets(title);

Otherentries(Scode, title);

}

void serial::Otherentries(int , char [])

{

cout<<"\nEnter duration: ";

cin>>duration;

cout<<"\nEnter no. of episodes: ";

cin>>noofepisodes;

}

void serial::Dispdata()

{

cout<<"\nSerial Details: ";

cout<<"\nSerial code: ";

cout<<Scode;

cout<<"\nSerial title: ";

puts(title);

cout<<"\nDuration: ";

cout<<duration;

cout<<"\nNo. of episodes: ";

cout<<noofepisodes;

}

void main()

{

clrscr();

serial s1;

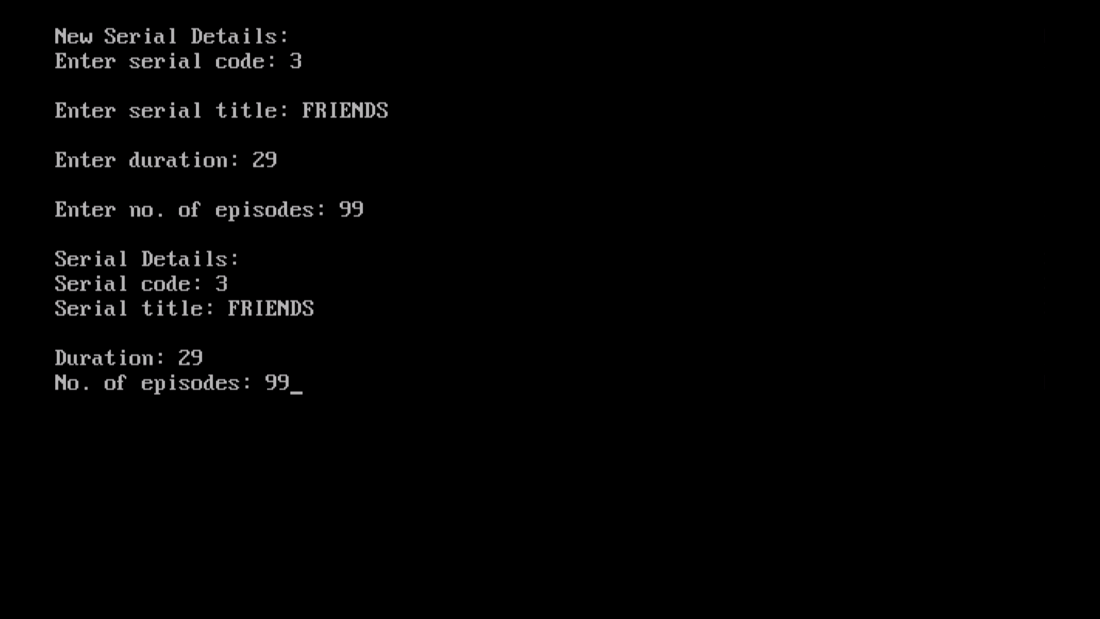
s1.Newserial();

s1.Dispdata();

getch();

}

**OUTPUT:**



1. **A class student has three data members name, roll number, marks of 5 subjects and two member functions to accept data and to assign streams on the basis of table given below. Develop a C++ program to accept the data and print the stream:**

**Average Marks Stream**

**96% or more Computer Science**

**91% - 95% Electronics**

**86% to 90% Mechanical**

**81% to 85% Electrical**

**76% to 80% Chemical**

**71% to 75% Civil**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class student

{

char name[30];

int rollno;

int marks[5];

public:

void accept();

void streamselec(float);

};

void student::accept()

{

int sum=0;

float avg=0, percent=0;

cout<<"\nEnter student details: ";

cout<<"\nEnter student name: ";

gets(name);

cout<<"\nEnter student roll no.: ";

cin>>rollno;

cout<<"\nEnter marks (out of 100): ";

for(int i=0; i<5; i++)

{

cout<<"\nMarks "<<i+1<<" ";

cin>>marks[i];

sum+=marks[i];

}

avg=sum/5;

percent=avg/5;

streamselec(percent);

}

void student::streamselec(float x)

{

cout<<"\nEligible Stream: ";

if(x>=96)

cout<<"Computer Science";

else if(x>=91 && x<=95)

cout<<"Electronics";

else if(x>=86 && x<=90)

cout<<"Mechanical";

else if(x>=81 && x<=85)

cout<<"Electrical";

else if(x>=76 && x<=85)

cout<<"Chemical";

else if(x>=71 && x<=75)

cout<<"Civil";

else

cout<<"\nNot Eligible";

}

void main()

{

clrscr();

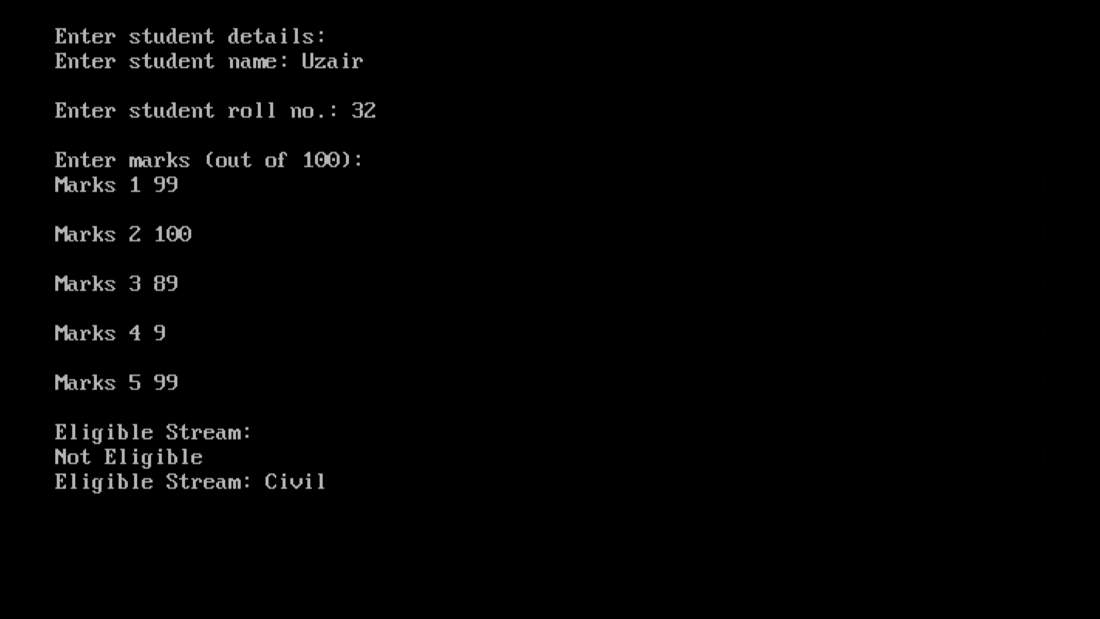
student s1;

s1.accept();

getch();

}

**OUTPUT:**



**CONSTRUCTORS AND DESTRUCTORS**

1. **Write C++ program to generate Fibonacci series using a class fib, which have the following data members and member functions:**

**Data members: first, second (i.e first 2 terms of the series) and n(no. of terms)**

**Constructor to initialize first, second and n**

**Parameterised Constructor to take the value of first, second and n from the user.**

**Function gen\_fib() to generate the series.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

class fib

{

private:

int first;

int second;

int n;

public:

fib()

{

first=0;

second=1;

n=10;

}

fib(int a, int b, int c)

{

first=a;

second=b;

n=c;

}

void gen\_fib();

};

void fib::gen\_fib()

{

int i, k;

cout<<"\nFibonacci Sequence: ";

cout<<first<<" "<<second;

for(k=1; k<n-1; k++)

{

i=first+second;

first=second;

second=i;

cout<<" "<<i;

}

}

void main()

{

clrscr();

fib f2, f1;

int x, y, z;

cout<<"\Enter first value: ";

cin>>x;

cout<<"\nEnter second value: ";

cin>>y;

cout<<"\nEnter limit: ";

cin>>z;

fib f2(x, y, z);

f2.gen\_fib();

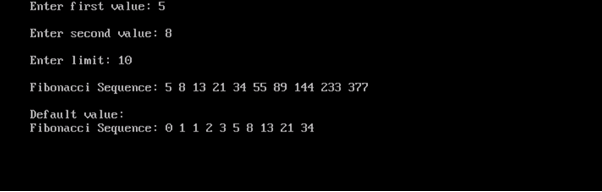
cout<<"\n"<<"\nDefault value: ";

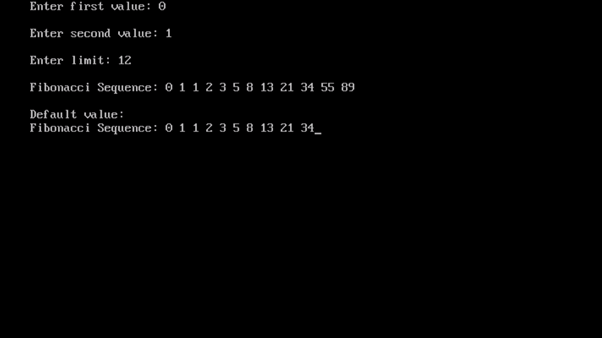
f1.gen\_fib();

getch();

}

**OUTPUT:**





1. **Write a C++ program to evaluate ab, by creating a class power where a and b are integer variables**

**Constructor to initialize a and b**

**Parameterized constructor to accept the values**

**Function disp() to display the result values**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<math.h>

class power

{

int a;

int b;

public:

power()

{

a=5;

b=2;

}

power(int x, int y)

{

a=x;

b=y;

}

void disp();

};

void power::disp()

{

int powe;

powe=pow(a,b);

cout<<"\nThe value is: ";

cout<<powe;

}

void main()

{

clrscr();

power p2, p1;

int i, j;

cout<<"\nEnter base no.: ";

cin>>i;

cout<<"\nEnter power number: ";

cin>>j;

power p1(i,j);

p1.disp();

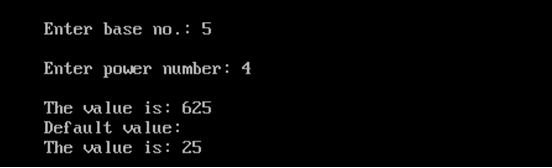
cout<<"\nDefault value: ";

p2.disp();

getch();

}

**OUTPUT:**



1. **Write a C++ program to find the factorial of a number using a constructor to initialize the value and a destructor (generating the message “You have done it.”).**

**PROGRAM:**

cout<<"\nYou have done it";

}

void fcts();

};

void fact::fcts()

{

int i;

for(i=n; i>0; i--)

{

facts=facts\*i;

}

cout<<"\nThe factorial of "<<n<<" is ";

cout<<facts;

}

void main()

{

clrscr();

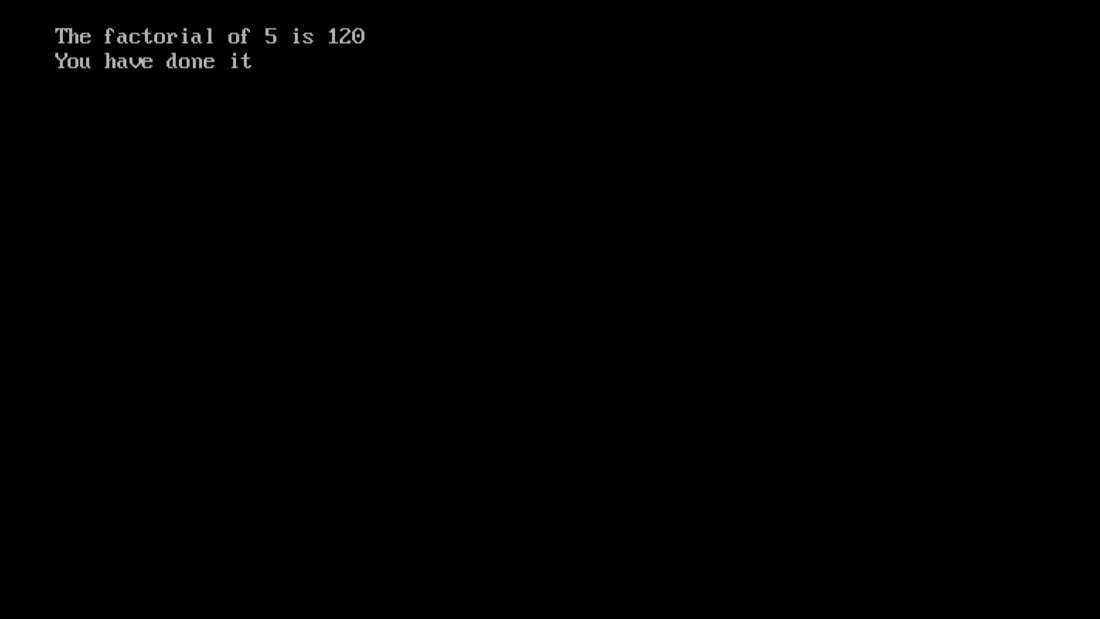
fact f1;

f1.fcts();

getch();

}

**OUTPUT:**



1. **Develop a program with the given fields and function:**

**Display a class play in C++ with the following**

* **Playcode integer**
* **Playtitle 25 character**
* **Duration float**
* **Noofscenes integer**

**Public member function of class play**

* **A constructor function to initialise duration as 45 and Noofscenes as 5.**
* **Newplay() function to accept values for Playcode and Playtitle.**
* **Moreinfo() function to assign the values of duration and Noofscenes with the help of corresponding values passed as parameters to this function.**
* **Showplay() function to display all the data members on the screen.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class play

{

int Playcode;

char Playtitle[25];

float Duration;

int Noofscenes;

public:

play()

{

Duration=45;

Noofscenes=5;

}

void Newplay();

void Moreinfo(float, int);

void Showinfo();

};

void play::Newplay()

{

cout<<"\nEnter play code: ";

cin>>Playcode;

cout<<"\nEnter play title: ";

gets(Playtitle);

}

void play::Moreinfo(float x,int y)

{

Duration=x;

Noofscenes=y;

}

void play::Showinfo()

{

cout<<"\n\t\t\tPlay Details ";

cout<<"\nPlay Title: ";

puts(Playtitle);

cout<<"\nPlay Code: ";

cout<<Playcode;

cout<<"\nDuration: ";

cout<<Duration;

cout<<"\nNo. of scenes: ";

cout<<Noofscenes;

}

void main()

{

clrscr();

play p1;

float dur;

int no;

p1.Newplay();

cout<<"\nEnter duration: ";

cin>>dur;

cout<<"\nEnter no. of scenes: ";

cin>>no;

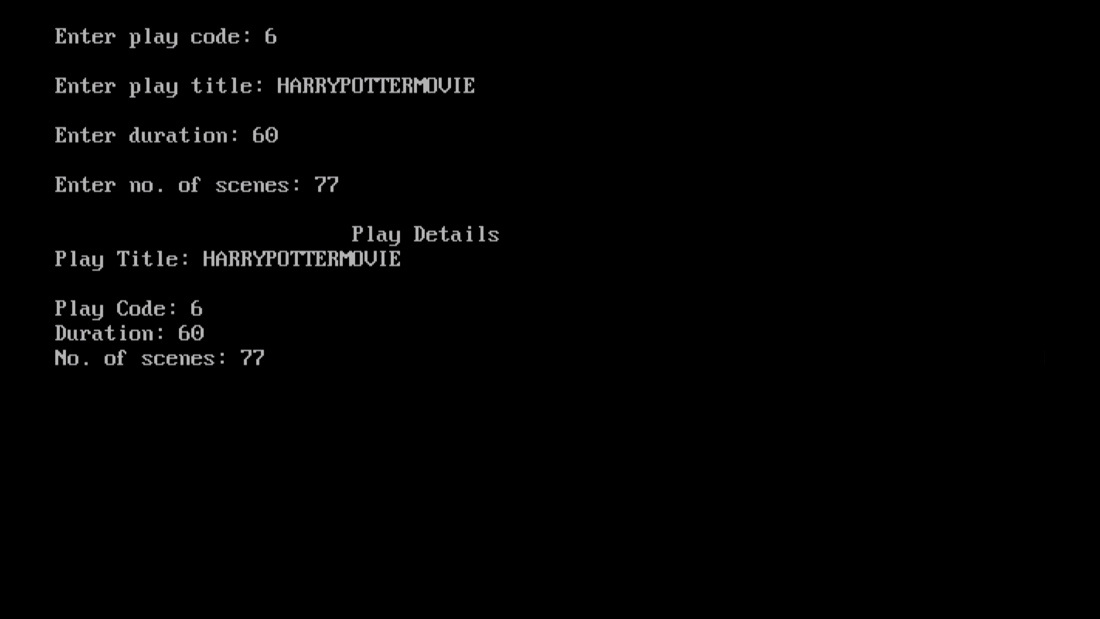
p1.Moreinfo(dur, no);

p1.Showinfo();

getch();

}

**OUTPUT:**



1. **Develop a program with the given fields and function :**

**Create a class box whose constructor function passes three values, each of which represents the length of one side of a box. From the box class compute the volume of the box and store the result in a double variable. Include a member function called vol() that displays the volume of each box object.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

class box

{

int l;

int b;

int h;

double volm;

public:

box(int x, int y, int z)

{

l=x;

b=y;

h=z;

}

void vol()

{

volm=l\*b\*h;

cout<<"\nThe volume of box is: ";

cout<<volm;

}

};

void main()

{

clrscr();

int i;

int l1, b2, h1;

cout<<"\nEnter length: ";

cin>>l1;

cout<<"\nEnter breadth: ";

cin>>b2;

cout<<"\nEnter height: ";

cin>>h1;

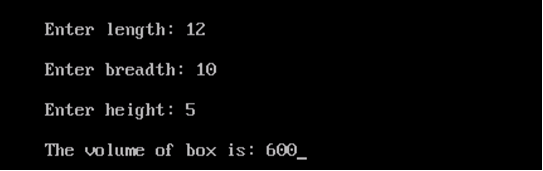
box b1(l1, b2, h1);

b1.vol();

getch();

}

**OUTPUT:**



**INHERITANCE**

1. **A publisher company markets both books and CDs. Create a class publication that stores the name (string) and price (float) of books and CDs. From this class derive two classes book which adds a page counts (type int), and CD which adds bytes (type int). Each of these classes should have a function getdata() to get data from the user and a function putdata() to display its data. Write a main() function to test the classes book and CD by creating instances of them, asking the user to input their data using the function getdata(), and then displaying the data with the function putdata().**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class publication

{

public:

char name[30];

float price;

};

class book: private publication

{

int counts;

public:

void getdata();

void putdata();

};

class CD: private publication

{

int bytes;

public:

void getdata();

void putdata();

};

void book::getdata()

{

cout<<"\nEnter name of the book: ";

gets(name);

cout<<"\nEnter price of the book: ";

cin>>price;

cout<<"\nEnter no. of pages of the book: ";

cin>>counts;

cout<<endl;

}

void book::putdata()

{

cout<<"\nname of the book: ";

puts(name);

cout<<"\nprice of the book: ";

cout<<price;

cout<<"\nBook pages: ";

cout<<counts<<endl;

}

void CD::getdata()

{

cout<<"\nEnter name of the CD: ";

gets(name);

cout<<"\nEnter price of the CD: ";

cin>>price;

cout<<"\nEnter bytes of the CD: ";

cin>>bytes;

cout<<endl;

}

void CD::putdata()

{

cout<<"\nName of the CD: ";

puts(name);

cout<<"\nPrice of CD: ";

cout<<price;

cout<<"\nBytes of the CD: ";

cout<<bytes;

}

void main()

{

clrscr();

book b1;

CD c1;

b1.getdata();

b1.putdata();

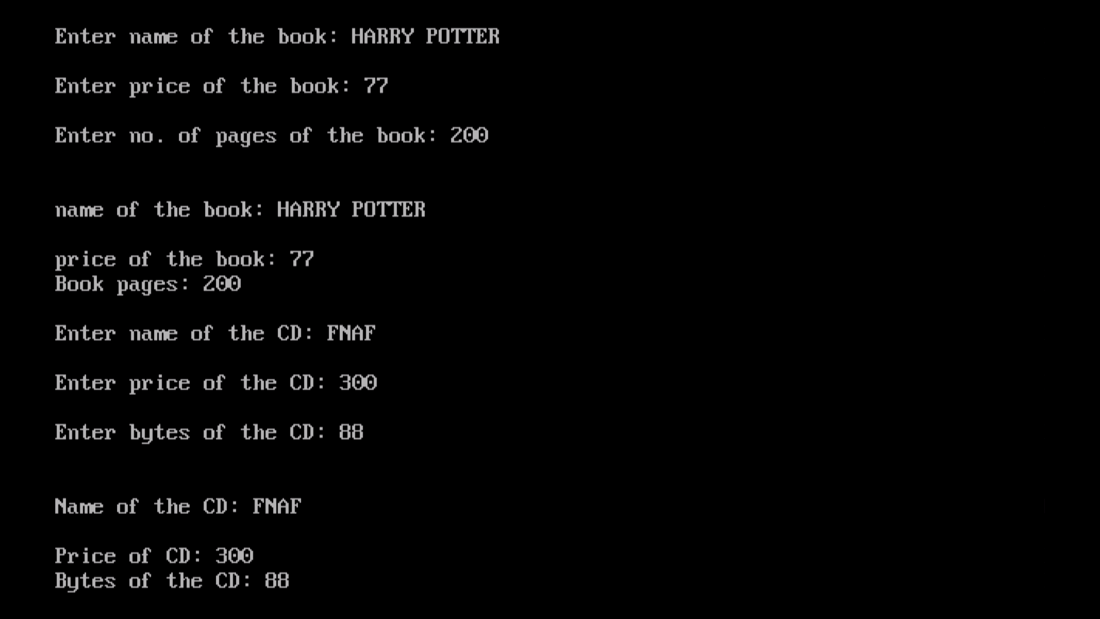
c1.getdata();

c1.putdata();

getch();

}

**OUTPUT:**



1. **Create a base class building that stores the number of floors a building has, the number of rooms and its total square footage. Create a derived class called house that inherits building and also stores the number of bedrooms and number of bathrooms. Next, create a derived class called office that inherits building and also stores the number of fire extinguishers and the number of telephones.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

class building

{

public:

int floors;

int rooms;

float sqft;

};

class house: private building

{

int bedrooms;

int bathrooms;

public:

void putdata();

void getdata();

};

class office: private building

{

int fireexting;

int tele;

public:

void putdata();

void getdata();

};

void house::getdata()

{

cout<<"\t\t\t\tHOUSE DETAILS";

cout<<"\nEnter no. of floors: ";

cin>>floors;

cout<<"\nEnter no. of rooms: ";

cin>>rooms;

cout<<"\nEnter no. of bedrooms: ";

cin>>bedrooms;

cout<<"\nEnter no. of bathrooms: ";

cin>>bathrooms;

cout<<"\nEnter total size of house: ";

cin>>sqft;

}

void house::putdata()

{

cout<<"\nNo. of floors: ";

cout<<floors;

cout<<"\nNo. of rooms: ";

cout<<rooms;

cout<<"\nNo. of bedrooms: ";

cout<<bedrooms;

cout<<"\nNo. of bathrooms: ";

cout<<bathrooms;

cout<<"\nTotal size of house: ";

cout<<sqft;

}

void office::getdata()

{

cout<<"\n\t\t\t\tOFFICE DETAILS";

cout<<"\nEnter no. of floors: ";

cin>>floors;

cout<<"\nEnter no. of rooms: ";

cin>>rooms;

cout<<"\nEnter no. of fire extinguishers ";

cin>>fireexting;

cout<<"\nEnter no. of telephones: ";

cin>>tele;

cout<<"\nEnter total size of office: ";

cin>>sqft;

}

void office::putdata()

{

cout<<"\nNo. of floors: ";

cout<<floors;

cout<<"\nNo. of rooms: ";

cout<<rooms;

cout<<"\nNo. of fire extinguishers: ";

cout<<fireexting;

cout<<"\nNo. of telephones: ";

cout<<tele;

cout<<"\nTotal size of office: ";

cout<<sqft;

}

void main()

{

clrscr();

house h1;

office o1;

h1.getdata();

h1.putdata();

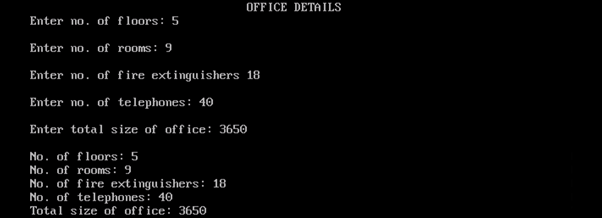
o1.getdata();

o1.putdata();

getch();

}

**OUTPUT:**



1. **Write a program that reads the data of a student and computes its grade using single inheritance.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class student

{

public:

char name[30];

int rollno;

};

class marks:private student

{

char grade;

int marks;

public:

void getdata();

void gradecalc();

void putdata();

};

void marks::getdata()

{

cout<<"\n\t\t\t\tSTUDENT DETAILS";

cout<<"\nEnter student name: ";

gets(name);

cout<<"\nEnter student roll no.: ";

cin>>rollno;

cout<<"\nEnter marks: ";

cin>>marks;

}

void marks::gradecalc()

{

if(marks>=91)

grade='A';

else if (marks<=90 && marks>=81)

grade='B';

else if(marks<=80 && marks>=71)

grade='C';

else if(marks<=70 && marks>=61)

grade='D';

else if(marks<=60 && marks>=51)

grade='E';

else

grade='F';

}

void marks::putdata()

{

cout<<"\nStudent name: ";

puts(name);

cout<<"\nStudent roll no.: ";

cout<<rollno;

cout<<"\nStudent marks (out of 100): ";

cout<<marks;

cout<<"\nStudent grade: ";

cout<<grade;

}

void main()

{

clrscr();

marks m1;

m1.getdata();

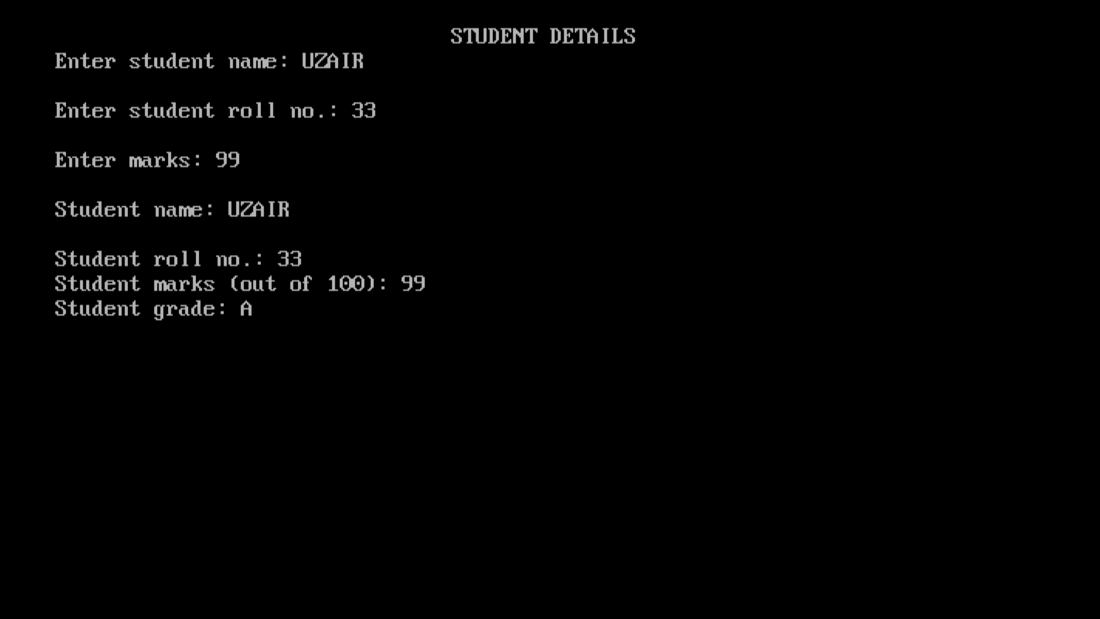
m1.gradecalc();

m1.putdata();

getch();

}

**OUTPUT:**



1. **A college maintains a list of its students graduating every year. At the end of the year, the college produces a report that lists the following:**

**Number of working Graduates:**

**Number of non-working graduates:**

**Name:**

**Age:**

**Subject:**

**Average Marks:**

**X % of the graduates this year are non-working and n% are first divisioners.**

**Write a C++ program for it that uses the following inheritance path:**

**Person  Student  Graduate**

**(name, age) (rollno, avg marks) (student, employed)**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class Person

{

public:

char name[30];

int age;

};

class Student:public Person

{

public:

char subject[30];

int rollno;

int marks[3];

int sum;

float avg;

};

class Graduate:public Student

{

int noofstud;

int noofemp;

public:

void getdata();

void putdata();

};

void Graduate::getdata()

{

sum=0;

cout<<"\nSTUDENT DETAILS";

cout<<"\nEnter student name: ";

gets(name);

cout<<"\nEnter student age: ";

cin>>age;

cout<<"\nEnter subject: ";

gets(subject);

for(int i=0; i<3; i++)

{

cout<<"\nEnter subject marks: ";

cin>>marks[i];

sum+=marks[i];

}

avg=0;

avg=sum/3;

cout<<"\nEnter no. of working graduates: ";

cin>>noofemp;

cout<<"\nEnter no. of non-working graduates: ";

cin>>noofstud;

}

void Graduate::putdata()

{

cout<<"\nNo.of working graduates: ";

cout<<noofemp;

cout<<"\nNo. of non-working graduates: ";

cout<<noofstud;

cout<<"\nName of student: ";

puts(name);

cout<<"\nAge of student: ";

cout<<age;

cout<<"\nSubject of student: ";

puts(subject);

cout<<"\Average marks of student: ";

cout<<avg;

}

void main()

{

clrscr();

Graduate g1;

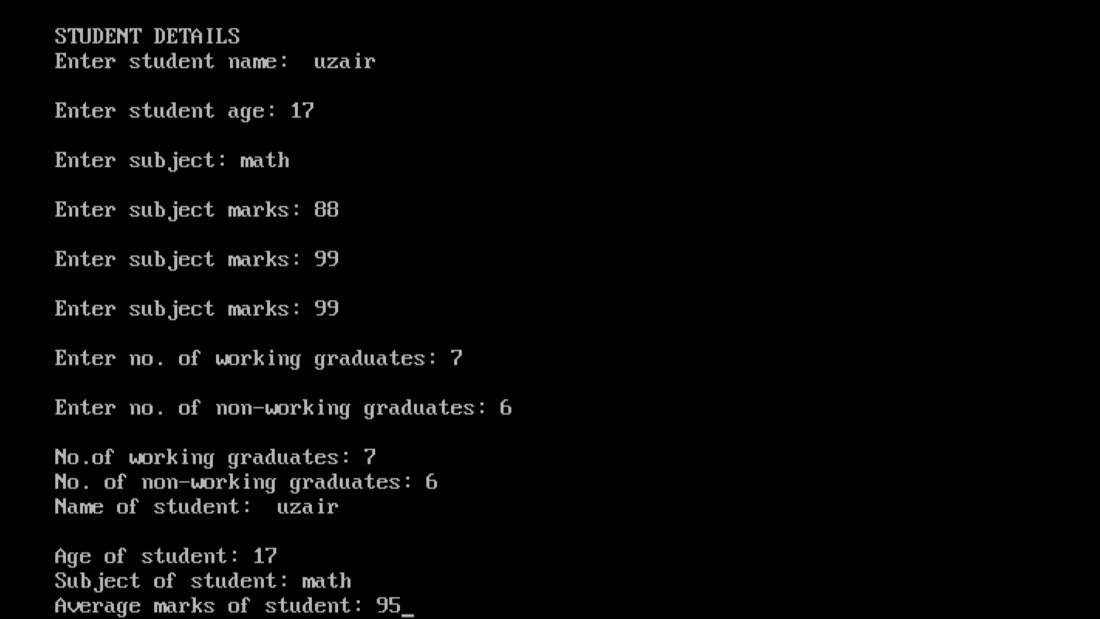
g1.getdata();

g1.putdata();

getch();

}

**OUTPUT:**



1. **Write a C++ program to read and display information about employee and managers. Employee is a class that contains employee number, name, address and department. Manager class contains all information of the employee class and a list of employees working under a manager.**

**PROGRAM:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class Employee

{

public:

int empno[10];

char ename[30];

char eaddress[30];

char edept[30];

}e[3];

class Manager:private Employee

{

int mngemp;

int flag;

public:

void getdata();

void putdata();

};

void Manager::getdata()

{

cout<<"\n\t\t\t\tEMPLOYEE DETAILS";

for(int i=0; i<3; i++)

{

cout<<"\nEnter employee name: ";

gets(e[i].ename);

cout<<"\nEnter employee no.: ";

cin>>empno[i];

cout<<"\nEnter employee address: ";

gets(e[i].eaddress);

cout<<"\nEnter employee department: ";

gets(e[i].edept);

}

}

void Manager::putdata()

{

cout<<"\nEmployee working under manager: ";

for(int i=0; i<3; i++)

{

cout<<"\nEmployee name: ";

puts(e[i].ename);

cout<<"\nEmployee no.: ";

cout<<empno[i];

cout<<"\nEmployee address: ";

puts(e[i].eaddress);

cout<<"\nEmployee department: ";

puts(e[i].edept);

}

}

void main()

{

clrscr();

Manager m1;

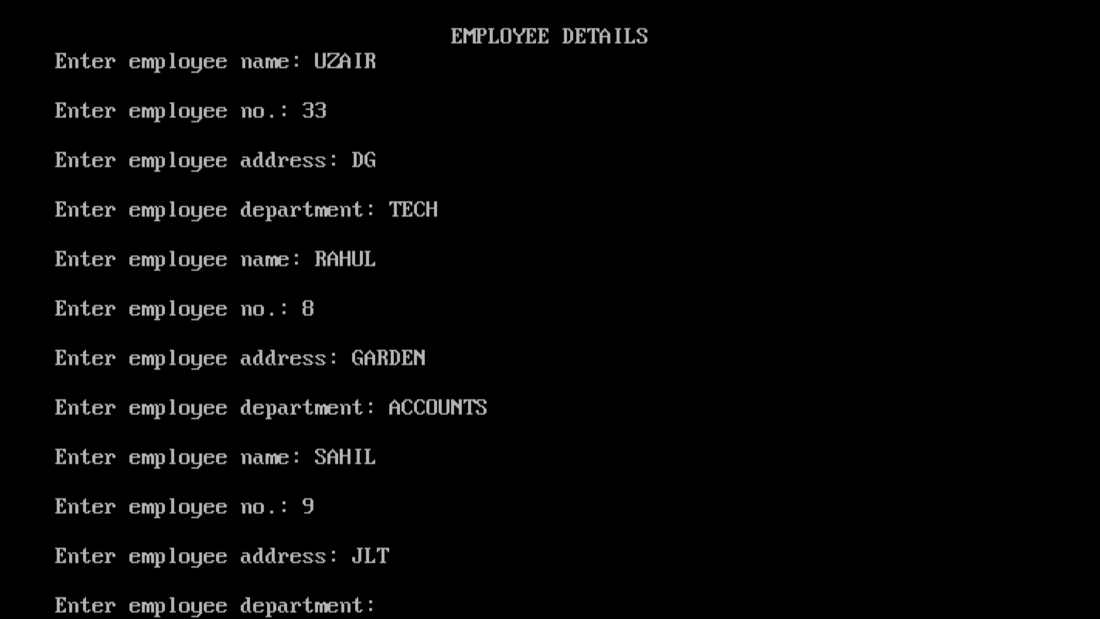
m1.getdata();

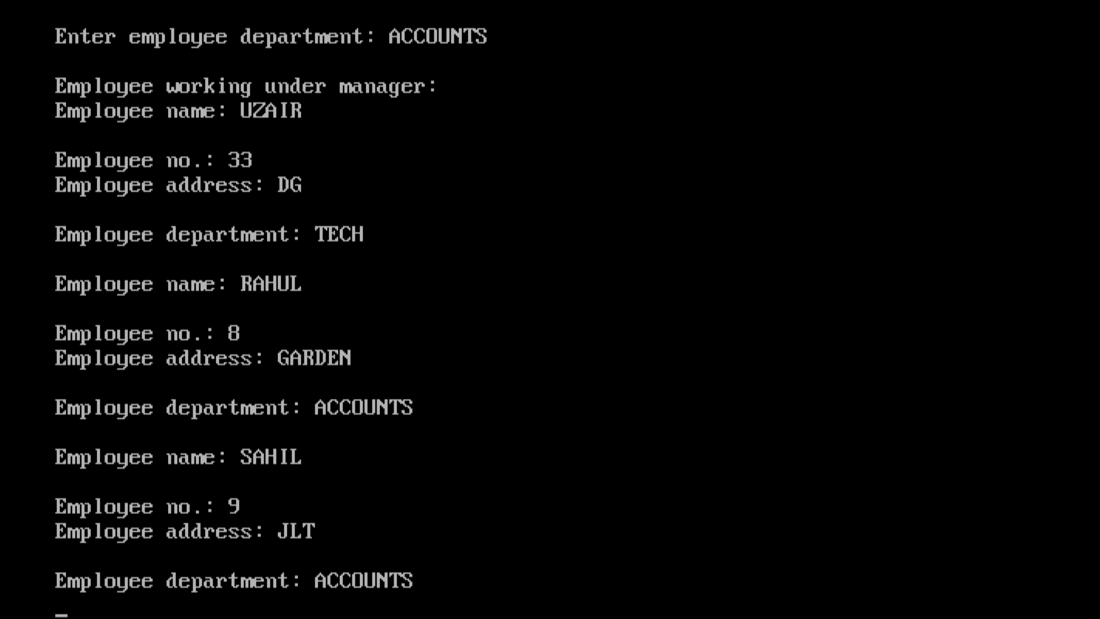
m1.putdata();

getch();

}

**OUTPUT:**





**FILE HANDLING**

1. **Write a program in C++ to count the number of uppercase alphabets and number of vowels in a text file “abc.txt”.**

**PROGRAM:**

#include<fstream.h>

#include<conio.h>

#include<ctype.h>

void main()

{

clrscr();

ifstream fin;

fin.open("abc.txt", ios::in);

char ch;

int ctr1=0, ctr2=0;

while(!fin.eof())

{

fin.get(ch);

if(isupper(ch))

ctr1++;

if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

ctr2++;

}

cout<<"\No. of uppercase characters: ";

cout<<ctr1;

cout<<"\nNo. of vowels: ";

cout<<ctr2;

fin.close();

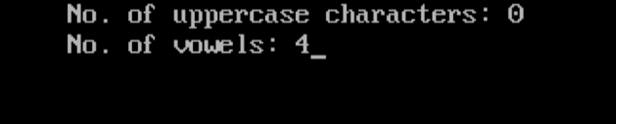
getch();

}

**TEXT:**

uzair rocks

**OUTPUT:**



1. **Write a C++ program to read and write structure emp(eno, ename, edesig, esal) using read() and write() function in a binary file.**

**PROGRAM:**

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

struct emp

{

int empno;

char ename[30];

char edesig[30];

long int sal;

} e;

void main()

{

clrscr();

int i;

ifstream fin;

ofstream fout;

cout<<"\nEMPLOYEE DETAILS";

cout<<"\nEnter employee name: ";

gets(e.ename);

cout<<"\nEnter employee code: ";

cin>>e.empno;

cout<<"\nEnter employee designation: ";

gets(e.edesig);

cout<<"\nEnter employee salary: ";

cin>>e.sal;

fout.open("emp.dat", ios::out|ios::binary);

fout.write((char\*)&e, sizeof(e));

fout.close();

fin.open("emp.dat", ios::in|ios::binary);

if(!fin)

{

cout<<"\nFile not found";

}

fin.read((char\*)&e, sizeof(e));

cout<<"\nEMPLOYEE DETAILS";

cout<<"\nEmployee name: ";

puts(e.ename);

cout<<"\nEmployee code: ";

cout<<e.empno;

cout<<"\nEmployee designation: ";

puts(e.edesig);

cout<<"\nEmployee salary: ";

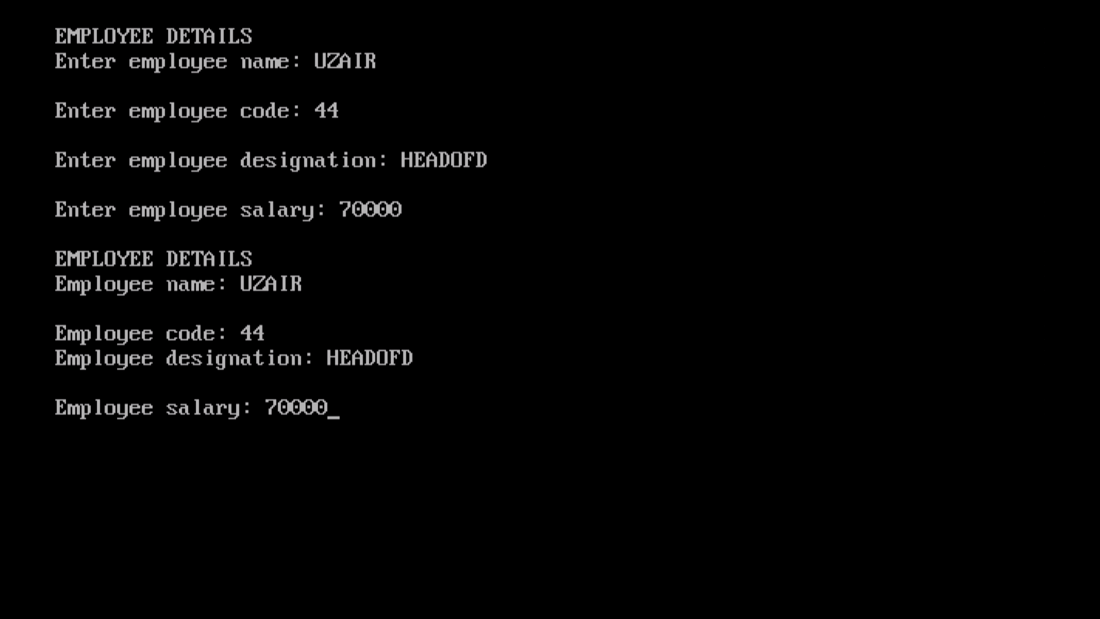
cout<<e.sal;

fin.close();

getch();

}

**OUTPUT:**



1. **Write a program to delete the record from file having records maintained through classes.**

**PROGRAM:**

#include<fstream.h>

#include<stdio.h>

#include<conio.h>

class student

{

char name[30];

int rollno;

float marks;

public:

void getdata()

{

cout<<"\nEnter name: ";

gets(name);

cout<<"\nEnter roll no.: ";

cin>>rollno;

cout<<"\nEnter marks (out of 100): ";

cin>>marks;

}

void putdata()

{

cout<<"\nName: ";

puts(name);

cout<<"\nRoll no.: ";

cout<<rollno;

cout<<"\nMarks (out of 100): ";

cout<<marks;

}

int getrno()

{

return rollno;

}

};

void main()

{

clrscr();

student s;

student stu;

int i, rno;

char entry='y';

fstream fin("stud.dat", ios::in|ios::binary|ios::out);

ofstream fout("temp.dat", ios::app|ios::binary);

while(entry=='y'||entry=='Y')

{

cout<<"\nStudent Details: ";

s.getdata();

fin.write((char\*)&s, sizeof(s));

cout<<"\nDo you want to enter more records? y/n";

cin>>entry;

}

fin.close();

fin.open("stud.dat", ios::in|ios::binary);

cout<<"\nEnter roll no. whose record is to be deleted: ";

cin>>rno;

char found='f';

char confirm='n';

while(fin.read((char\*)&s, sizeof(s)))

{

if(s.getrno()==rno)

{

s.putdata();

found='t';

cout<<"\nDo you want to delete this record?";

cin>>confirm;

if(confirm=='n')

fout.write((char\*)&s, sizeof(s));

}

else

fout.write((char\*)&s, sizeof(s));

}

if(found=='f')

cout<<"\nRecord not found";

fin.close();

fout.close();

remove("stud.dat");

rename("temp.dat", "stud.dat");

fin.open("stud.dat", ios::in);

cout<<"\nNew Record: ";

while(fin.read((char\*)&stu, sizeof(stu)))

{

stu.putdata();

}

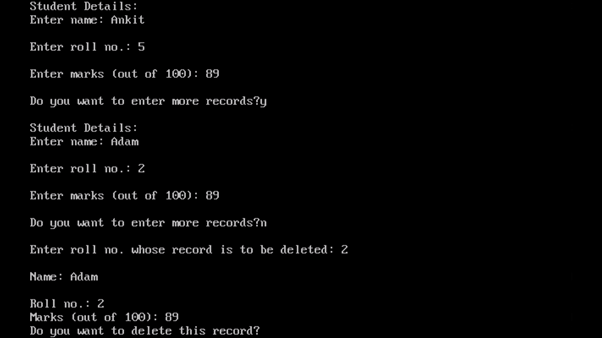
fin.close();

getch();

}

**OUTPUT:**





1. **Write a program to search a record based on rollno in a file that has records maintained through class (rollno, name, marks, average and grade) and member function to assign grade on the basis of table given below:**

**Average Marks Grade**

**90% or more A1**

**89% - 80% A2**

**70% to 70% B1**

**69% to 60% B2**

**59% to 50% C1**

**59% to 40% C2**

**Below 40% FAIL**

**PROGRAM:**

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

class student

{

int rollno;

char name[30];

int marks[3];

float avg;

char grade[3];

public:

void getdata();

void gradecalc();

void putdata();

int getrno()

{

return rollno;

}

};

void student::getdata()

{

float sum=0;

cout<<"\nEnter student details: ";

cout<<"\nEnter name: ";

gets(name);

cout<<"\nEnter roll no.: ";

cin>>rollno;

cout<<"\Enter student's marks; ";

for(int i=0; i<3; i++)

{

cout<<"\nMarks "<<i+1<<" ";

cin>>marks[i];

sum+=marks[i];

}

avg=sum/3;

}

void student::gradecalc()

{

if(avg>=90)

strcpy(grade, "A1");

else if(avg>=80 && avg<=89)

strcpy(grade, "A2");

else if(avg>=70 && avg<=79)

strcpy(grade, "B1");

else if(avg>=60 && avg<=69)

strcpy(grade, "B2");

else if(avg>=50 && avg<=59)

strcpy(grade, "C1");

else if(avg>=40 && avg<=49)

strcpy(grade, "C2");

else

strcpy(grade, "FAIL");

}

void student::putdata()

{

cout<<"\nStudent details: ";

cout<<"\nName: ";

puts(name);

cout<<"\nRoll no.: ";

cout<<rollno;

cout<<"\nStudent's average: ";

cout<<avg;

cout<<"\nGrade: ";

cout<<grade;

}

void main()

{

clrscr();

student s;

student stu;

int i, rno;

char entry='y';

char found='f';

fstream fin("stud.dat", ios::in|ios::binary|ios::out);

while(entry=='y')

{

cout<<"\nStudent Details: ";

s.getdata();

fin.write((char\*)&s, sizeof(s));

cout<<"\nDo you want to enter more records?";

cin>>entry;

}

fin.close();

cout<<"\nEnter roll no. of record to be searched: ";

cin>>rno;

fin.open("stud.dat", ios::in|ios::out);

while(fin.read((char\*)&s, sizeof(s)))

{

if(s.getrno()==rno)

{

s.gradecalc();

s.putdata();

found='y';

break;

}

}

if(found=='f')

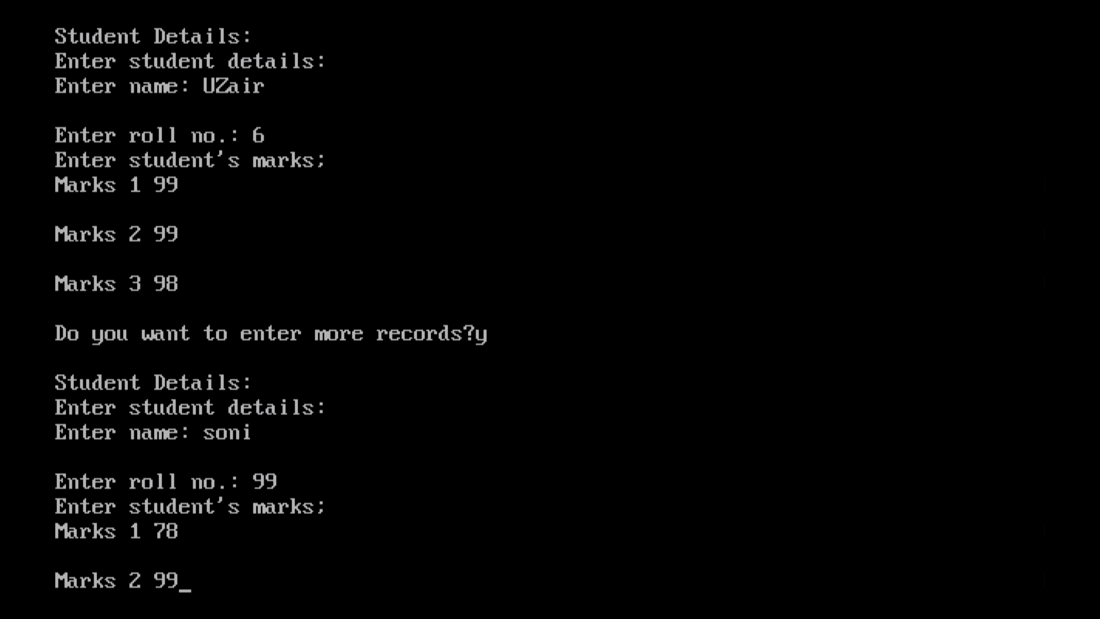
cout<<"\nRecord not found.";

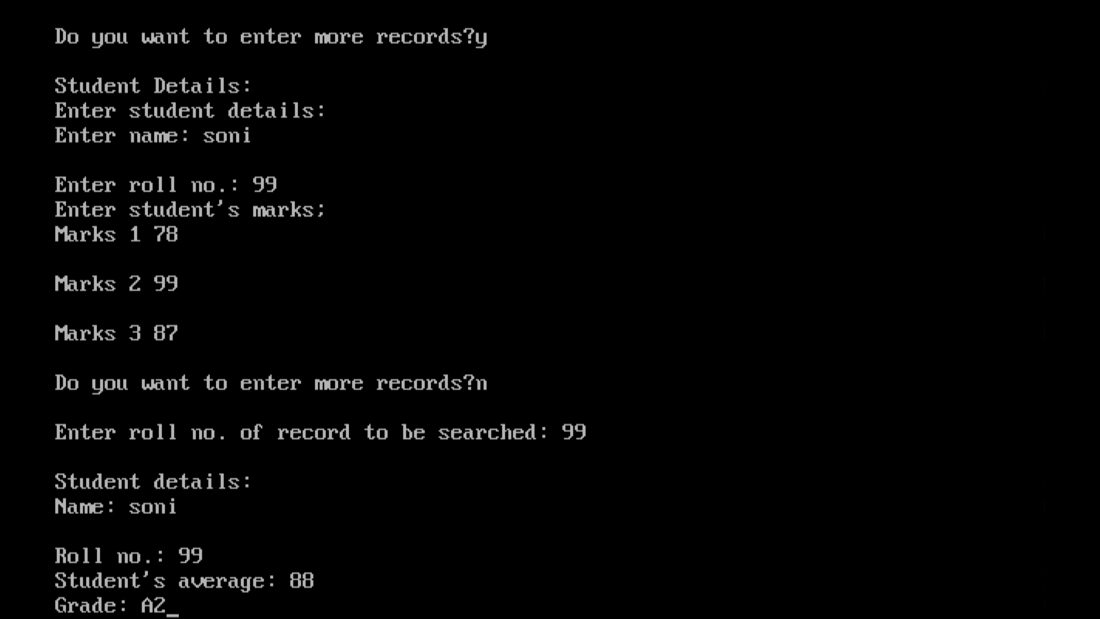
fin.close();

getch();

}

**OUTPUT:**





1. **Write a program to append data in a file having records maintained through classes(rollno, name, marks, average and grade) and member function to assign grade on the basis of table given below:**

**Average Marks Grade**

**90% or more A1**

**89% - 80% A2**

**70% to 70% B1**

**69% to 60% B2**

**59% to 50% C1**

**59% to 40% C2**

**Below 40%**

**FAIL**

**PROGRAM:**

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

class student

{

int rollno;

char name[30];

int marks[3];

float avg;

char grade[5];

public:

void getdata();

void gradecalc();

void putdata();

int getrno()

{

return rollno;

}

};

void student::getdata()

{

int i;

float sum=0;

cout<<"\nEnter student details: ";

cout<<"\nEnter name: ";

gets(name);

cout<<"\nEnter roll no: ";

cin>>rollno;

cout<<"\Enter student's marks; ";

for(i=0; i<3; i++)

{

cout<<"\nMarks "<<i+1<<" ";

cin>>marks[i];

sum+=marks[i];

}

avg=sum/3;

}

void student::gradecalc()

{

if(avg>=90)

strcpy(grade, "A1");

else if(avg>=80 && avg<=89)

strcpy(grade, "A2");

else if(avg>=70 && avg<=79)

strcpy(grade, "B1");

else if(avg>=60 && avg<=69)

strcpy(grade, "B2");

else if(avg>=50 && avg<=59)

strcpy(grade, "C1");

else if(avg>=40 && avg<=49)

strcpy(grade, "C2");

else

strcpy(grade, "FAIL");

}

void student::putdata()

{

cout<<"\nStudent details: ";

cout<<"\nName: ";

puts(name);

cout<<"\nRoll no.: ";

cout<<rollno;

cout<<"\nStudent's average: ";

cout<<avg;

cout<<"\nGrade: ";

gradecalc();

puts(grade);

}

void main()

{

clrscr();

student s, stu;

int i, rno;

char found='y';

fstream fout("stud.dat", ios::binary|ios::out);

s.getdata();

fout.write((char\*)&s, sizeof(s));

fout.close();

fout.open("stud.dat", ios::app|ios::binary);

cout<<"\nAppending Record: ";

while(found=='y'||found=='Y')

{

stu.getdata();

fout.write((char\*)&stu, sizeof(stu));

cout<<"\nDo you want to enter more records?";

cin>>found;

}

fout.close();

fout.open("stud.dat", ios::in|ios::binary);

while(fout.read((char\*)&s, sizeof(s)))

{

s.putdata();

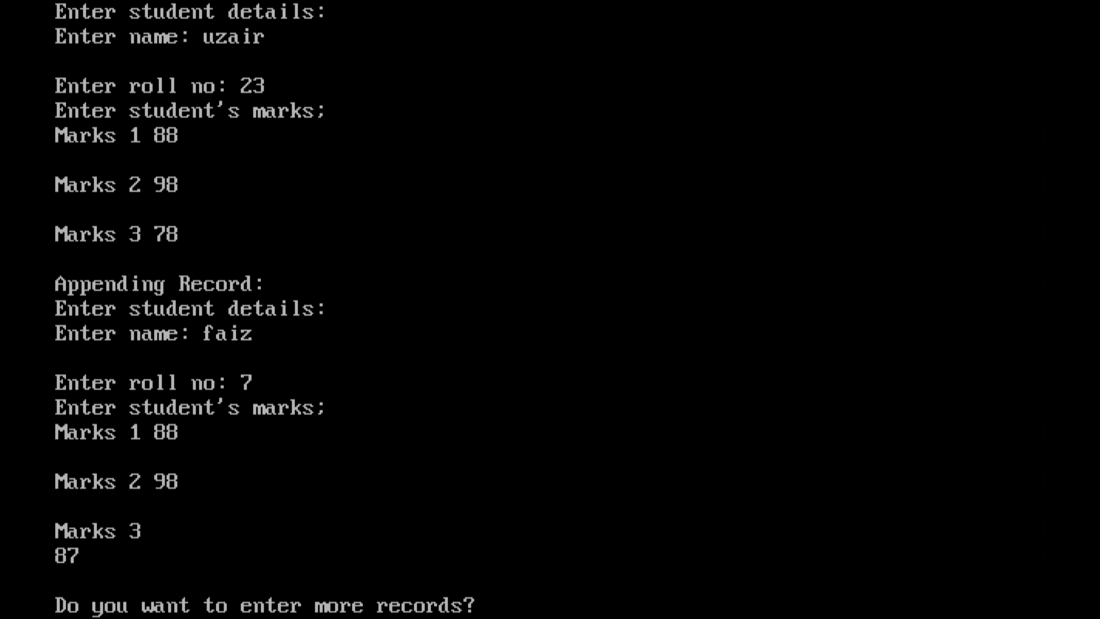
}

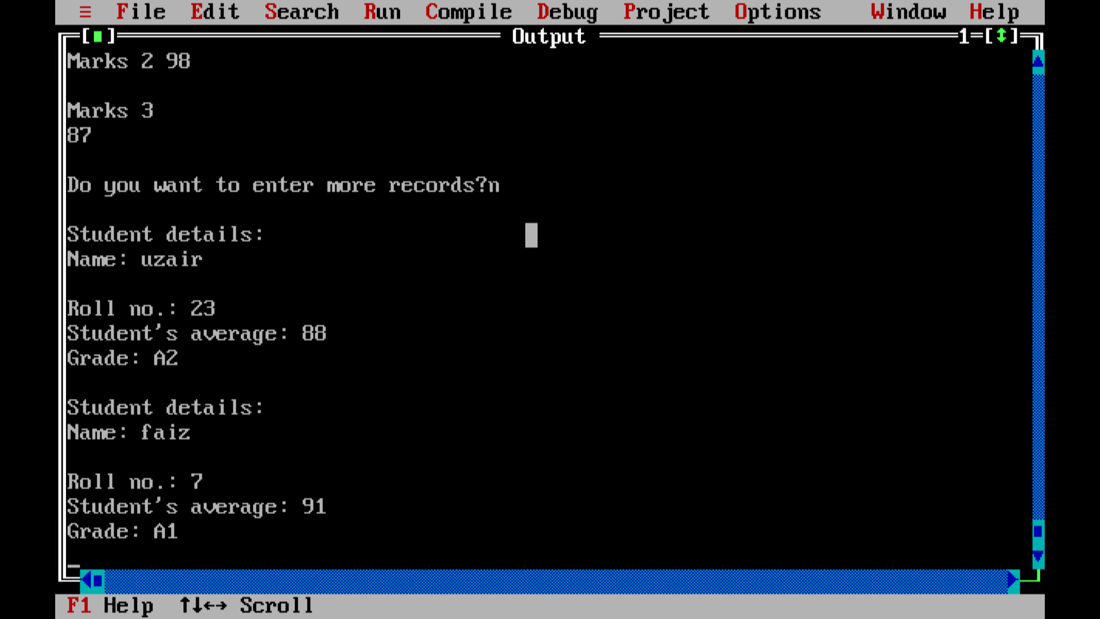
fout.close();

getch();

}

**OUTPUT:**





1. **Write a program to copy all the lines that do not begin with a capital letter to a new file “ABC.txt” from "XYZ.txt"**

**PROGRAM:**

#include<fstream.h>

#include<conio.h>

#include<ctype.h>

#include<string.h>

#include<stdio.h>

void main()

{

clrscr();

fstream fin;

fstream fout;

char ch[50];

char ch1[50];

int x;

fin.open("XYZ.TXT", ios::in);

fout.open("ABC.TXT", ios::in|ios::out);

cout<<"\nORIGINAL: ";

while(!fin.eof())

{

fin.getline(ch, 50);

puts(ch);

int i=0;

if(islower(ch[i]))

{

fout<<ch<<endl;

}

}

fin.close();

fout.seekg(0);

cout<<"\nNEW: ";

while(!fout.eof())

{

fout.getline(ch1, 50);

puts(ch1);

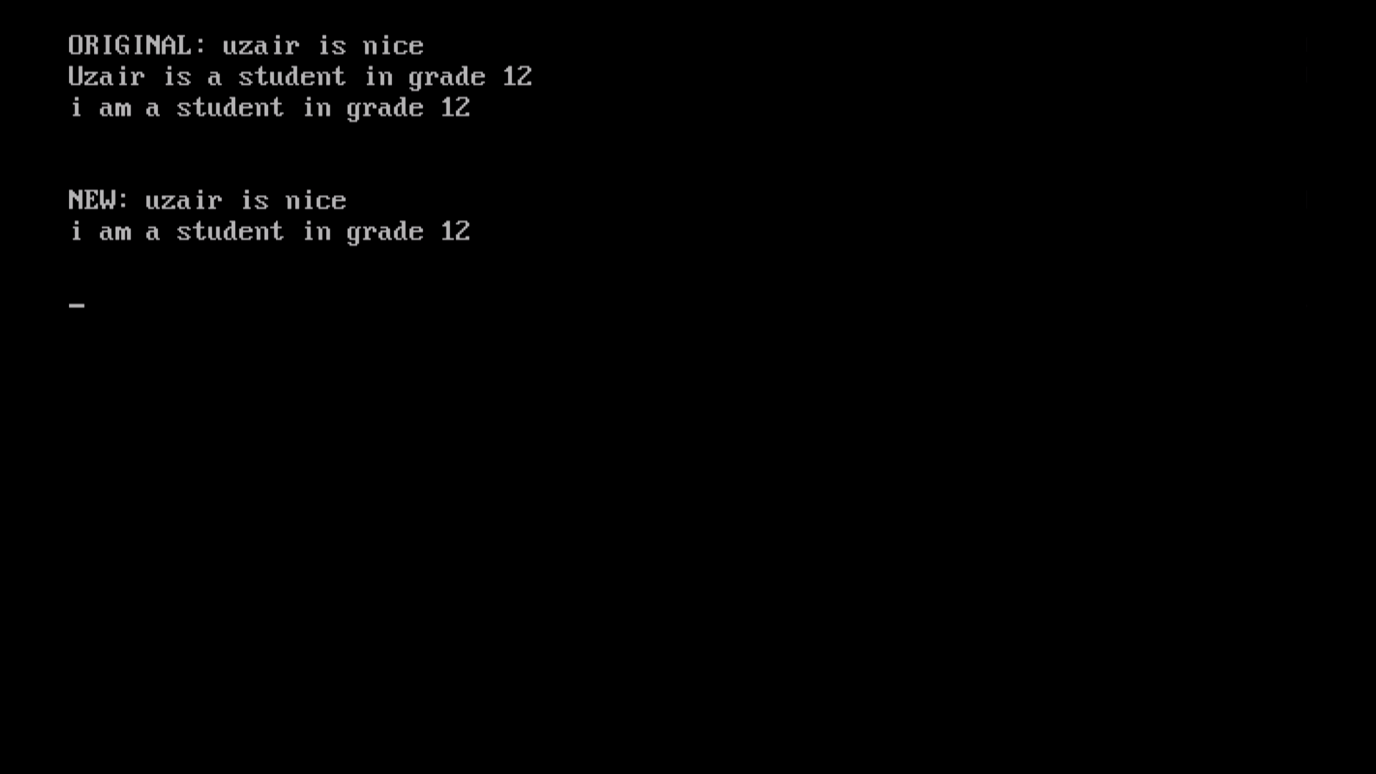
}

fout.close();

getch();

}

**OUTPUT:**



**STACKS, QUEUES AND LINKED LISTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **PROGRAMS** | **TEACHER’S**  **SIGNATURE** |
| 1. | Write a program to push and pop elements in a stack implemented as array. |  |
| 2. | Write a program to push and pop elements in a stack implemented as linked list. |  |
| 3. | Write a program to insert and delete elements in a queue implemented as array. |  |
| 4. | Write a program to insert and delete elements in a queue implemented as linked list. |  |

1. **Write a program to push and pop elements in a stack implemented as array.**

**Source code:**

#include<iostream.h>

#include<conio.h>

int stack[4], n=4, top=-1;

void push(int num)

{

if(top>=n-1)

cout<<"Stack Overflow"<<endl;

else

{

top++;

stack[top]=num;

}

}

void pop()

{

if(top==-1)

cout<<"Underflow"<<endl;

else

{

cout<<"The popped element is "<< stack[top] <<endl;

top--;

}

}

void display()

{

if(top>=0)

{

cout<<"Stack elements are:";

for(int i=top; i>=0; i--)

cout<<stack[i]<<" ";

cout<<endl;

}

else

cout<<"Stack is empty";

}

void main()

{

clrscr();

int ch, num;

cout<<"1) Push in stack"<<endl;

cout<<"2) Pop from stack"<<endl;

cout<<"3) Display stack"<<endl;

cout<<"4) Exit"<<endl;

do

{

cout<<"Enter choice: "<<endl;

cin>>ch;

switch(ch)

{

case 1:

{

cout<<"Enter value to be pushed:"<<endl;

cin>>num;

push(num);

break;

}

case 2:

{

pop();

break;

}

case 3:

{

display();

break;

}

case 4:

{

cout<<"Exit"<<endl;

break;

}

default:

{

cout<<"Invalid Choice"<<endl;

}

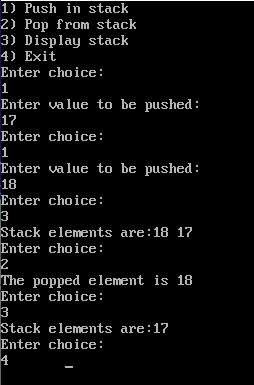
}

}while(ch!=4);

getch();

}

**Output:**



1. **Write a program to push and pop elements in a stack implemented as linked list.**

**Source code:**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct nd{

int info;

nd \*next;

}\*top,\*newptr,\*save,\*ptr;

nd \*create\_new\_nd(int);

void push(nd\*);

void pop();

void display(nd\*);

void main()

{

clrscr();

int inf;

char ch='y';

int choice;

top=NULL;

while(ch=='y')

{

cout<<"1.Push"<<endl

<<"2.Pop"<<endl

<<"3.Display"<<endl;

cin>>choice;

if(choice==1)

{

cout<<"Enter the information for nd"<<endl;

cin>>inf;

newptr=create\_new\_nd(inf);

if(newptr==NULL)

cout<<"Cannot create new nd"<<endl;

push(newptr); }

if(choice==2)

{

pop();

display(top);

}

else

{

display(top);

}

cout<<" do you want to continue (y/n) "<<endl;

cin>>ch;

}

getch();

}

nd \*create\_new\_nd(int n)

{

ptr=new nd;

ptr->info=n;

ptr->next=NULL;

return ptr;

}

void push(nd \*np)

{

if(top==NULL)

top=np;

else

{

save=top;

top=np;

np->next=save;

}

}

void display(nd \*np)

{

cout<<" stack is "<<endl;

while(np!=NULL)

{

cout<<np->info<<"<-";

np=np->next;

}

cout<<"!!!!"<<endl;

}

void pop() {

if(top==NULL){

cout<<"underflow"<<endl;}

else

{

ptr=top;

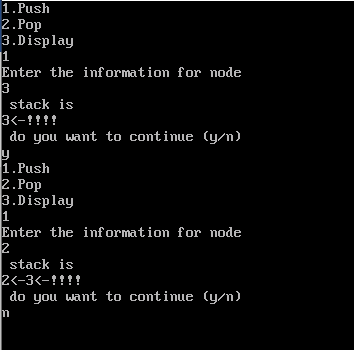
top=top->next;

delete ptr;

}

}

**Output:**



1. **Write a program to insert and delete elements in a queue implemented as array.**

**Source code:**

#include<iostream.h>

#include<conio.h>

int queue[10], n = 10, front = - 1, rear = - 1;

void Insert()

{

int val;

if (rear == n - 1)

cout<<"Queue Overflow"<<endl;

else

{

if (front == - 1)

{front = 0;}

cout<<"Insert the element in queue : "<<endl;

cin>>val;

rear++;

queue[rear] = val;

}

}

void Delete()

{

if (front == - 1)

{

cout<<"Queue Underflow ";

return ;

}

else

{

cout<<"Element deleted from queue is : "<< queue[front] <<endl;

front++;

}

}

void Display()

{

if (front == - 1)

cout<<"Queue is empty"<<endl;

else

{

cout<<"Queue elements are : ";

for (int i = front; i <= rear; i++)

cout<<queue[i]<<" ";

cout<<endl;

}

}

void main()

{

clrscr();

int ch;

cout<<"1) Insert element to queue"<<endl;

cout<<"2) Delete element from queue"<<endl;

cout<<"3) Display all the elements of queue"<<endl;

cout<<"4) Exit"<<endl;

do

{

cout<<"Enter your choice : "<<endl;

cin>>ch;

switch (ch)

{

case 1: Insert();

break;

case 2: Delete();

break;

case 3: Display();

break;

case 4: cout<<"Exit"<<endl;

break;

default: cout<<"Invalid choice"<<endl;

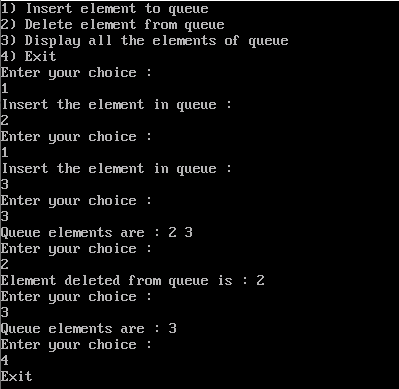
}

} while(ch!=4);

getch();

}

**Output:**



1. **Write a program to insert and delete elements in a queue implemented as linked list.**

**Source code:**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct nd

{

int info;

nd \*next;

} \*front,\*newptr,\*save,\*ptr,\*rear;

nd\* create(int a);

void insert(nd \*);

void display(nd \*);

void del();

int main()

{

clrscr();

front=rear=NULL;

int choice=0;

int inf;

char ch='y';

while(ch=='y')

{

cout <<"1.insert 2.delete 3.display "<<endl;

cin>>choice;

if(choice==1)

{

cout <<"enter information of new nd "<<endl;

cin>>inf;

newptr=create(inf);

if(newptr==NULL)

{

cout<<" aborting...error "<<endl;

exit(0);

}

insert(newptr);

}

else if(choice==2)

{

del();

display(front);

}

else if(choice==3)

{display(front);}

else

{

cout<<"incorrect choice ";

exit(0);

}

cout<<"do you want to continue (y/n) "<<endl;

cin>>ch;

}

getch();

}

nd \*create(int a)

{

ptr= new nd;

ptr->info=a;

ptr->next=NULL;

return ptr;

}

void insert(nd \*np)

{

if(front==NULL)

{

front=rear=np;

}

else

{

rear->next=np;

rear=np;

}

}

void del()

{

if(front==NULL)

{

cout<<"underflow "<<endl;

}

else

{

ptr=front;

front=front->next;

delete ptr;

}

}

void display(nd \*np)

{

cout <<"queue is "<<endl;

while(np!=NULL)

{

cout<<np->info<<" ";

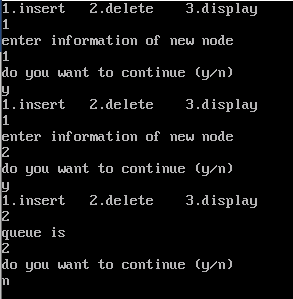
np=np->next;

}

cout<<endl;

}

**Output:**



**SQL**

**STRUCTURED QUERY LANGUAGE**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **PROGRAMS** | **TEACHER’S**  **SIGNATURE** |
| 1. | Consider the following tables Product and Client. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii) |  |
| 2. | Consider the following tables Product and Client. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii) |  |
| 3. | Consider the following tables Stock and Dealers and answer (a1) and (a2) parts of this question |  |
| 4. | Consider the following tables EMPLOYEE and SALGRADE and answer (A) and (B) parts of this question. |  |
| 5. | Consider the following tables STORES AND ITEM and answer (A) and (B) parts of this question |  |
| 6. | Consider the following tables Item and Customer. Write answer for the statement (i) to (viii). |  |
| 7. | Consider the following tables SENDER AND RECIPIENT and answer (A) and (B) parts of this question |  |

**QUESTION 1**

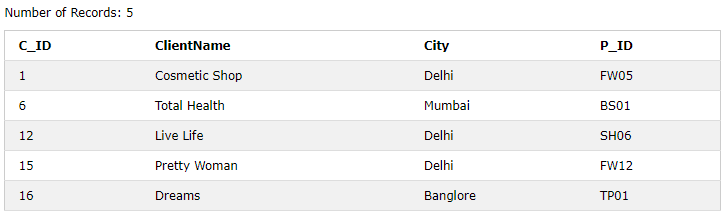
Consider the following tables Product and Client. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii)

**TABLE: Product**

A screenshot of a cell phone

Description automatically generated

**TABLE: Clients**



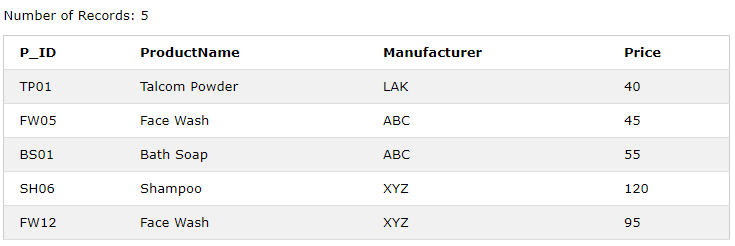
1. **SQL Commands**
2. **To display the details of those clients whose city is Delhi**
3. **To display the details of Products whose price is in the range of 50 to 100 (Both values included)**
4. **To display the ClientName, City from table Client, and ProductName and Price from table Products with their corresponding matching P\_ID.**
5. **To increase the Price of all Products by 10.**
6. **Output Queries**
7. **SELECT DISTINCT city FROM client;**
8. **SELECT manufacturer, Max(price), Min(price), Count (\*)   
   FROM   product   
   GROUP BY manufacturer;**
9. **SELECT clientname, productname   
   FROM   product, client;  
   WHERE client.p\_id = product.p\_id;**
10. **SELECT productname, price \* 4   
    FROM   product;**

**CREATING TABLES**

**PRODUCT**

CREATE TABLE Product (P\_ID varchar(10), ProductName char(30), Manufacturer char (30), Price int);

INSERT INTO Product VALUES('TP01',  'Talcom Powder', 'LAK',  40);   
INSERT INTO Product VALUES('FW05',  'Face Wash',  'ABC',  45);   
INSERT INTO Product VALUES('BS01',  'Bath Soap', 'ABC', 55);   
INSERT INTO Product VALUES('SH06', 'Shampoo', 'XYZ', 120);   
INSERT INTO Product VALUES('FW12', 'Face Wash', 'XYZ', 95);



**CLIENTS**

CREATE TABLE CLIENTS(C\_ID int, ClientName char(30), City char(30), P\_ID varchar(10));

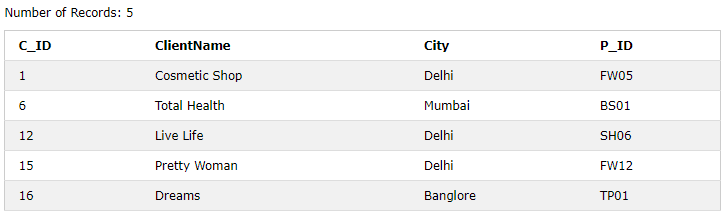
INSERT INTO Clients VALUES(1, 'Cosmetic Shop', 'Delhi', 'FW05');

INSERT INTO Clients VALUES(6, 'Total Health', 'Mumbai', 'BS01');

INSERT INTO Clients VALUES(12, 'Live Life', 'Delhi', 'SH06');

INSERT INTO Clients VALUES(15, 'Pretty Woman', 'Delhi', 'FW12');

INSERT INTO Clients VALUES(16, 'Dreams', 'Banglore', 'TP01');



**ANSWERS**

1. SELECT \*   
   FROM   clients  
   WHERE city = 'Delhi';



1. SELECT \*

FROM product

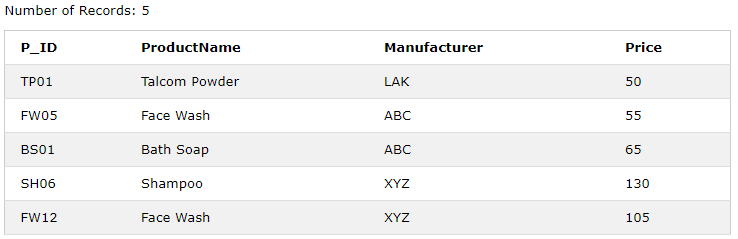
WHERE price BETWEEN 50 AND 100;



1. SELECT clientname, city, productname, price  
   FROM   clients, product   
   WHERE  clients.p\_id = product.p\_id;



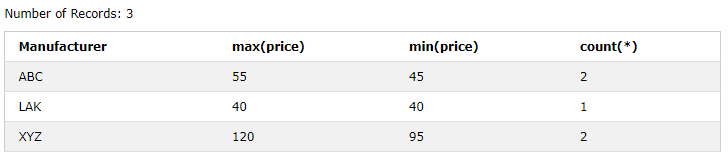
1. UPDATE product   
   SET    price = price + 10;



1. **SELECT DISTINCT city FROM client;**

****

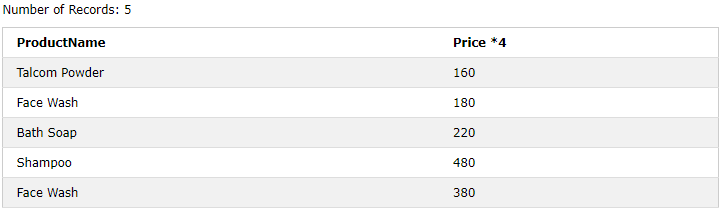
1. **SELECT manufacturer, Max(price), Min(price), Count (\*)   
   FROM   product   
   GROUP BY manufacturer;**

****

1. **SELECT clientname, productname   
   FROM   product, client;**



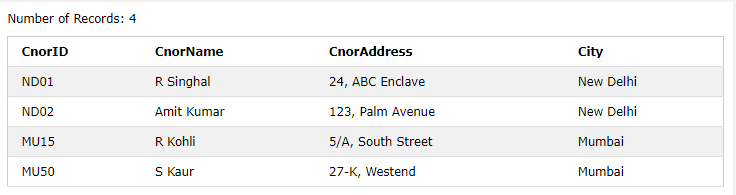
1. **SELECT productname, price \* 4   
   FROM   product;**



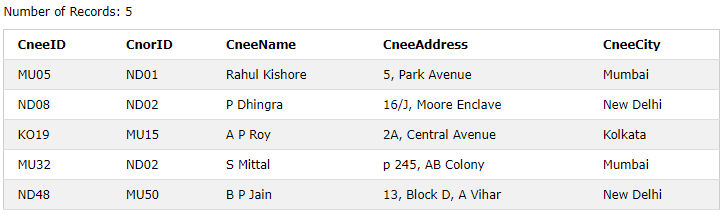
**QUESTION 2**

**Consider the following tables Product and Client. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii)**

**TABLE: Consignor**



**TABLE: Consignee**



1. **SQL Commands**
   1. To display the names of all Consignors from Mumbai
   2. To display the CneeID, CnorName, CnorAddress, CneeName, CneeAddress for corresponding Cnor\_ID
   3. To display consignee details in ascending order of CneeName
   4. To display number of consignors from each city
2. **Output Queries**
   1. SELECT DISTINCT City FROM Consignor;
   2. SELECT A.CnorName, B.CneeName FROM Consignor A, Consignee B

WHERE A.CnorID = B.CnorID and B.CneeCity = ‘Mumbai’;

* 1. SELECT CneeName, CneeAddress FROM Consignee

WHERE CneeCity NOT IN (‘Mumbai’, ‘Kolkata’);

* 1. SELECT CneeID, CneeName FROM Consignee

WHERE CnorID = ‘MU15’ OR CnorID = ‘ND01’;

**CREATING TABLES**

**CONSIGNOR**

CREATE TABLE Consignor

(CnorID varchar(10), CnorName char(30), CnorAddress varchar(30), City char(30));

INSERT INTO CONSIGNOR

VALUES('ND01', 'R Singhal', '24, ABC Enclave', 'New Delhi');

INSERT INTO CONSIGNOR

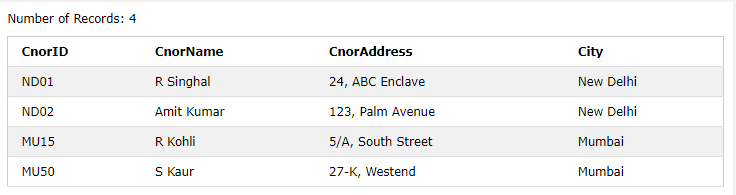
VALUES('ND02', 'Amit Kumar', '123, Palm Avenue', 'New Delhi');

INSERT INTO CONSIGNOR

VALUES('MU15', 'R Kohli', '5/A, South Street', 'Mumbai');

INSERT INTO CONSIGNOR

VALUES('MU50', 'S Kaur', '27-K Westend', 'Mumbai');



**CONSIGNEE**

CREATE TABLE Consignee(CneeID varchar(30), CnorID varchar(30), CneeName char(30), CneeAddress varchar(30), CneeCity char(30));

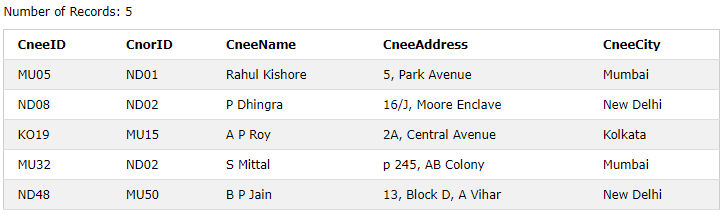
INSERT INTO Consignee values('MU05', 'ND01', 'Rahul Kishore', '5, Park Avenue', 'Mumbai');

INSERT INTO Consignee values('ND08', 'ND02', 'P Dhingra', '16/J, Moore Enclave', 'Mumbai');

INSERT INTO Consignee values('KO19', 'MU15', 'A P Roy', '2A, Central Avenue', 'Mumbai');

INSERT INTO Consignee values('MU32', 'ND02', 'S Mittal', 'p 245, AB Colony', 'Mumbai');

INSERT INTO Consignee values('ND48', 'MU50', 'B P Jain', '13, Block D, A Vihar', 'Mumbai');



**ANSWERS**

1. SELECT cnorname

FROM consignor

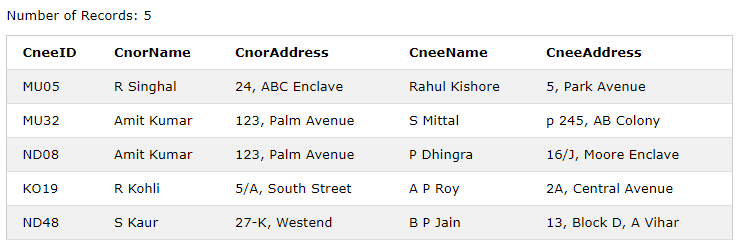
WHERE City = ‘Mumbai’;



1. SELECT CneeID, CnorName, CnorAddress, CneeName, CneeAddress

FROM Consignor, Consignee

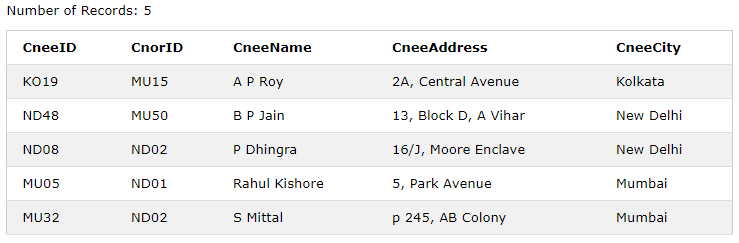
WHERE Consignor.CnorID = Consignee.CnorID;



1. SELECT \*

FROM Consignee

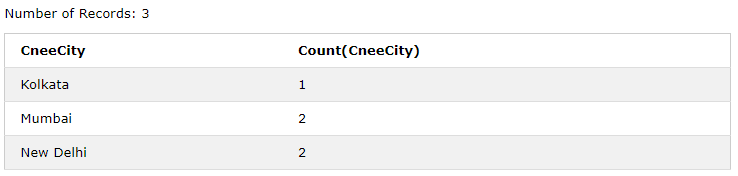
ORDER BY CneeName;



1. SELECT CneeCity, Count(CneeCity)

FROM Consignee

GROUP BY CneeCity

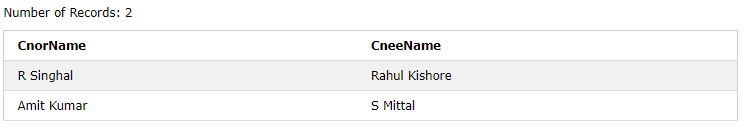


v. **SELECT DISTINCT City FROM Consignor;**



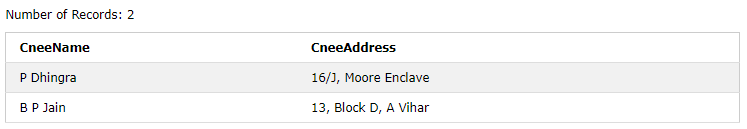
1. **SELECT A.CnorName, B.CneeName FROM Consignor A, Consignee B**

**WHERE A.CnorID = B.CnorID and B.CneeCity = ‘Mumbai’;**



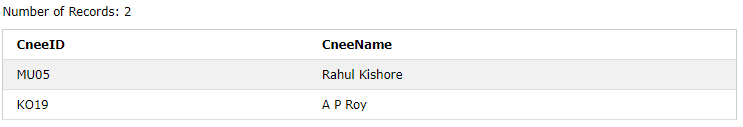
1. **SELECT CneeName, CneeAddress FROM Consignee**

**WHERE CneeCity NOT IN (‘Mumbai’, ‘Kolkata’);**



1. **SELECT CneeID, CneeName FROM Consignee**

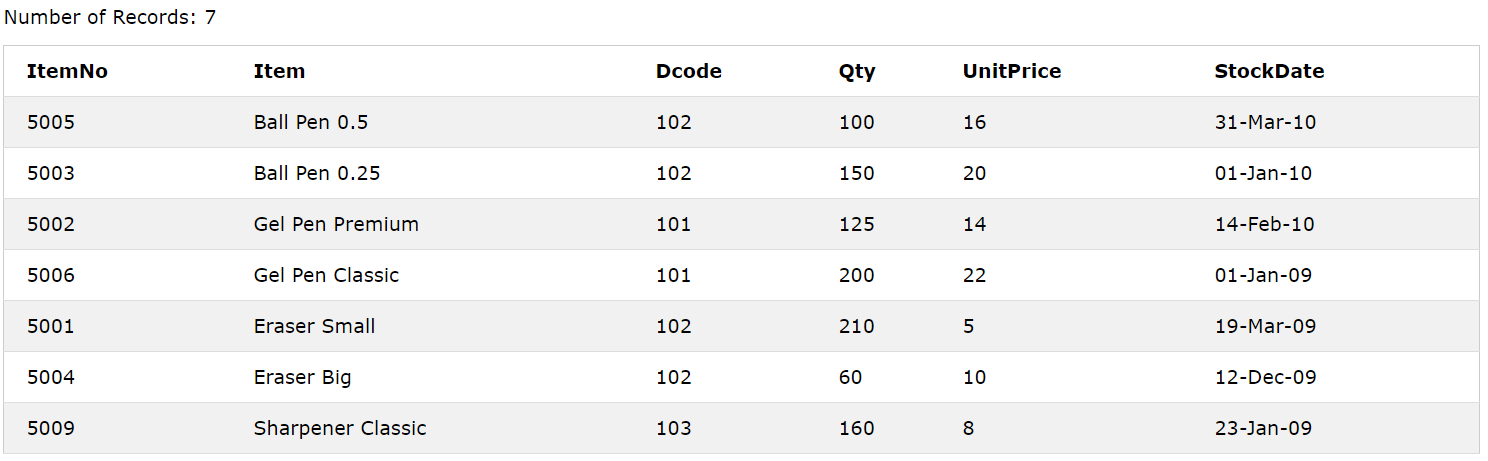
**WHERE CnorID = ‘MU15’ OR CnorID = ‘ND01’;**



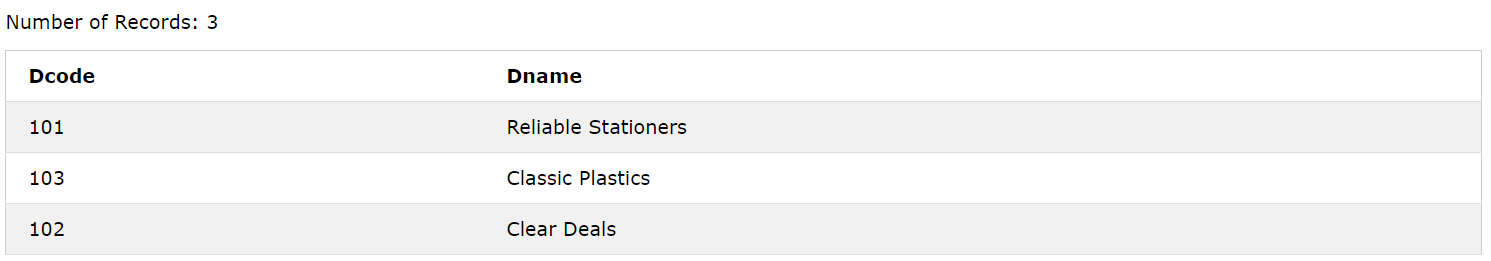
**QUESTION 3**

Consider the following tables Stock and Dealers and answer (a1) and (a2) parts of this question:

**Table: Stock**



**Table: Dealers**



1. **SQL Commands**
   1. To display details of all Items in the Stock table in ascending order of StockDate
   2. To display ItemNo and Item name of those items from Stock table whose UnitPrice is more than Rupees 10.
   3. To display the details of those items whose dealer code (Dcode) is 102 or Quantity in Stock (Qty) is more than 100 from the table Stock.
   4. To display Maximum UnitPrice of items for each dealer individually as per Dcode from the table stock.
2. **Output Queries**
3. SELECT COUNT(DISTINCT Dcode) FROM Stock;
4. SELECT Qty\*UnitPrice from Stock

WHERE ItemNo = 5006;

1. SELECT Item, Dname FROM Stock S, Dealer D

WHERE S.Dcode = D.Dcode AND ItemNo = 5004;

1. SELECT MIN(StockDate) FROM Stock;

**CREATING TABLES**

**STOCK**

CREATE TABLE Stock

(ItemNo int, Item varchar(40), Dcode int, Qty int, UnitPrice int, StockDate date);

INSERT INTO stock VALUES(5005, 'Ball Pen 0.5', 102, 100, 16, '31-Mar-10');

INSERT INTO stock VALUES(5003, 'Ball Pen 0.25', 102, 150, 20, '01-Jan-10');

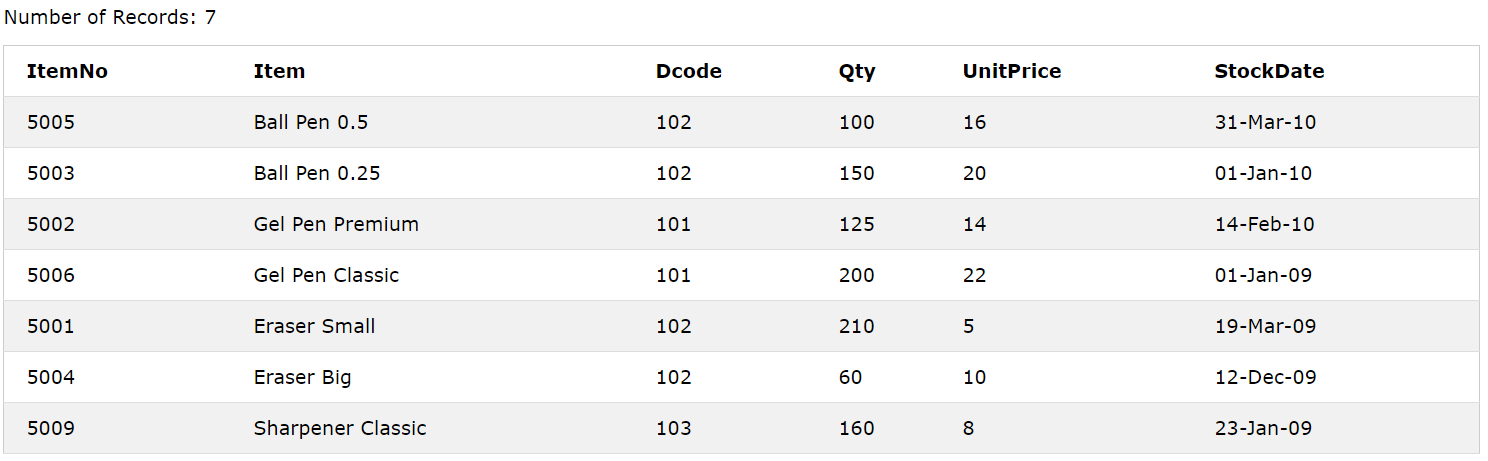
INSERT INTO stock VALUES(5002, 'Gel Pen Premium', 101, 125, 14, '14-Feb-10');

INSERT INTO stock VALUES(5006, 'Gel Pen Classic', 101, 200, 22, '01-Jan-09');

INSERT INTO stock VALUES(5001, 'Eraser Small', 102, 210, 5, '19-Mar-09');

INSERT INTO stock VALUES(5004, 'Eraser Big', 102, 60, 10, '12-Dec-09');

INSERT INTO stock VALUES(5009, 'Sharpener Classic', 103, 160, 8, '23-Jan-09');



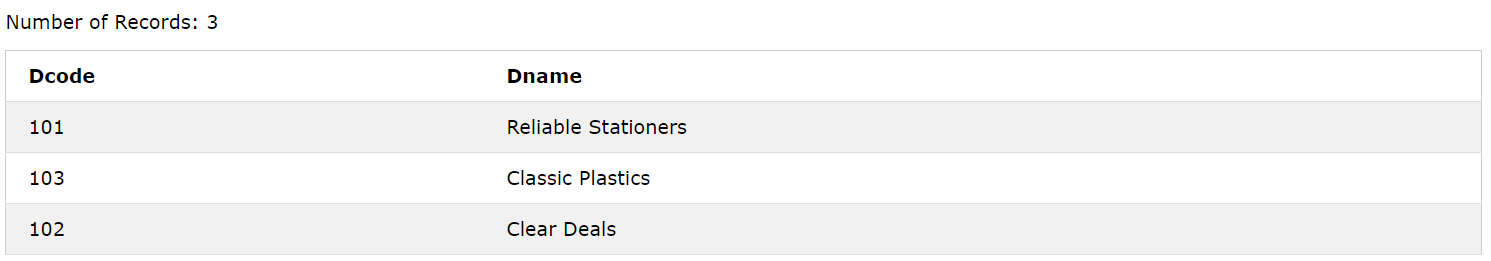
**DEALERS**

CREATE TABLE Dealers(Dcode int, Dname char(30));

INSERT INTO Dealers VALUES(101, 'Reliable Stationers');

INSERT INTO Dealers VALUES(103, 'Classic Plastics');

INSERT INTO Dealers VALUES(102, 'Clear Deals');



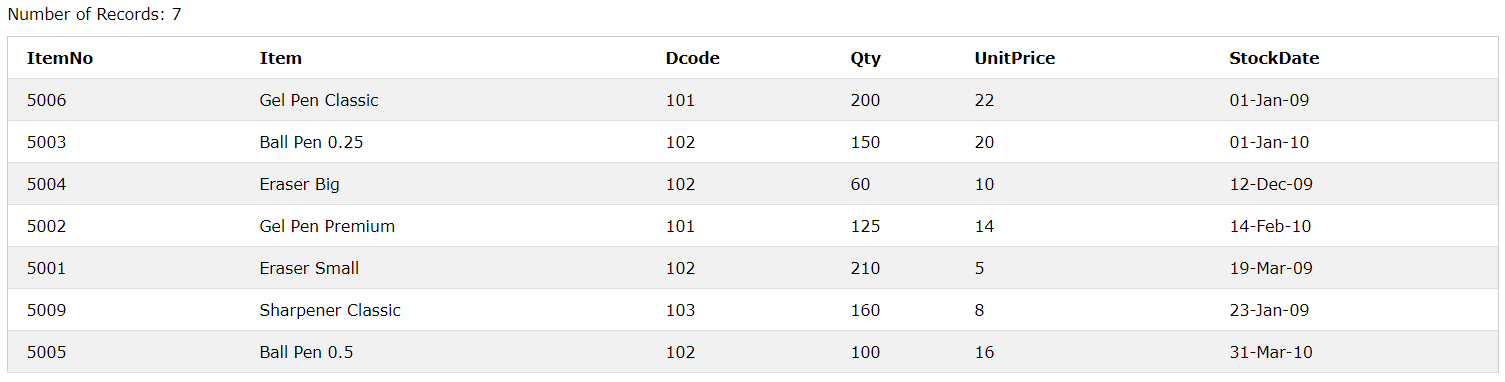
**ANSWERS**

1. SELECT \*

FROM Stock

ORDER BY Stockdate;

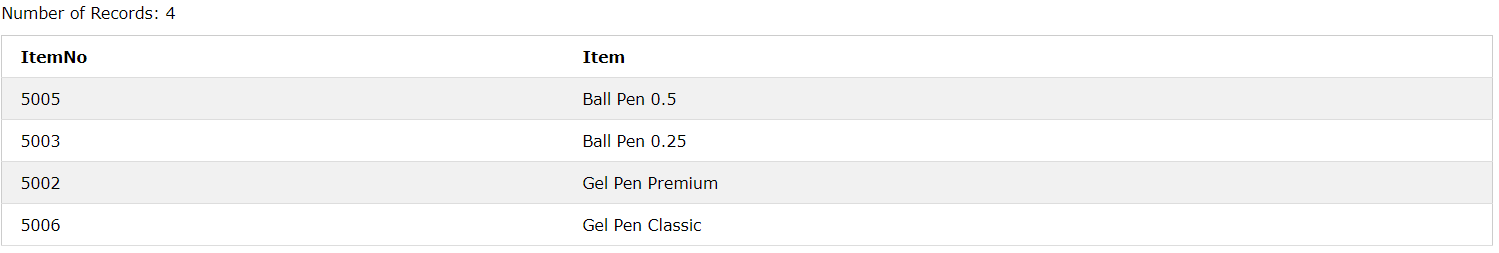
/\*date in YY/MM/DD format\*/



1. SELECT ItemNo, Item

FROM Stock

WHERE UnitPrice > 10;



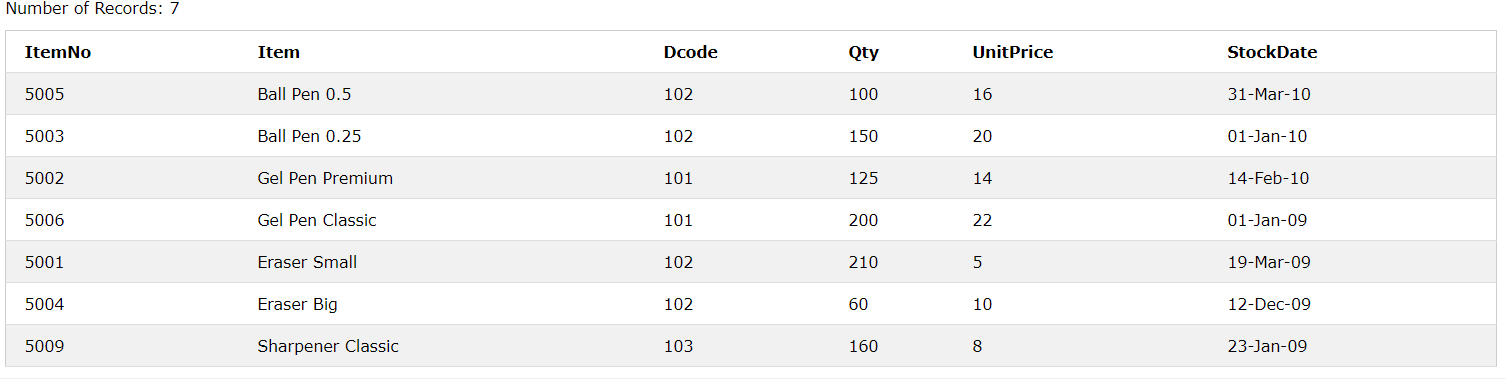
1. SELECT \*

FROM Stock

WHERE Dcode = 102

OR

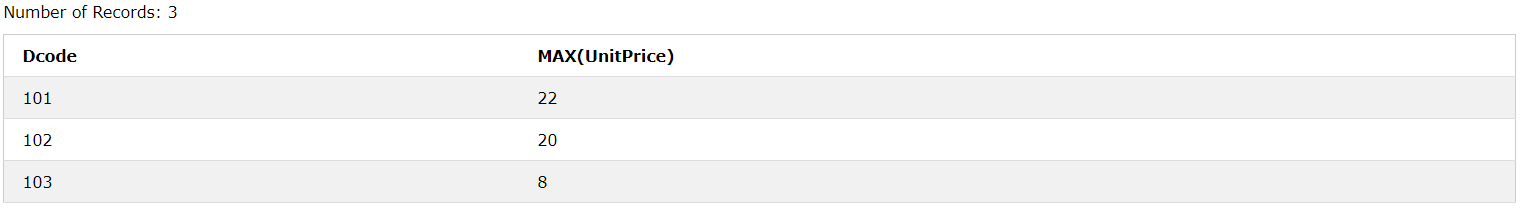
Qty >100;



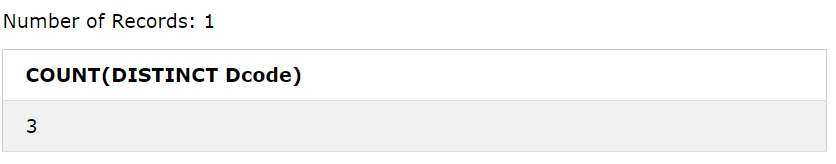
1. SELECT Dcode, MAX(UnitPrice)

FROM Stock

GROUP BY Dcode;

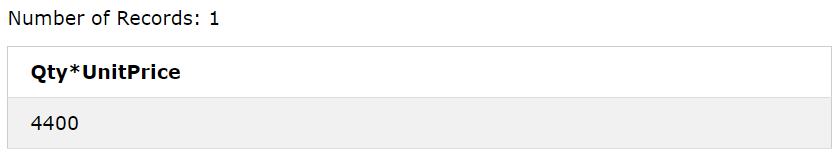


1. **SELECT COUNT(DISTINCT Dcode) FROM Stock;**



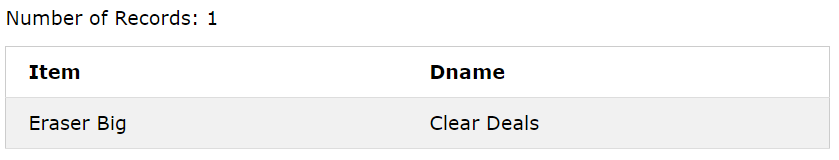
1. **SELECT Qty\*UnitPrice from Stock**

**WHERE ItemNo = 5006;**



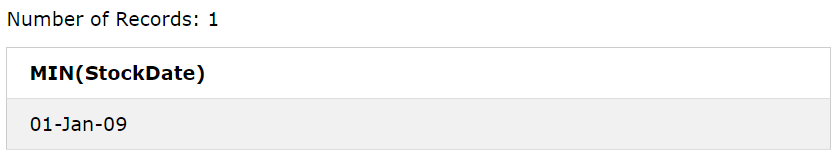
1. **SELECT Item, Dname FROM Stock S, Dealer D**

**WHERE S.Dcode = D.Dcode AND ItemNo = 5004;**



1. **SELECT Item, Dname FROM Stock S, Dealer D**

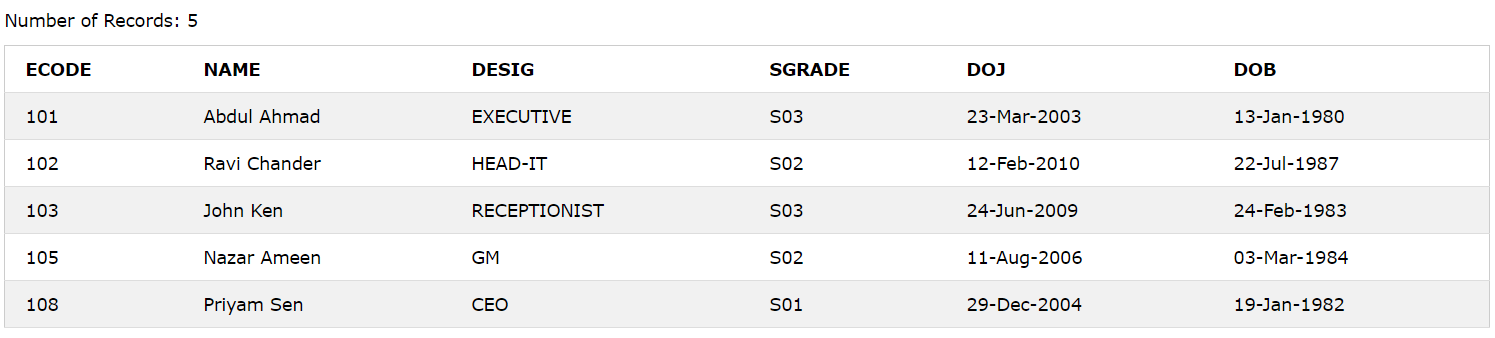
**WHERE S.Dcode = D.Dcode AND ItemNo = 5004;**



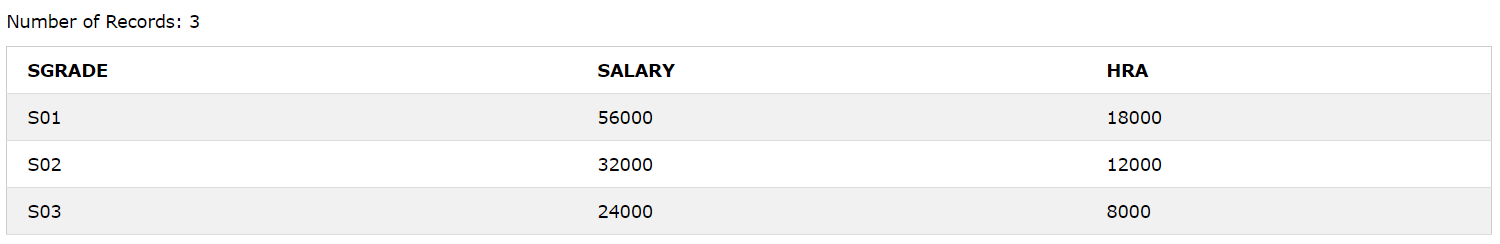
**QUESTION 4**

Consider the following tables EMPLOYEE and SALGRADE and answer (A) and (B) parts of this question.

**Table: EMPLOYEE**



**Table: SALGRADE**



1. **SQL Commands**
2. To display the details of all Employees in descending order of DOJ.
3. To display NAME and DESIG of those Employees, whose SALGRADE is either S02 or S03.
4. To display the content of all the employees table, whose DOJ is in between ’09-Feb-2006’ and ’08-Aug-2009’.
5. To add a new row with the following: 19, ‘Harish Roy’, ‘HEAD-IT’, ‘S02’, ‘09-Sep-2007’, ‘21-Apr-1983’.
6. **Output Queries**
7. SELECT COUNT (Sgrade), Sgrade from employee group by sgrade;
8. SELECT MIN(DOB), MAX(DOJ) FROM Employee;
9. SELECT Name, Salary from Employee E,

SALGRADE S WHERE E.Sgrade = S.sgrade and E.Ecode<103;

1. SELECT SGRADE, SALARY + HRA FROM SALGRADE WHERE SGRADE = ‘S02’;

**CREATING TABLES**

**EMPLOYEE**

CREATE TABLE Employee

(

ECODE int,

NAME char(30),

DESIG char(30),

SGRADE varchar(30),

DOJ date,

DOB date

);

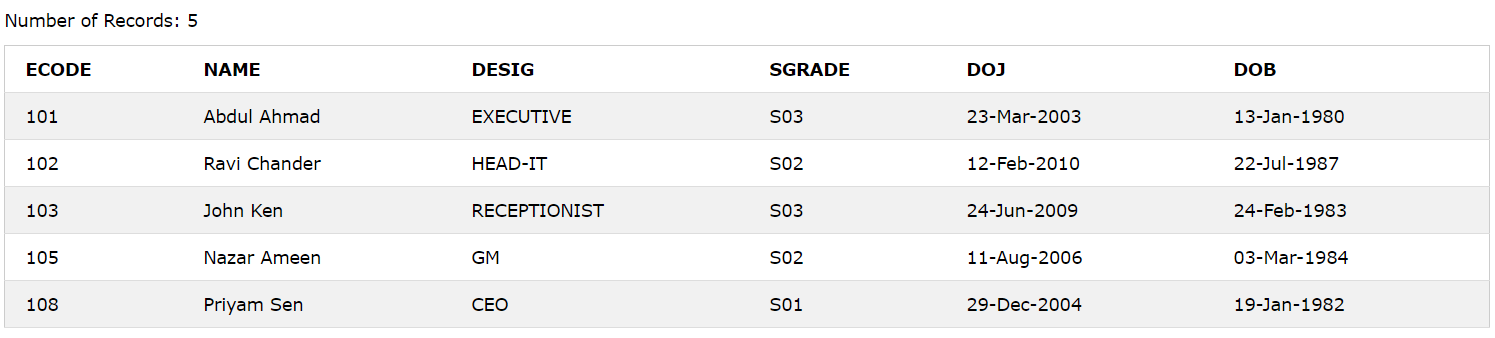
INSERT INTO Employee VALUES(101, 'Abdul Ahmad', 'EXECUTIVE', 'S03', '23-Mar-2003', '13-Jan-1980');

INSERT INTO Employee VALUES(102, 'Ravi Chander', 'HEAD-IT', 'S02', '12-Feb-2010', '22-Jul-1987');

INSERT INTO Employee VALUES(103, 'John Ken', 'RECEPTIONIST', 'S03', '24-Jun-2009', '24-Feb-1983');

INSERT INTO Employee VALUES(105, 'Nazar Ameen', 'GM', 'S02', '11-Aug-2006', '03-Mar-1984');

INSERT INTO Employee VALUES(108, 'Priyam Sen', 'CEO', 'S01', '29-Dec-2004', '19-Jan-1982');



**SALGRADE**

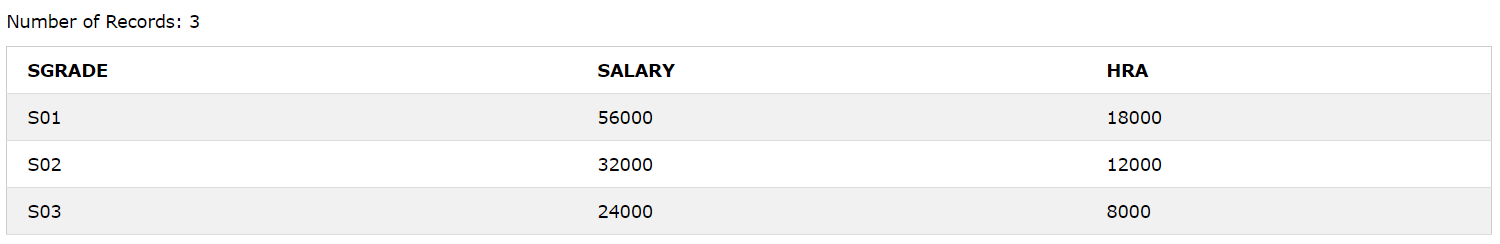
CREATE TABLE SALGRADE

( SGRADE varchar(20), SALARY int, HRA int );

INSERT INTO SALGRADE VALUES('S01', 56000, 18000);

INSERT INTO SALGRADE VALUES('S02', 32000, 12000);

INSERT INTO SALGRADE VALUES('S03', 24000, 8000);



**ANSWERS**

1. SELECT \*

FROM Employee

ORDER BY doj DESC;



1. SELECT name, desig

FROM Employee, SALGRADE

WHERE SALGRADE.sgrade = Employee.sgrade;

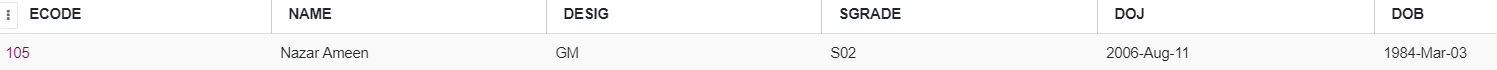


1. SELECT \*

FROM Employee

WHERE doj

BETWEEN '2006-Aug-09' and '2009-Aug-08';



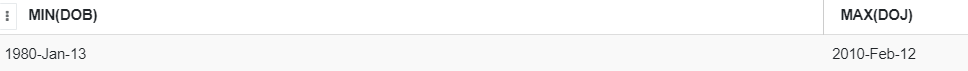
1. INSERT INTO Employee VALUES(19, 'Harish Roy', 'HEAD-IT', 'S02', '09-Sept-2007', '21-Apr-1983');



1. **SELECT COUNT (Sgrade), Sgrade from employee group by sgrade;**

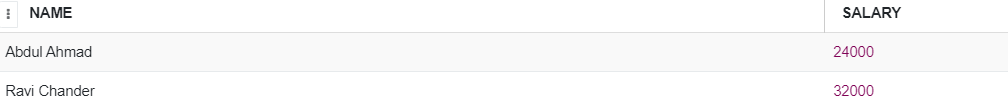


1. **SELECT MIN(DOB), MAX(DOJ) FROM Employee;**

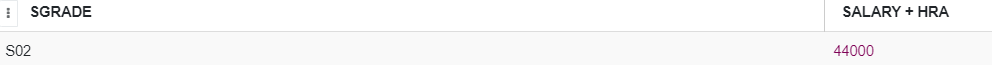


1. **SELECT Name, Salary from Employee E,**

**SALGRADE S WHERE E.Sgrade = S.sgrade and E.Ecode<103;**



1. **SELECT SGRADE, SALARY + HRA FROM SALGRADE WHERE SGRADE = ‘S02’;**



**QUESTION 5**

Consider the following tables STORES AND ITEM and answer (A) and (B) parts of this question

**Table: STORE**

A screenshot of a cell phone

Description automatically generated

**Table: ITEM**

A screenshot of a cell phone

Description automatically generated

1. **SQL Commands**
   1. **To display Iname and Price of all the Items in ascending order of their price.**
   2. **To display SNo and SName of all Stores located in CP.**
   3. **To display minimum and maximum price of each IName from the table Item.**
   4. **To display IName, Price of all items and their respective SName where they are available.**
2. **Output Queries**
   1. **SELECT DISTINCT INAME FROM ITEM WHERE PRICE >= 5000;**
   2. **SELECT AREA, COUNT (\*) FROM STORE GROUP BY AREA;**
   3. **SELECT COUNT(DISTINCT AREA) FROM STORE;**
   4. **SELECT INAME, PRICE \* 0.05 DISCOUNT FROM ITEM WHERE SNO IN (‘S02’, ‘S03’)**

**CREATING TABLES**

**STORE**

CREATE table STORE(Sno varchar(3) not null primary key,SName char(40),Area char(20));

INSERT INTO STORE VALUES ('S01','ABC Copmutronics','GK II');

INSERT INTO STORE VALUES ('S02','All Infotech Media','CP');

INSERT INTO STORE VALUES ('S03','Tech Shoppe','Nehru Place');

INSERT INTO STORE VALUES ('S04','Geeks Techo Soft','Nehru Place');

INSERT INTO STORE VALUES ('S05','Hitech Tech Store','CP');

A screenshot of a cell phone

Description automatically generated

**ITEM**

CREATE table ITEM(INo varchar(3) not null primary key,IName char(20),Price int,SNo varchar(3));

INSERT INTO ITEM VALUES ('T01','Mother Board',12000,'S01');

INSERT INTO ITEM VALUES ('T02','Hard Disk',5000,'S01');

INSERT INTO ITEM VALUES ('T03','Keyborad',500,'S02');

INSERT INTO ITEM VALUES ('T04','Mouse',300,'S01');

INSERT INTO ITEM VALUES ('T05','Mother Board',13000,'S02');

INSERT INTO ITEM VALUES ('T06','Keyboard',400,'S03');

INSERT INTO ITEM VALUES ('T07','LCD',6000,'S04');

INSERT INTO ITEM VALUES ('T08','LCD',5500,'S05');

INSERT INTO ITEM VALUES ('T09','Mouse',350,'S05');

INSERT INTO ITEM VALUES ('T10','Hard Disk',4500,'S03');

A screenshot of a cell phone

Description automatically generated

**ANSWERS**

1. SELECT IName,Price FROM ITEM

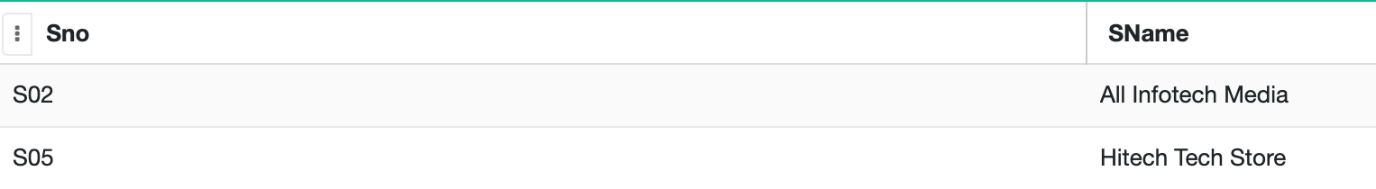
ORDER BY Price;

A screenshot of a cell phone

Description automatically generated

1. SELECT SNo, SName FROM STORE

WHERE Area='CP';



1. SELECT IName, MIN(Price), MAX(Price) FROM ITEM

GROUP BY IName;

A screenshot of a cell phone

Description automatically generated

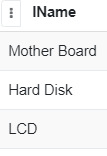
1. SELECT IName,Price,SName FROM ITEM,STORE

WHERE ITEM.SNo=STORE.SNo;

A screenshot of a cell phone

Description automatically generated

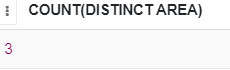
1. **SELECT DISTINCT INAME FROM ITEM WHERE PRICE >= 5000;**



1. **SELECT AREA, COUNT (\*) FROM STORE GROUP BY AREA;**



1. **SELECT COUNT(DISTINCT AREA) FROM STORE;**



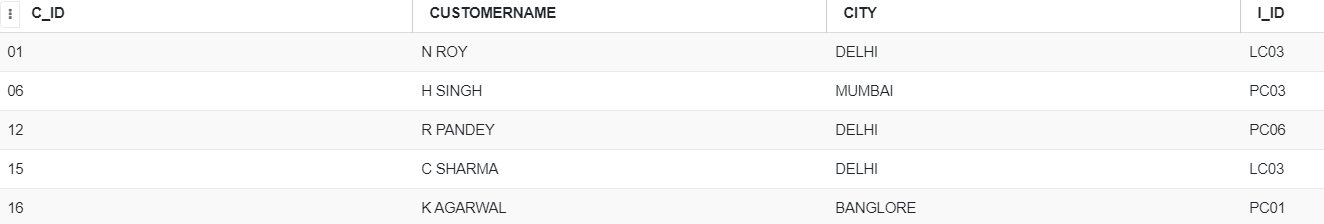
1. **SELECT INAME, PRICE \* 0.05 DISCOUNT FROM ITEM WHERE SNO IN (‘S02’, ‘S03’)**



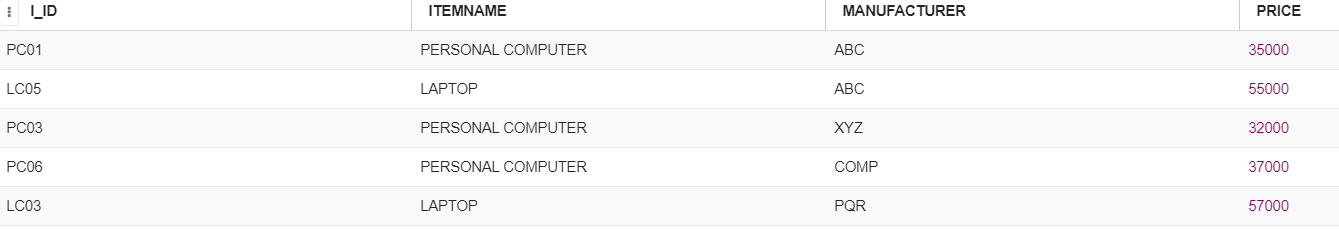
**QUESTION 6**

Consider the following tables Item and Customer. Write answer for the statement (i) to (viii).

**TABLE: CUSTOMER**



**TABLE: ITEM**



**A. SQL Commands**

1. To display the details of those customers whose city is Delhi.
2. To display the details of item whose price is in the range of 35000 to 55000 (both values included).
3. To display the customer name, city from table Customer, and ItemName and Price from table Item, with their corresponding i\_ID.
4. To increase the price of all items by 1000 in the table Item

**B. Output Queries**

* 1. SELECT DISTINCT CITY FROM CUSTOMER;
  2. SELECT ITEMNAME, MAX(PRICE), COUNT(\*) FROM ITEM GROUP BY ITEMNAME;
  3. SELECT CustomerName, Manufacturer from Item, Customer WHERE Item.I\_id=Customer.I\_id;
  4. SELECT ItemName,Price \* 100 From Item WHERE Manufacturer=’ABC’;

**CREATING TABLES**

**CUSTOMER**

CREATE TABLE CUSTOMER (C\_ID VARCHAR(10), CUSTOMERNAME CHAR(20), CITY CHAR (20), I\_ID VARCHAR(10));

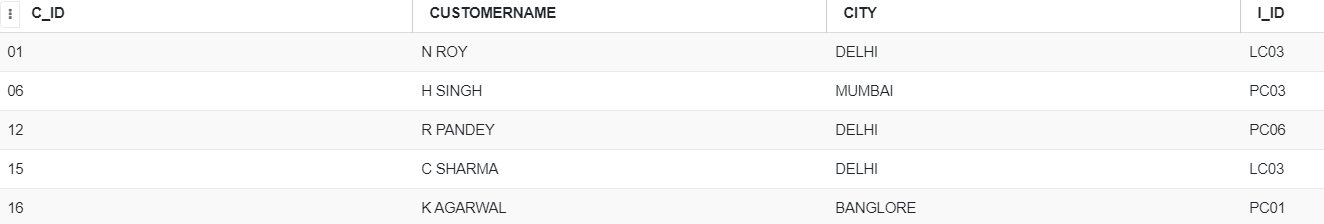
INSERT INTO CUSTOMER VALUES ('01', 'N ROY', 'DELHI', ‘LC03’);

INSERT INTO CUSTOMER VALUES ('06', 'H SINGH', 'MUMBAI', ‘PC03');

INSERT INTO CUSTOMER VALUES ('12', 'R PANDEY', 'DELHI', ‘PC06’);

INSERT INTO CUSTOMER VALUES ('15', 'C SHARMA', 'DELHI', ‘LC03');

INSERT INTO CUSTOMER VALUES ('16', 'K AGARWAL', 'BANGLORE', ‘PC01');



**ITEM**

CREATE TABLE ITEM (I\_ID VARCHAR(10), ITEMNAME CHAR(20), MANUFACTURER CHAR (20), PRICE DECIMAL)

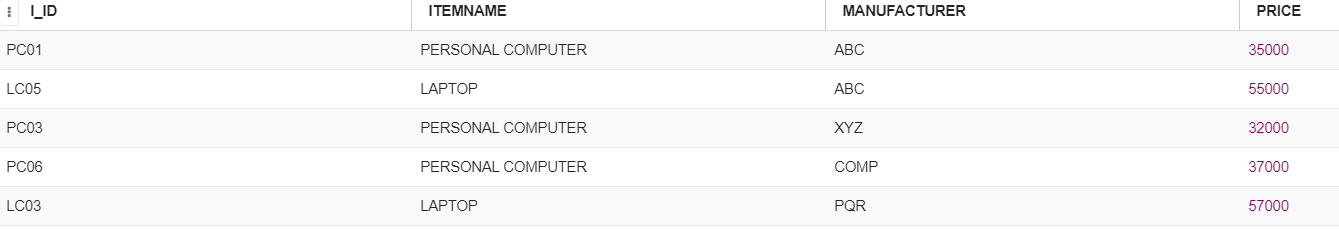
INSERT INTO ITEM VALUES ('PC01', 'PERSONAL COMPUTER', 'ABC', 35000)

INSERT INTO ITEM VALUES ('LC05', 'LAPTOP', 'ABC', 55000)

INSERT INTO ITEM VALUES ('PC03', 'PERSONAL COMPUTER', 'XYZ', 32000)

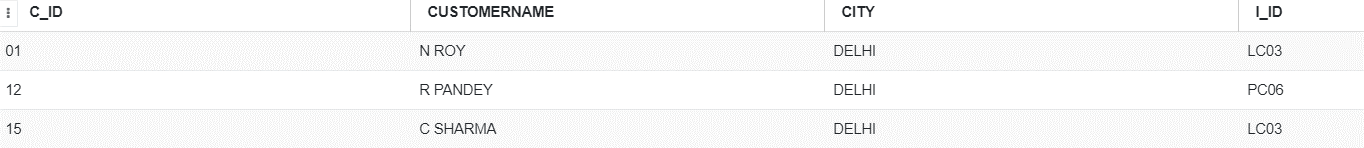
INSERT ITEM VALUES ('PC06', 'PERSONAL COMPUTER', 'COMP', 37000)

INSERT INTO ITEM VALUES ('LC03', 'LAPTOP', 'PQR', 57000)

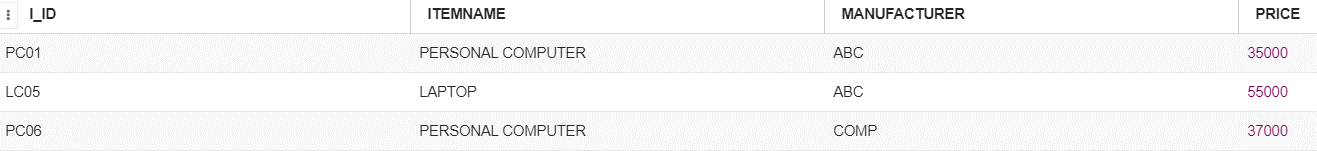


**ANSWERS**

1. SELECT \* FROM CUSTOMER WHERE CITY=‘DELHI’;



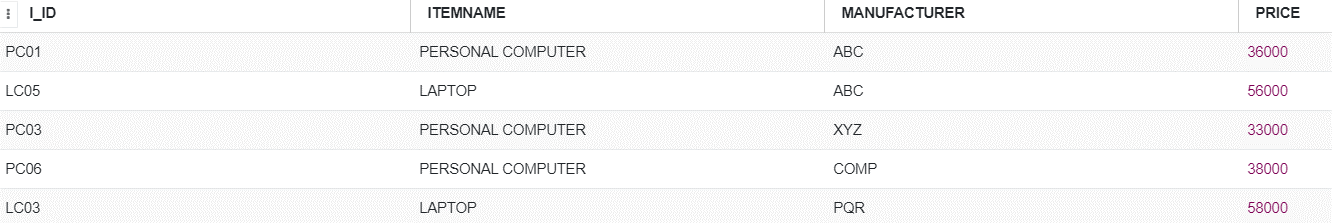
ii) SELECT \* FROM ITEM WHERE PRICE BETWEEN 35000 AND 55000;



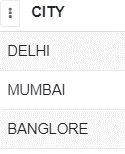
iii) SELECT CUSTOMERNAME, CITY, ITEMNAME, PRICE FROM CUSTOMER, ITEM WHERE CUSTOMER.I\_ID=ITEM.I\_ID;



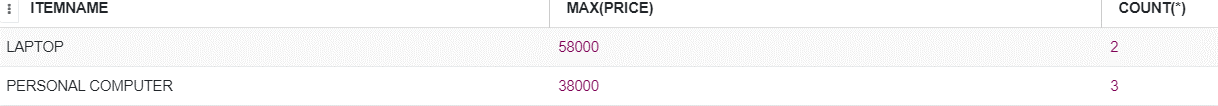
iv) UPDATE ITEM SET PRICE=PRICE+1000



1. **SELECT DISTINCT CITY FROM CUSTOMER;**



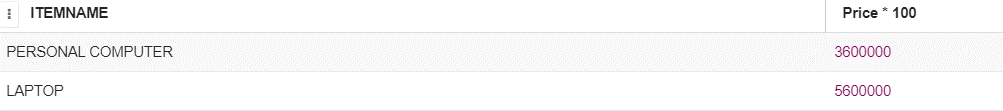
1. **SELECT ITEMNAME, MAX(PRICE), COUNT(\*) FROM ITEM GROUP BY ITEMNAME;**



1. **SELECT CustomerName, Manufacturer from Item, Customer WHERE Item.I\_id=Customer.I\_id;**



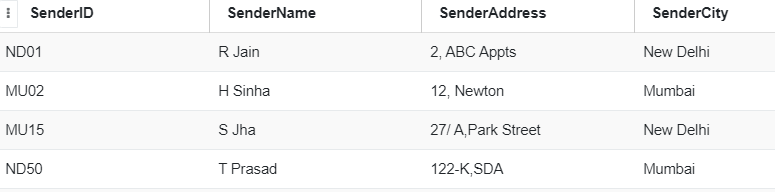
viii) **SELECT ItemName,Price \* 100 From Item WHERE Manufacturer='ABC';**



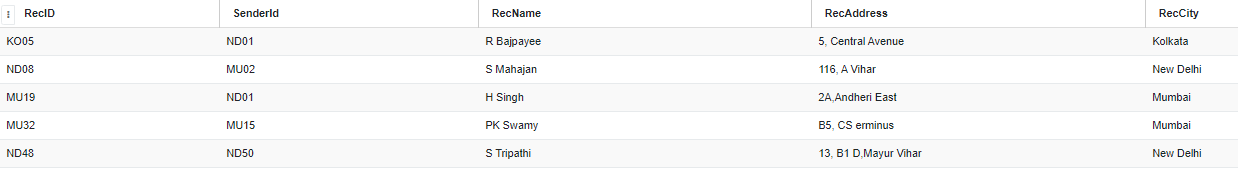
**QUESTION 7**

Consider the following. Answer (A) and (B) parts of this question

**TABLE: SENDER**



**TABLE: RECIPIENT**



1. **SQL Commands**
   1. To display the names of all senders from Mumbai.
   2. To display the recID, senderName, senderAddress, RecName, RecAddress for every recipt.
   3. To display the sender details in ascending order of SenderName.
   4. To display number of Recipients from each city.
2. **Output Queries** 
   1. SELECT DISTINCT SenderCity FROM Sender;
   2. SELECT A.SenderName A, B.RecName FROM Sender A, Recipient B

WHERE A.SenderID=B. SenderID AND B.RecCity=’Mumbai’;

* 1. SELECT RecName,RecAddress FROMRecipient WHERE RecCity Not IN (‘Mumbai’,Kolkata’);
  2. SELECT RecID, RecName FROM Recipient WHERE SenderID = ‘MU02’ OR SenderID = ‘ND50’;

**CREATING TABLES**

**SENDER**

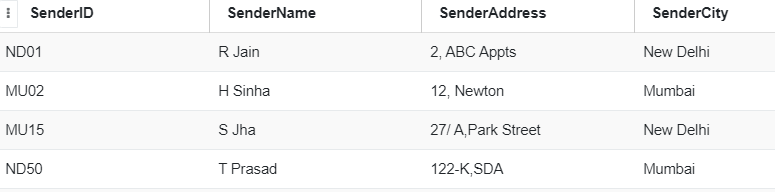
CREATE TABLE SENDER (SenderID varchar(5), SenderName char(30), SenderAddress varchar(30), SenderCity char(30));

INSERT INTO SENDER VALUES(‘ND01’ , ‘R Jain’, ‘2, ABC Appts’, ‘New Delhi’);

INSERT INTO SENDER VALUES (‘MU02’, ‘H Sinha’, ‘12, Newton’, ‘Mumbai’);

INSERT INTO SENDER VALUES(‘MU15 , ‘S Jha’ , ‘27/A, Park Street’, ‘New Delhi’);

INSERT INTO SENDER VALUES(‘ND50’ , ‘T Prasad’, ‘122-K, SDA’, ‘Mumbai’);



**RECIPIENT**

CREATE TABLE Recipient( RecID varchar(5), SenderId varchar(5), RecName char(30), RecAddress char(30), RecCity char(30));

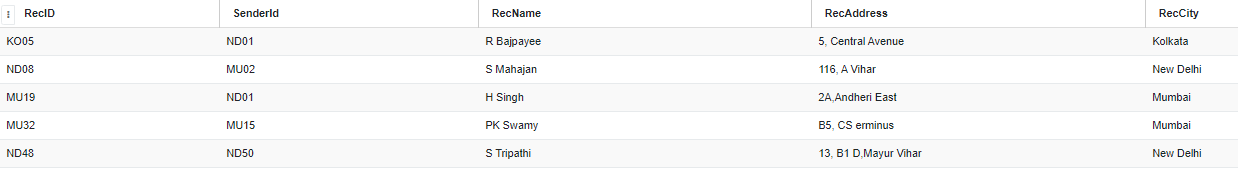
INSERT INTO Recipient VALUES('KO05', 'ND01', 'R Bajpayee', '5, Central Avenue', 'Kolkata');

INSERT INTO Recipient VALUES('ND08', 'MU02', 'S Mahajan', '116, A Vihar', 'New Delhi’);

INSERT INTO Recipient VALUES('MU19', 'ND01', 'H Singh', '2A,Andheri East', ‘Mumbai;);

INSERT INTO Recipient VALUES ('MU32', 'MU15', 'PK Swamy', 'B5, CS erminus', ‘Mumbai’);

INSERT INTO Recipient VALUES ('ND48', 'ND50', 'S Tripathi', '13, B1 D,Mayur Vihar', 'New Delhi');



**ANSWERS**

1. SELECT \* FROM Sender

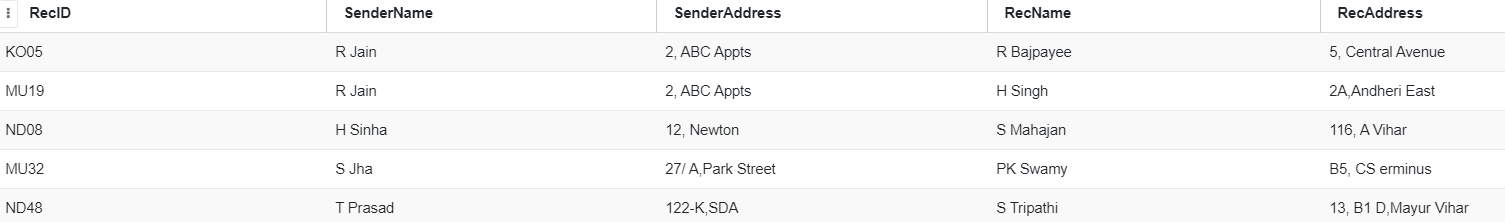
WHERE SenderCity ='Mumbai';



1. SELECT recID, SenderName, SenderAddress, RecName, RecAddress

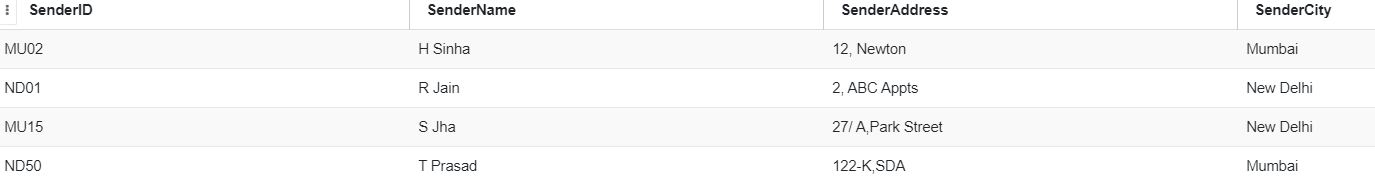
FROM Sender, Recipient

WHERE Sender.Senderid=Recipient.SenderId;



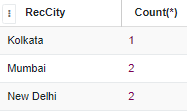
1. SELECT \* FROM Sender

ORDER BY SenderName;

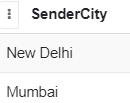


1. SELECT RecCity,Count(\*) FROM Recipient

GROUP BY RecCity;

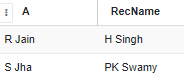


* 1. **SELECT DISTINCT SenderCity FROM Sender;**



* 1. **SELECT A.SenderName A, B.RecName FROM Sender A, Recipient B**

**WHERE A.SenderID=B. SenderID AND B.RecCity=’Mumbai’;**



* 1. **SELECT RecName,RecAddress FROM Recipient WHERE RecCity Not IN (‘Mumbai’,Kolkata’);**



* 1. **SELECT RecID, RecName FROM Recipient WHERE SenderID = ‘MU02’ OR SenderID = ‘ND50’;**

