**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: ";

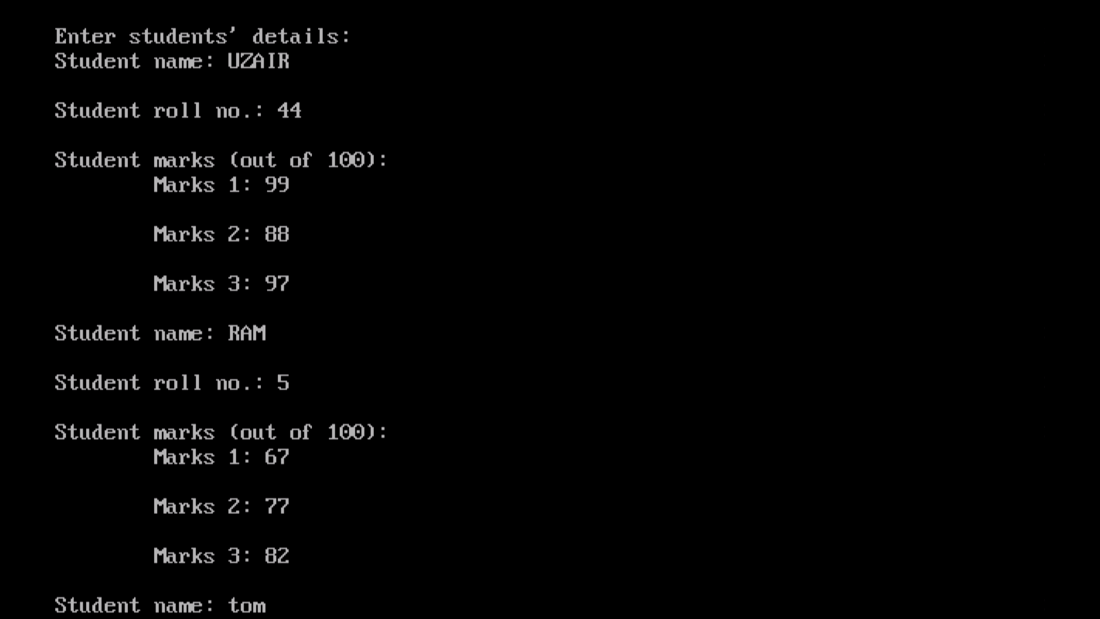
cout<<stud[i].sums;

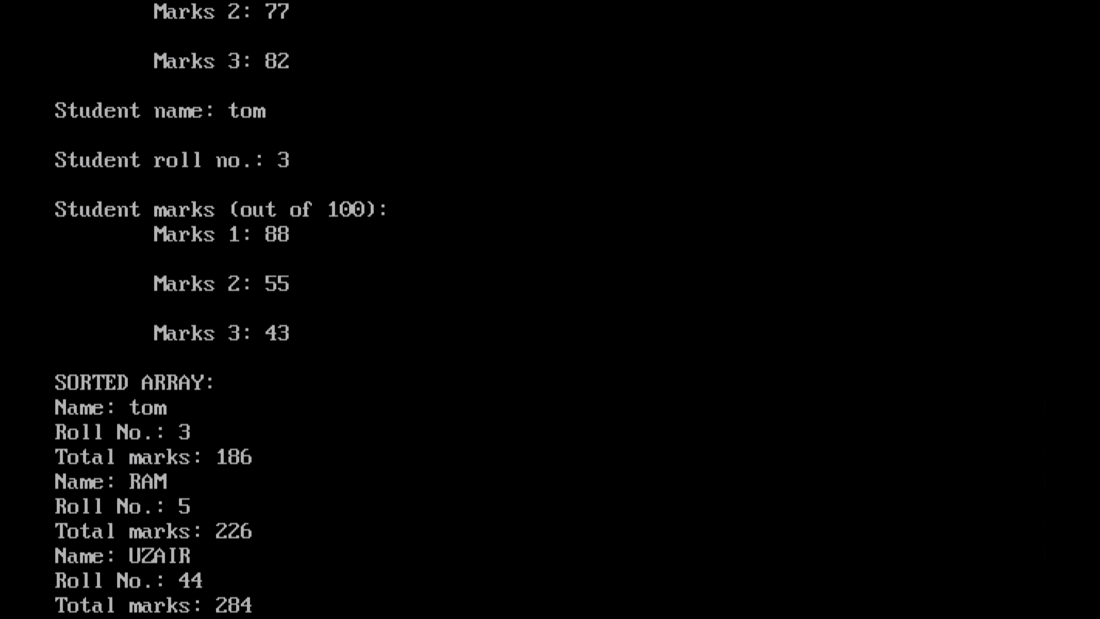
}

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;

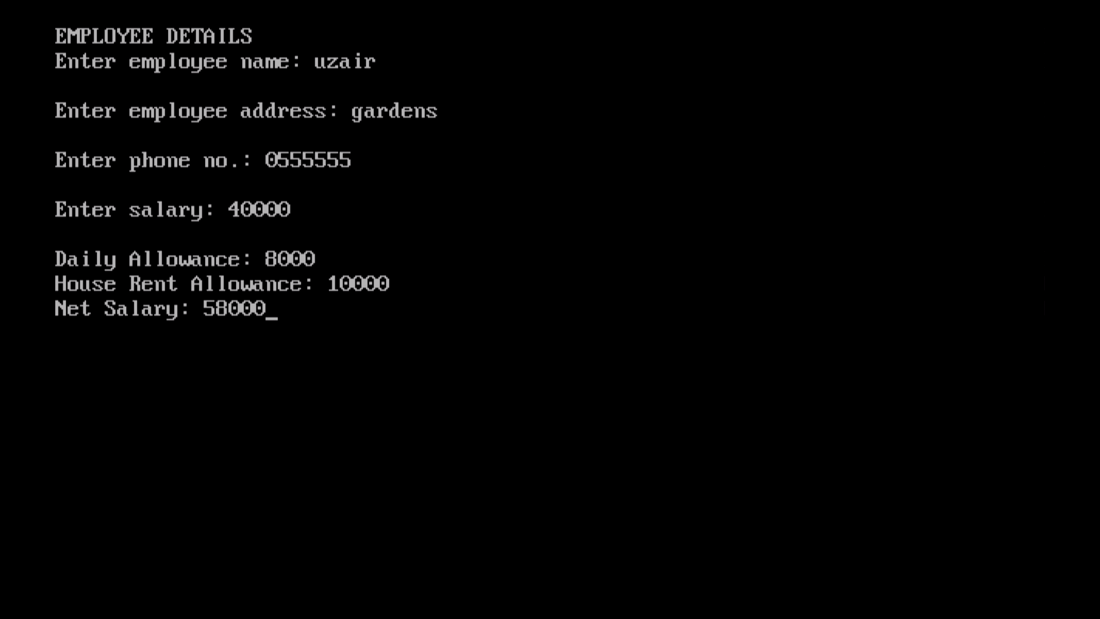
cout<<"\nNet Salary: ";

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: ";

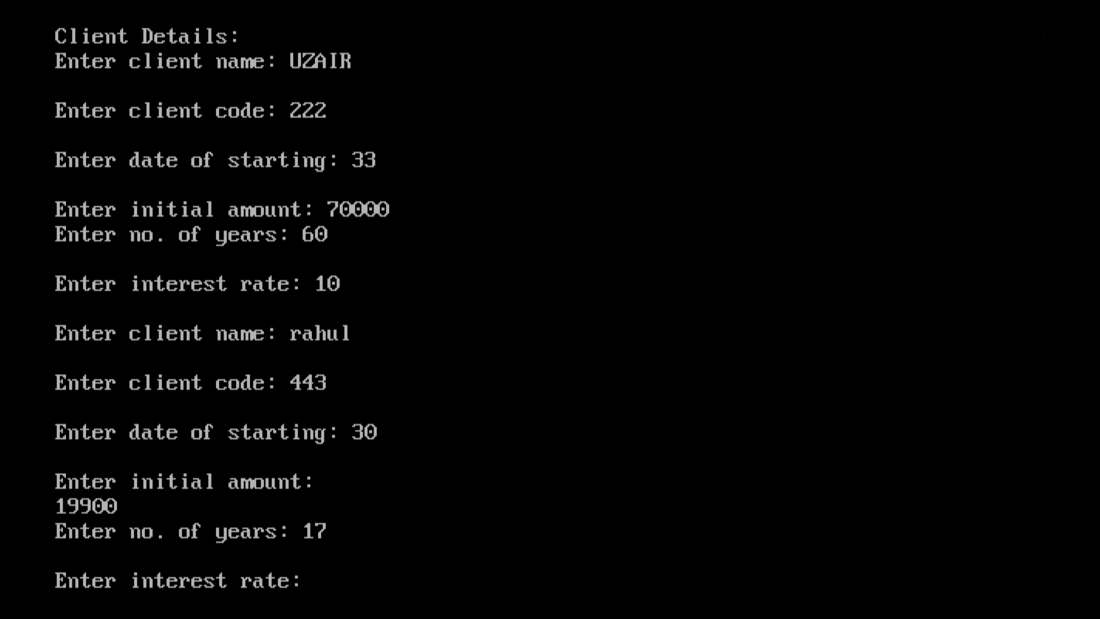
cout<<comp;

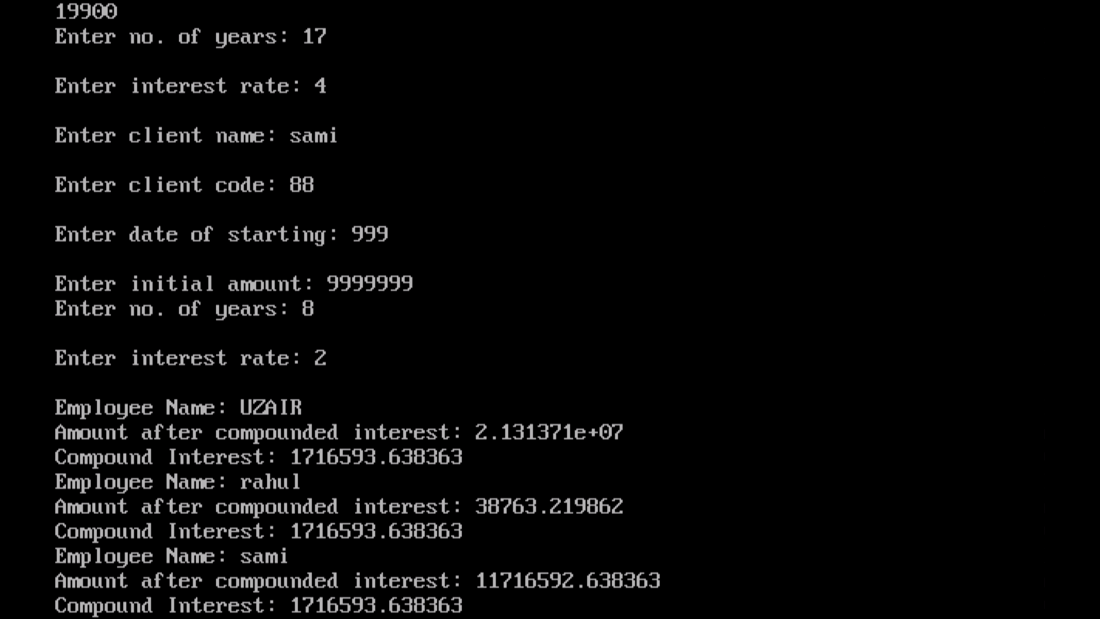
}

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];

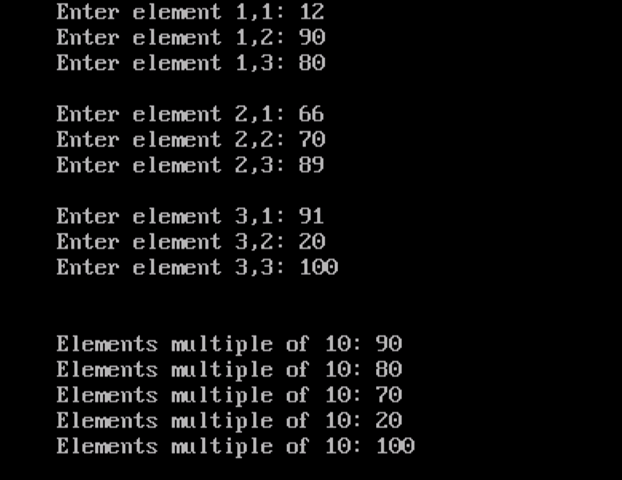
}

}

}

}

**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;

}

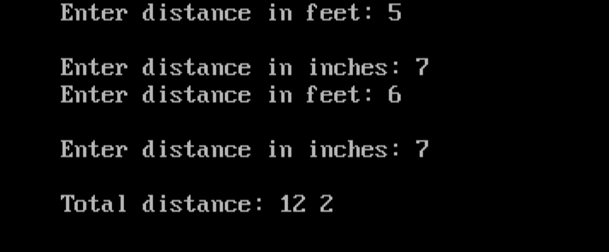
cout<<"\nTotal distance: ";

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;

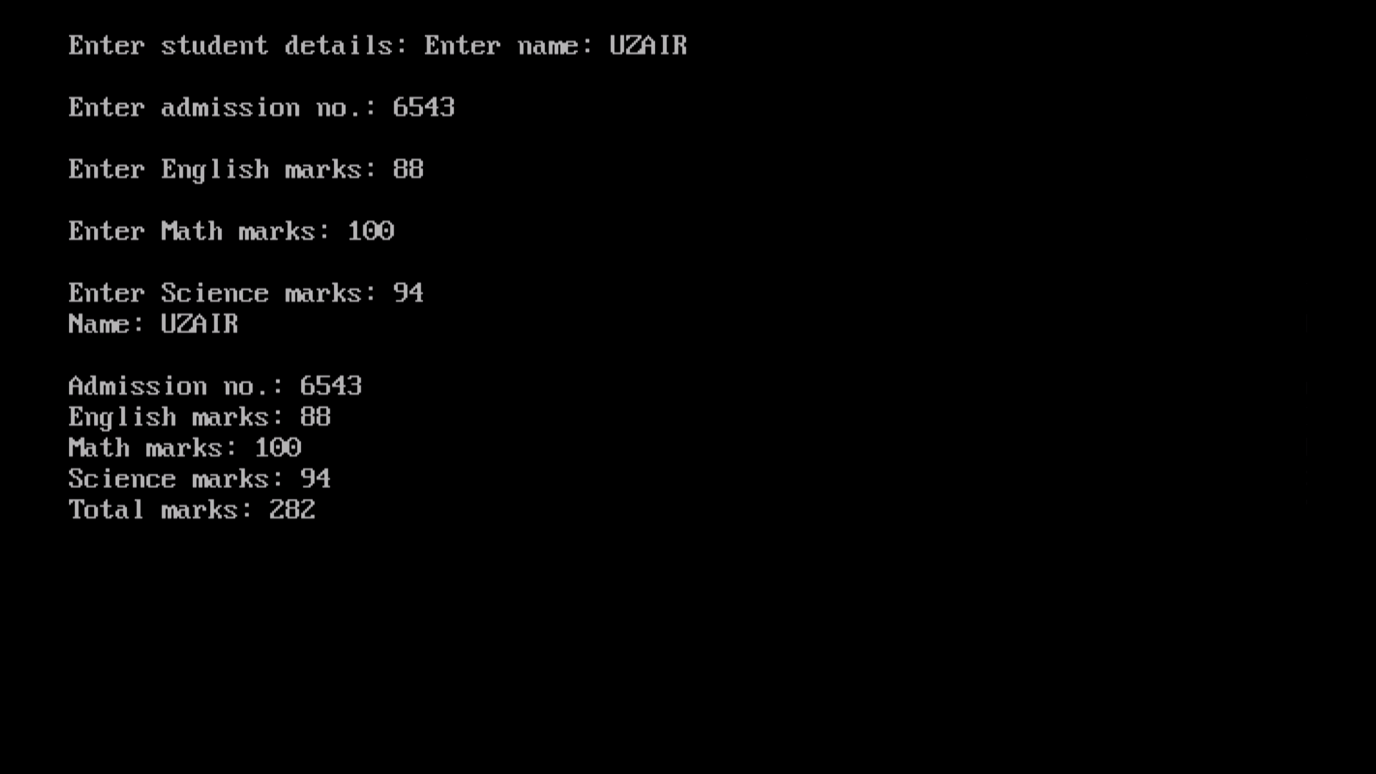
s1.Takedata();

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;

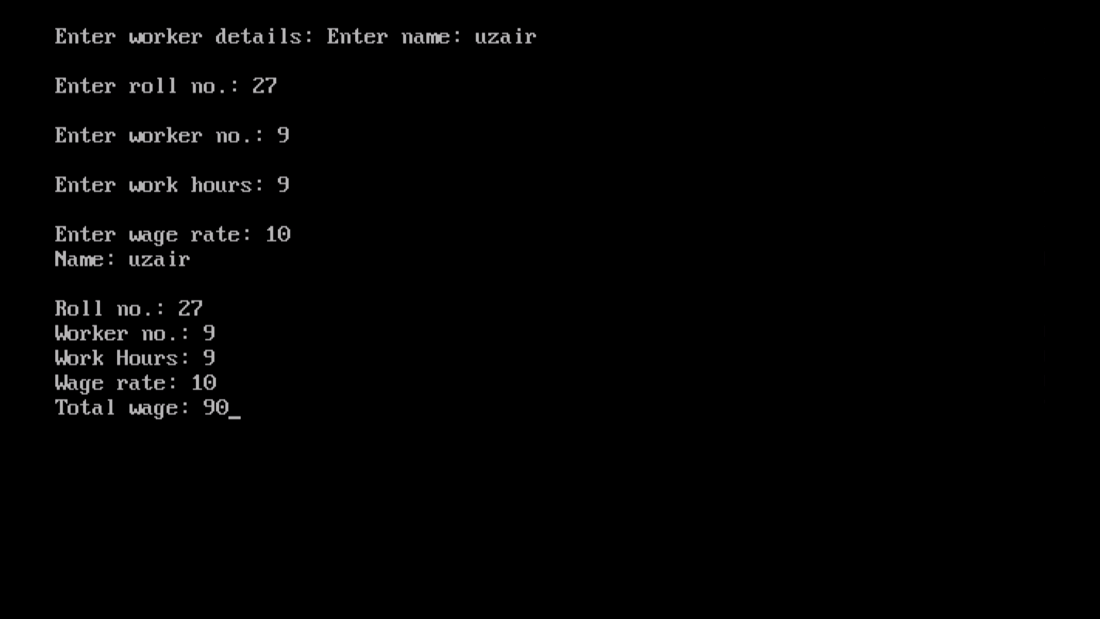
w1.In\_data();

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);

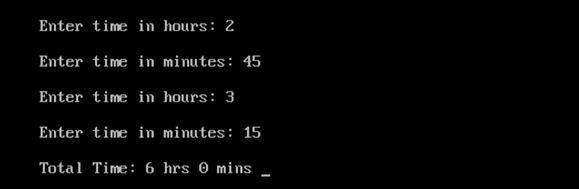
z.addtime(c1, c2);

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;

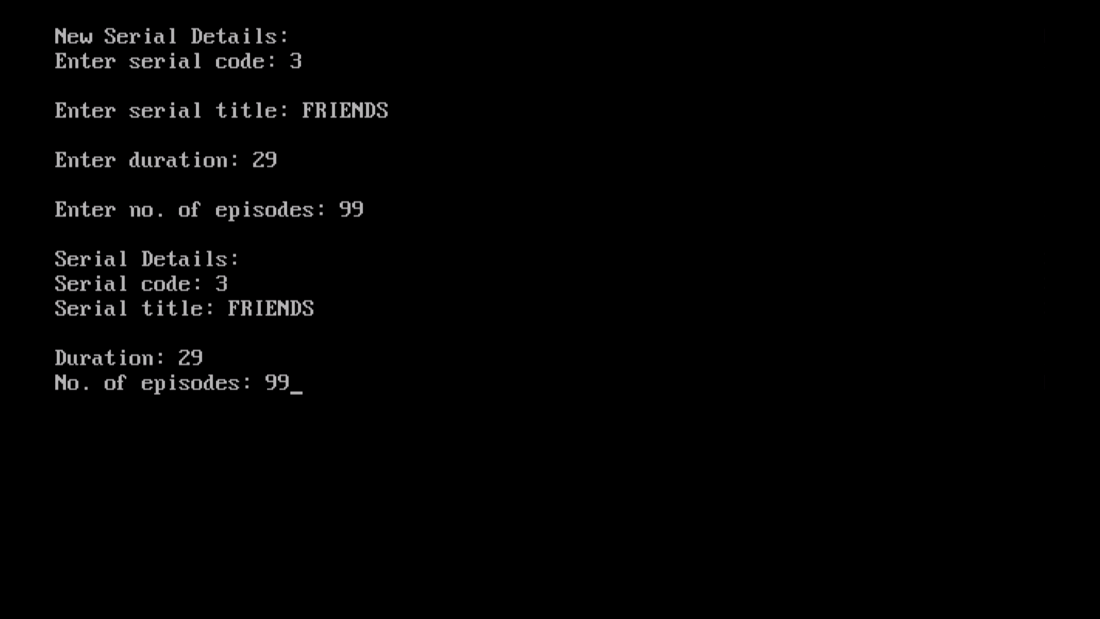
s1.Newserial();

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();

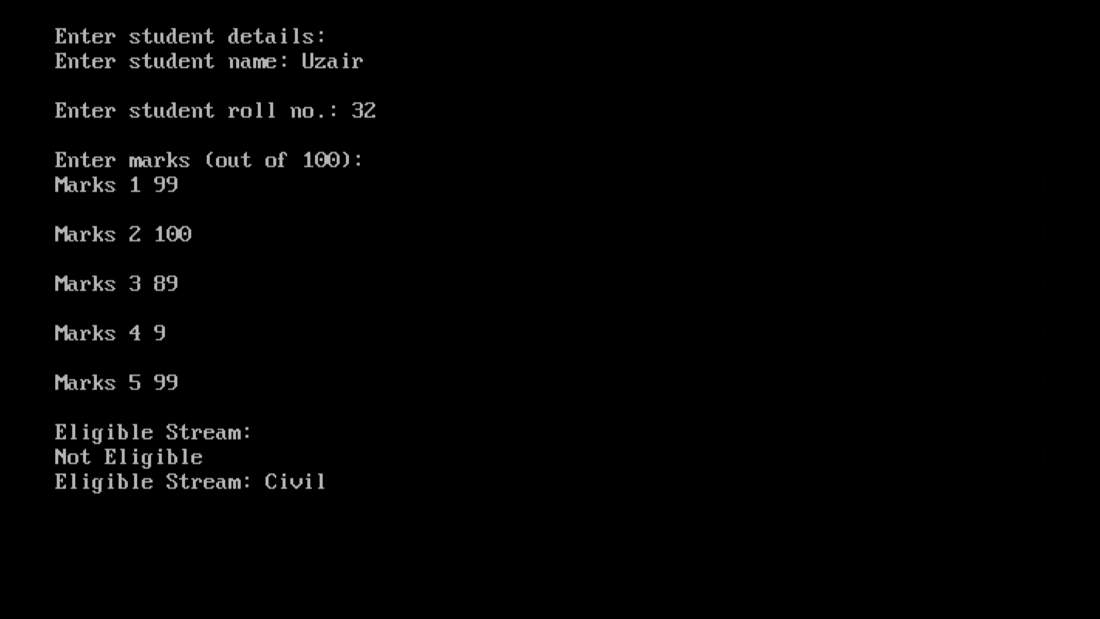
student s1;

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();

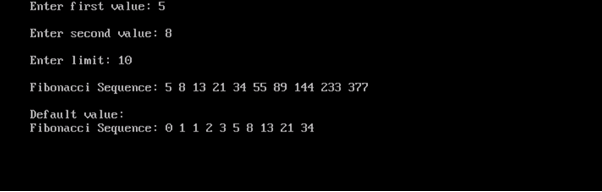
cout<<"\n"<<"\nDefault value: ";

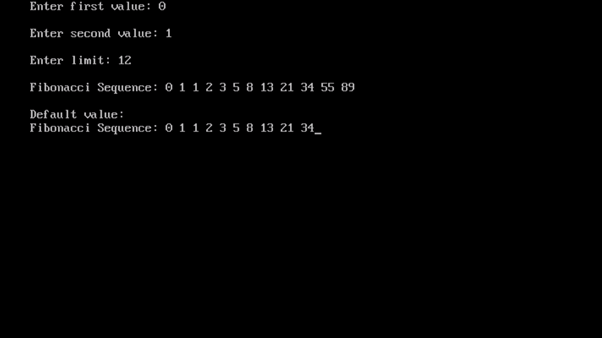
f1.gen\_fib();

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();

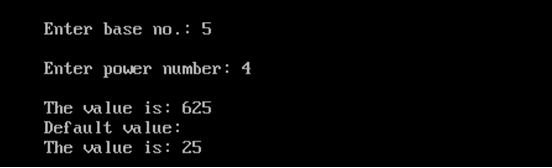
cout<<"\nDefault value: ";

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();

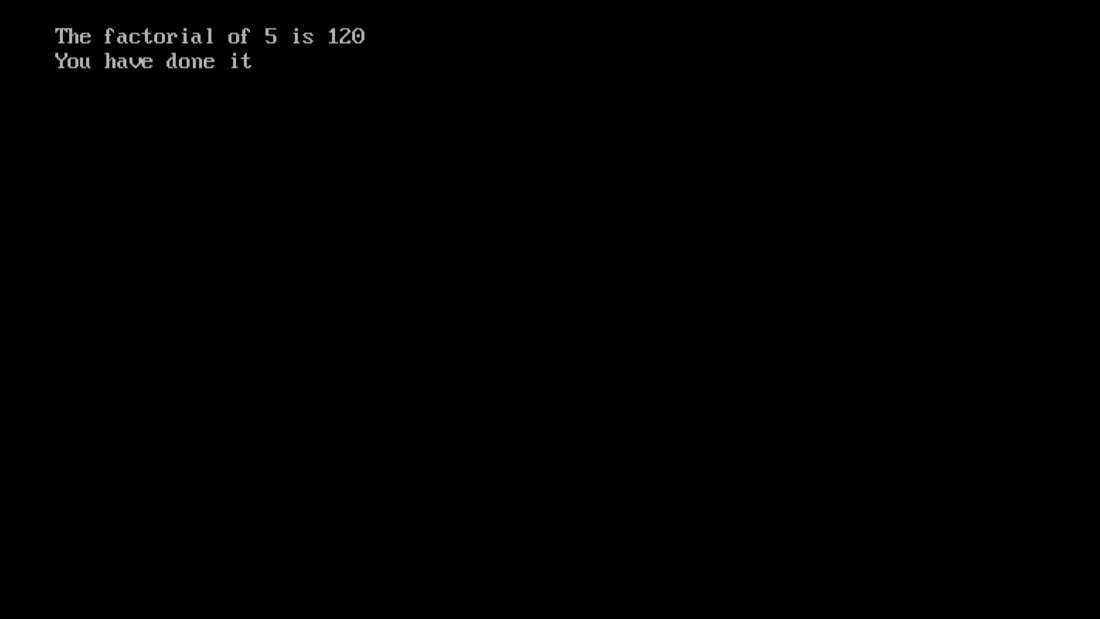
fact f1;

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;

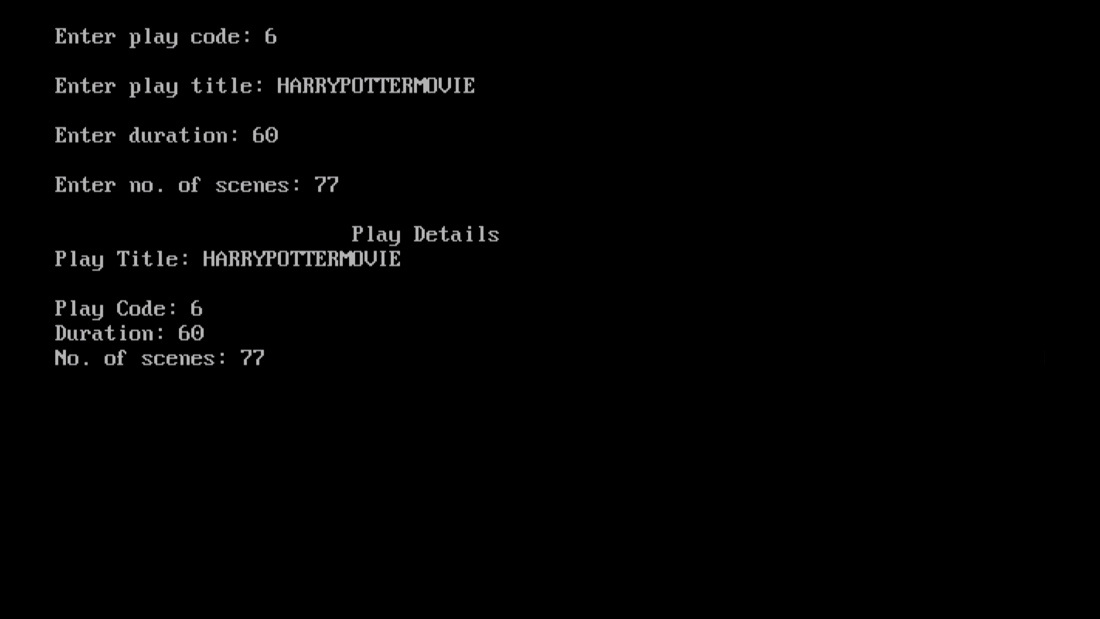
p1.Moreinfo(dur, no);

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;

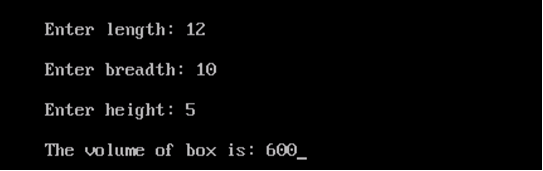
box b1(l1, b2, h1);

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();

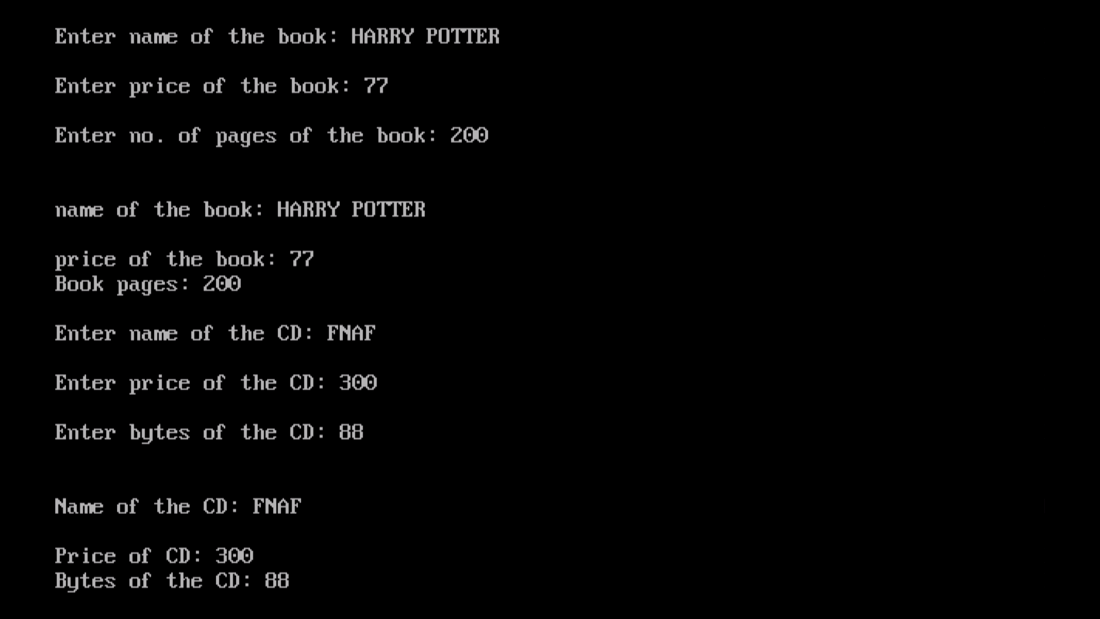
c1.getdata();

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();

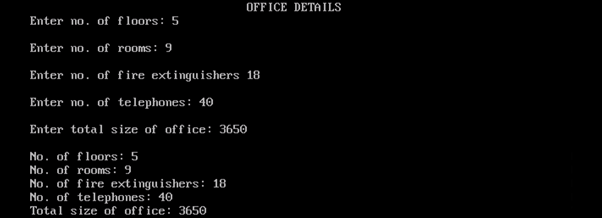
o1.getdata();

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();

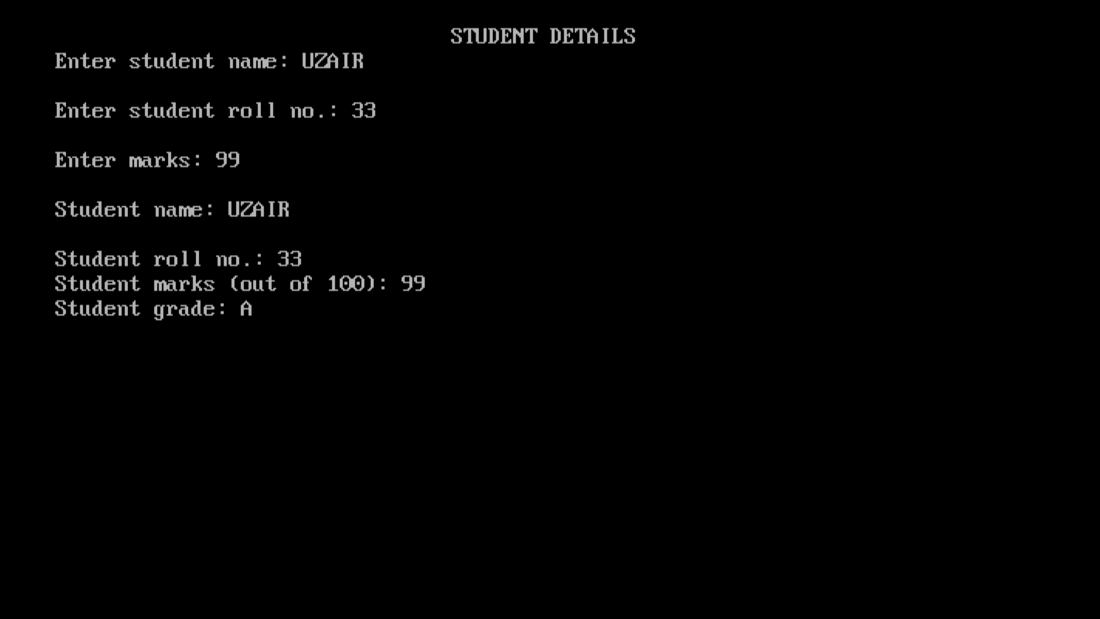
m1.gradecalc();

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;

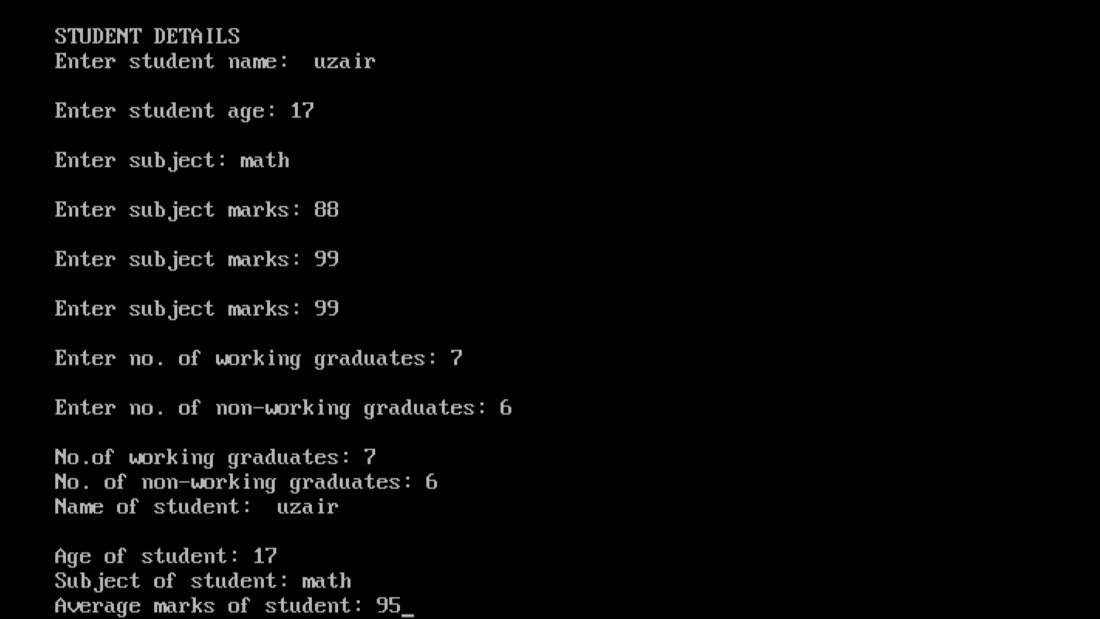
g1.getdata();

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;

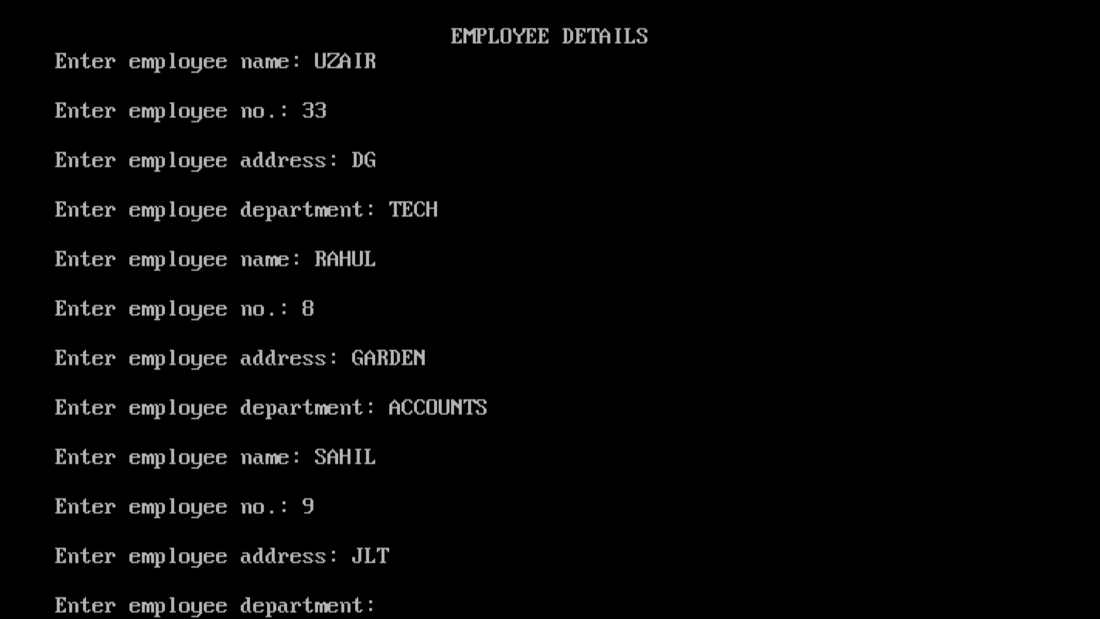
m1.getdata();

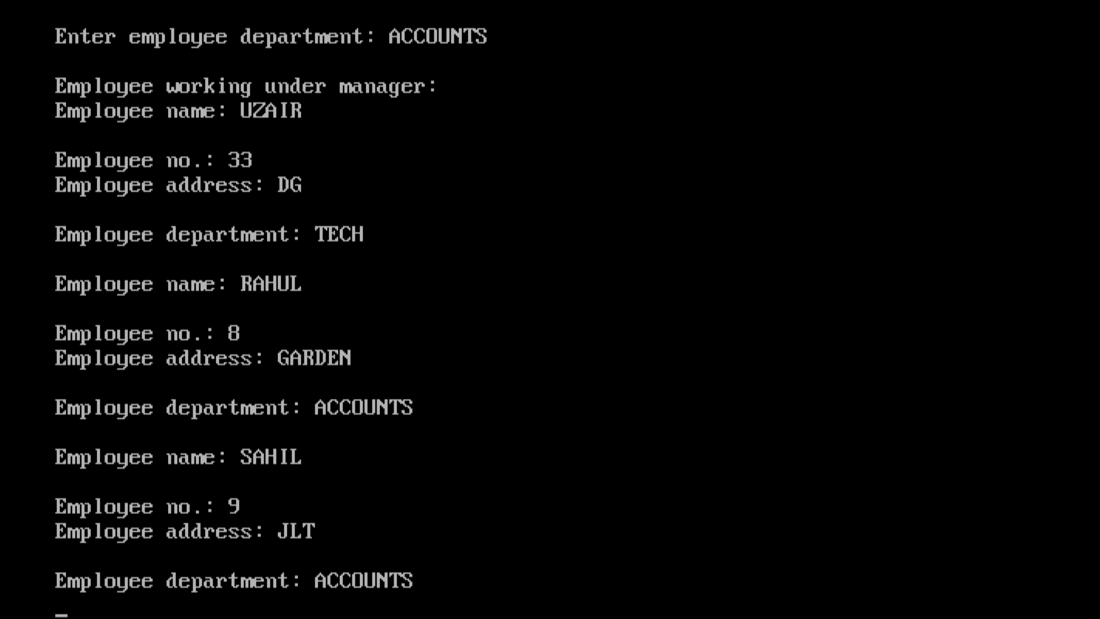
m1.putdata();

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();

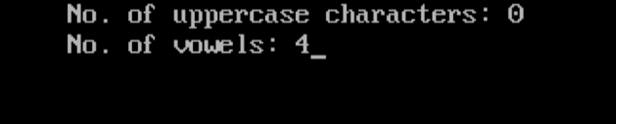
getch();

}

**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: ";

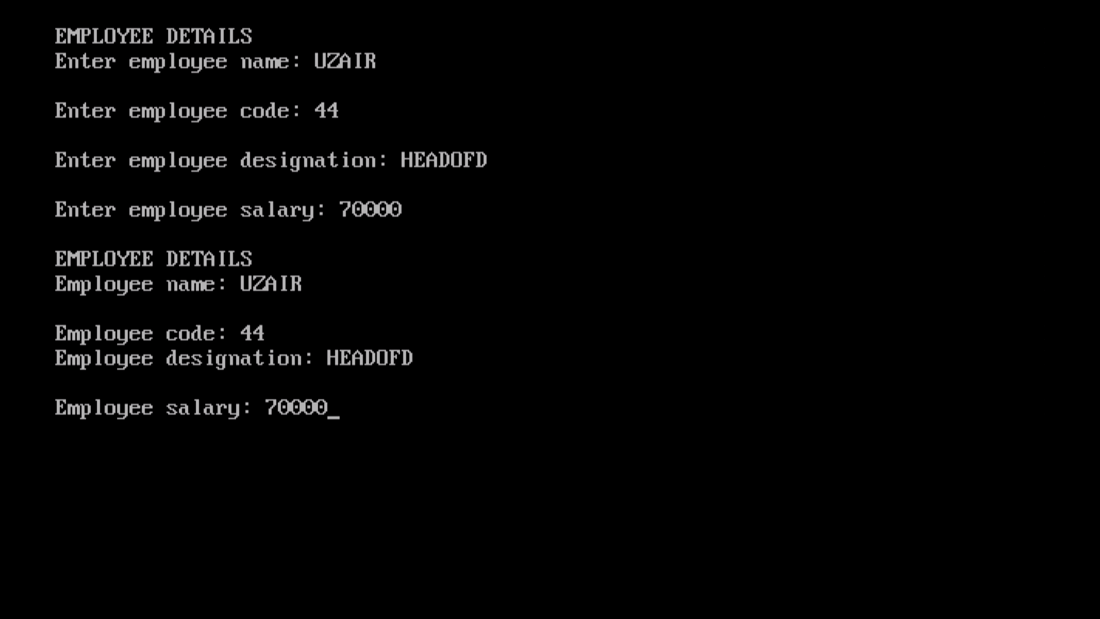
cout<<e.sal;

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();

}

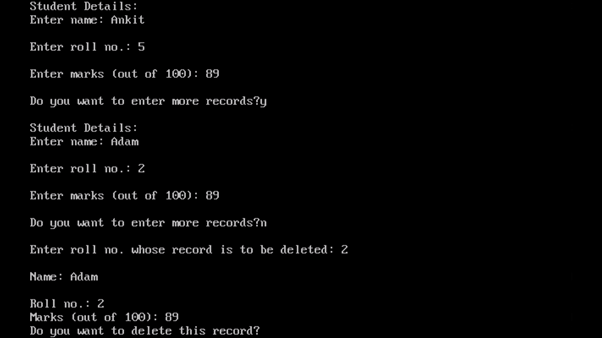
fin.close();

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')

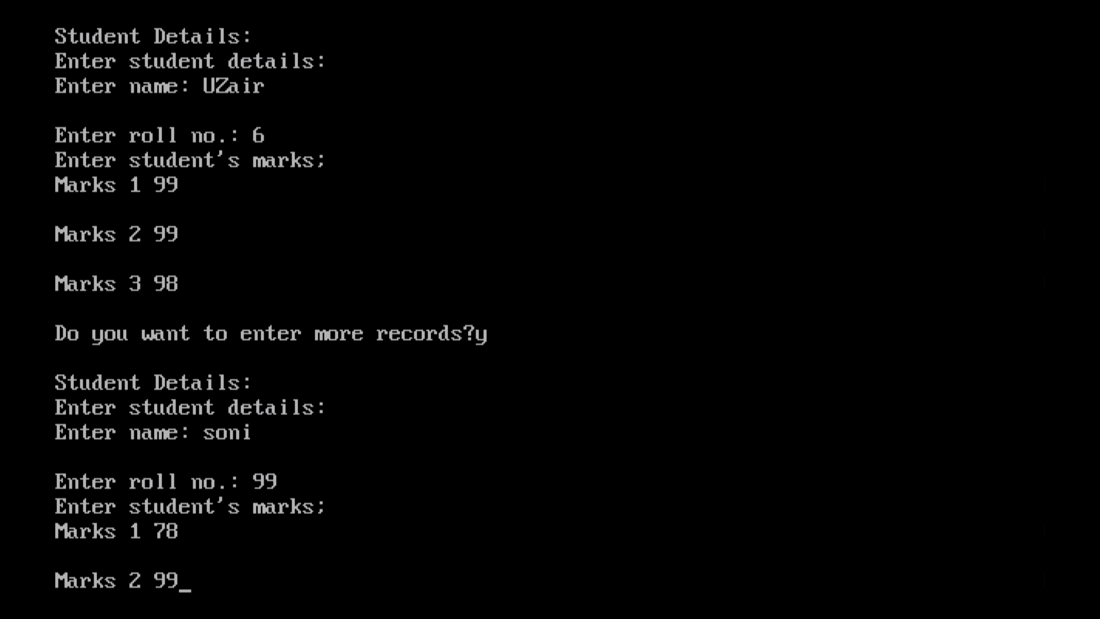
cout<<"\nRecord not found.";

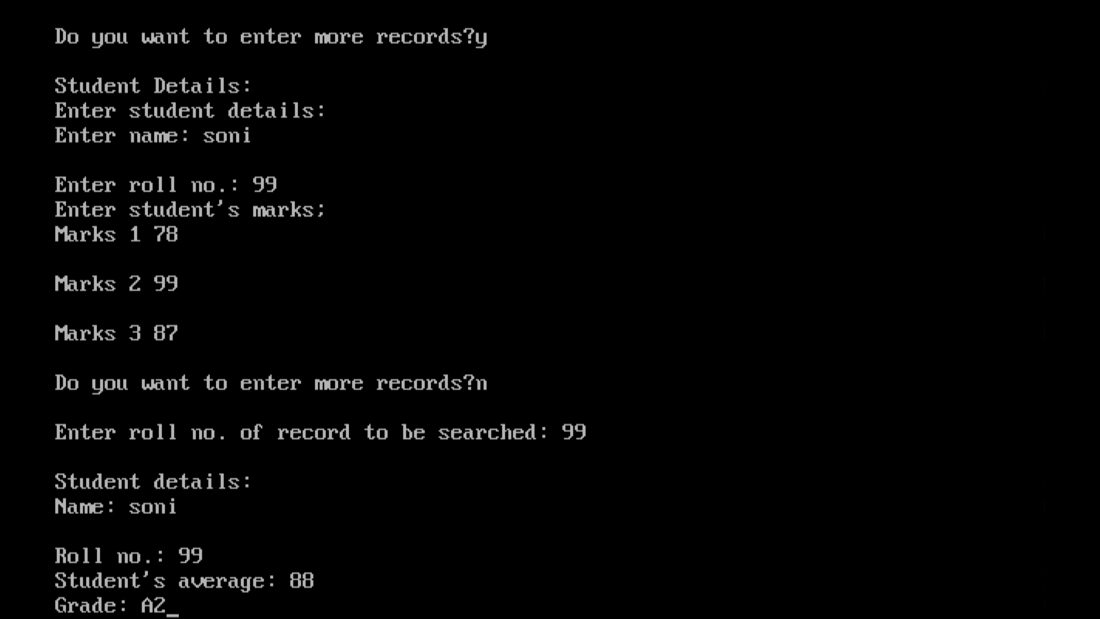
fin.close();

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();

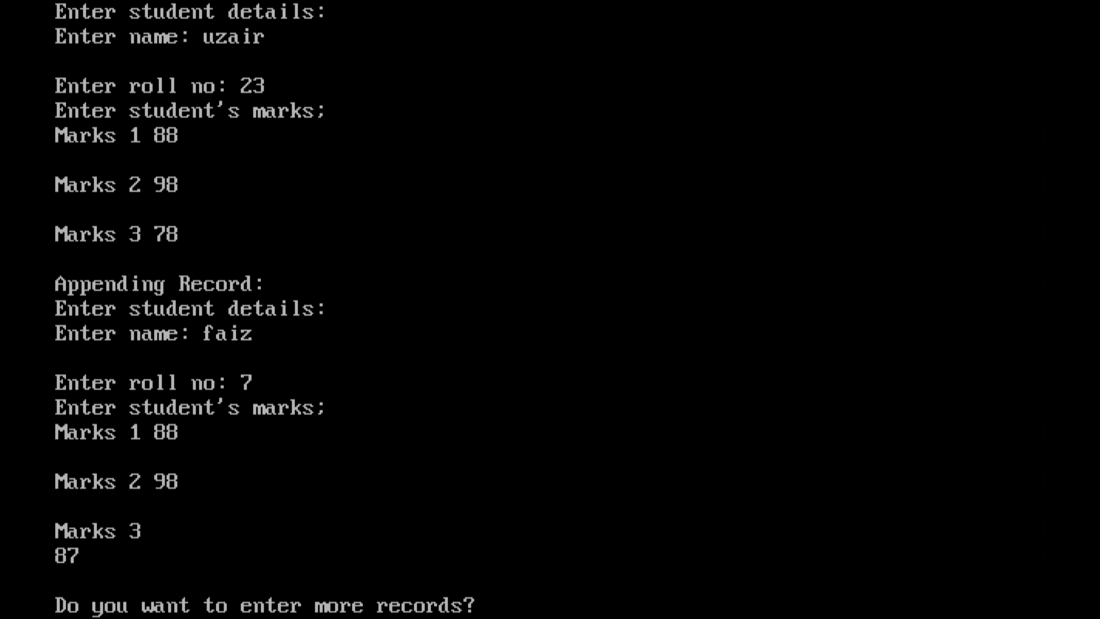
}

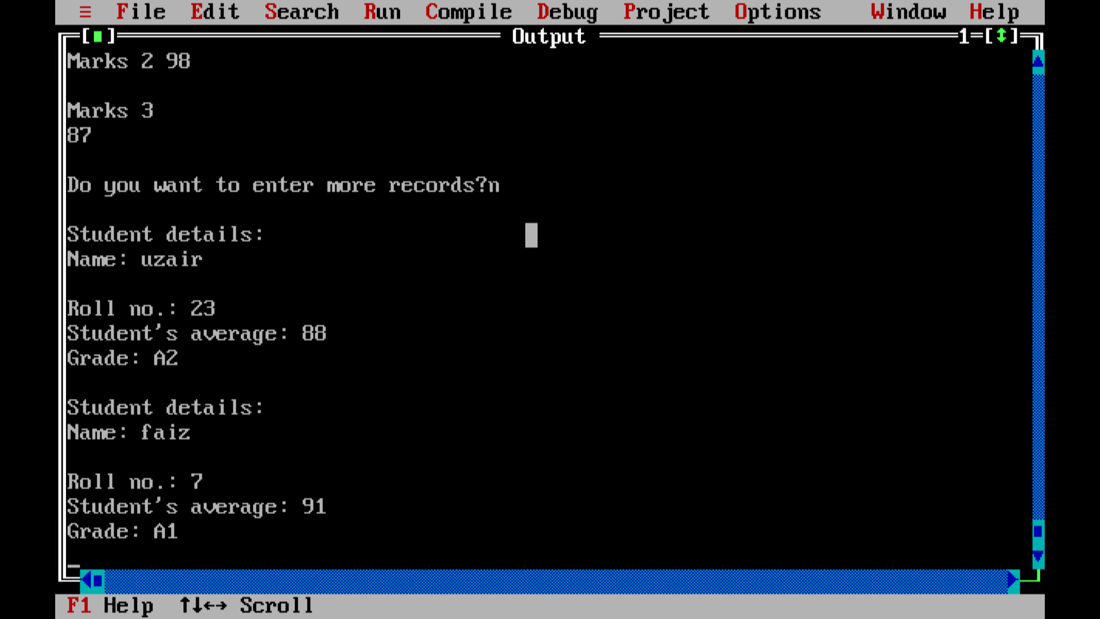
fout.close();

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:**

