|  |  |
| --- | --- |
| Computer project file  C++ and SQL | Siddharth Nandi  Class: XII/A Roll No.: 31 |

C++ PROGRAMS

**1. Program to calculate aggregate marks of a student using class and inheritance.**

#include<iostream.h>

#include<conio.h>

class applicant

{

private:

long admno;

char name[20], grade;

float agg;

void grademe();

public:

void enter();

void result();

};

void applicant::grademe()

{

if (agg>=80)

grade = 'A';

else if (agg>=65)

grade = 'B';

else if (agg>=50)

grade = 'C';

else if (agg>=33)

grade = 'D';

else

grade = 'E';

}

void applicant::enter()

{

cout<<"\nEnter name of student : ";

cin.getline(name,20);

cout<<"Enter Admission number : ";

cin>>admno;

cout<<"Enter aggregate marks : ";

cin>>agg;

grademe();

}

void applicant::result()

{

cout<<"\n\n================================================================================";

cout<<"\n\nName of student : "<<name;

cout<<"\nAdmission number : "<<admno;

cout<<"\nAggregate Marks : "<<agg;

cout<<"\nGrade : "<<grade;

}

void main()

{

applicant a1;

clrscr();

a1.enter();

a1.result();

getch();

}

**2. Program to calculate deposit and withdrawal amount from bank account using classes.**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<process.h>

int const min = 500;

float bal;

class Account

{

private:

char name[50];

long a\_no;

public:

void getdata()

{

cout<<endl;

cout<<"Enter name: ";

gets(name);

cout<<"Enter account number: ";

cin>>a\_no;

cout<<"Enter opening bal: ";

cin>>bal;

}

void display()

{

cout<<endl;

cout<<"Name: "<<name<<endl;

cout<<"Account number: "<<a\_no<<endl;

cout<<"Opening bal: "<<bal<<endl;

}

};

class Current : public Account

{

private:

float deposit, withdraw;

public:

void depo()

{

cout<<"Enter amount to be deposited: ";

cin>>deposit;

display();

bal = bal - deposit;

cout<<"Balance after deposit: "<<bal<<endl;

}

void with()

{

cout<<"Enter withdraw amount: ";

cin>>withdraw;

if (withdraw<bal)

{

display();

bal = bal - withdraw;

cout<<"Balance after withdraw: "<<bal<<endl;

}

else

{

cout<<"You cannot withdraw money."<<endl;

}

}

void check\_balance()

{

if (bal<min)

{

cout<<"Opening balance should be more than Rs. 500."<<endl;

bal = bal - 150;

cout<<endl<<"After penalty, main balance: Rs. "<<bal<<endl;

}

}

};

class Savings : public Account

{

private:

float deposit2, withdraw2, interest;

public:

void deposit()

{

cout<<endl<<"Enter amount to be deposited: ";

cin>>deposit2;

display();

bal = bal + deposit2;

cout<<"After deposit, balance: Rs. "<<bal<<endl;

}

void withdraw()

{

cout<<"Enter amount to be withdrawn: ";

cin>>withdraw2;

if (withdraw2<bal)

{

display();

bal = bal - withdraw2;

cout<<"After withdrawal, balance: Rs. "<<bal<<endl;

}

else

{

cout<<endl<<"You cannot withdraw money."<<endl;

}

}

void calculate\_interest()

{

interest = (bal\*2)/100;

bal = bal + interest;

cout<<endl<<"After calculating, interest balance is: "<<bal<<endl;

}

};

void main()

{

clrscr();

Current c;

Savings s;

char ch;

int choice, choice2;

cout<<"Enter 'S' for savings and 'C' for current: ";

cin>>ch;

if (ch == 'C' || ch == 'c')

{

c.getdata();

c.check\_balance();

l2:

cout<<"\n 1. Display \n 2. Deposit \n 3. Withdraw \n 4. Exit \n";

cout<<"Enter your choice: ";

cin>>choice;

switch (choice)

{

case 1: c.display();

goto l2;

break;

case 2: c.depo();

goto l2;

break;

case 3: c.with();

goto l2;

break;

case 4: exit (0);

}

}

else if (ch == 'S' || ch == 's')

{

s.getdata();

l1:

cout<<"\n 1. Display \n 2. Deposit \n 3. Withdraw \n 4. Calculate interest \n 5. Exit \n";

cout<<"Enter your choice: ";

cin>>choice2;

switch(choice2)

{

case 1: s.display();

goto l1;

break;

case 2: s.deposit();

goto l1;

break;

case 3: s.withdraw();

goto l1;

break;

case 4: s.calculate\_interest();

goto l1;

break;

case 5: exit (0);

}

}

else

{

cout<<"Wrong choice"<<endl;

}

getch();

}

**3. Program to take input of marks of student and store it to a text file.**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<stdio.h>

void main()

{

clrscr();

int maths, phy, chem, eng, cs;

char name[50];

fstream fout;

fout.open("Marks.txt", ios::out);

cout<<"\n";

cout<<"Enter student name : ";

gets(name);

cout<<"Marks in Maths : ";

cin>>maths;

cout<<"Marks in Physics : ";

cin>>phy;

cout<<"Marks in Chemistry : ";

cin>>chem;

cout<<"Marks in English : ";

cin>>eng;

cout<<"Marks in Computer Science : ";

cin>>cs;

fout<<"Student name : "<<name<<"\n\n";

fout<<"Marks in Maths : "<<maths<<"\n";

fout<<"Marks in Physics : "<<phy<<"\n";

fout<<"Marks in Chemistry : "<<chem<<"\n";

fout<<"Marks in English : "<<eng<<"\n";

fout<<"Marks in Computer Science: "<<cs<<"\n\n";

fout.close();

getch();

}

**4. Program to calculate total marks of a student in five subjects using classes.**

#include<iostream.h>

#include<conio.h>

void main()

{

class stdres

{

private:

char sname[20];

int maths, phy, chem, eng, cs, admno;

float tot;

public:

void resint()

{

cout<<"Enter Name : ";

cin.getline(sname,20);

cout<<"Enter admission number : ";

cin>>admno;

cout<<"Enter marks in maths : ";

cin>>maths;

cout<<"Enter marks in physics : ";

cin>>phy;

cout<<"Enter marks in chemistry : ";

cin>>chem;

cout<<"Enter marks in english : ";

cin>>eng;

cout<<"Enter marks in C.S. : ";

cin>>cs;

cout<<"\n---------------------------------------------------------------";

}

int total()

{

return maths + phy + chem + eng + cs;

}

void displayint()

{

cout<<"\nStudent name : "<<sname;

cout<<"\nAdmission number : "<<admno;

cout<<"\nMarks in Maths : "<<maths;

cout<<"\nMarks in Physics : "<<phy;

cout<<"\nMarks in Chemistry : "<<chem;

cout<<"\nMarks in English : "<<eng;

cout<<"\nMarks in Computer Science: "<<cs;

cout<<"\nTotal marks : "<<total();

}

};

stdres s1;

clrscr();

s1.resint();

s1.displayint();

getch();

}

**5. Program to calculate the total air fair depending on the number of passengers using classes.**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

class tour

{

private:

char tcode[20];

int noofkids, noofadults, kilometres;

float tot\_fare;

public:

tour()

{

strcpy(tcode, "NULL");

noofadults = 0;

noofkids = 0;

kilometres = 0;

tot\_fare = 0;

}

void assignfare()

{

if (kilometres >= 1000)

tot\_fare = noofadults \* 500 + noofkids \* 250;

else if ((kilometres < 1000) && (kilometres >= 500))

tot\_fare = noofadults \* 300 + noofkids \* 150;

else

tot\_fare = noofadults \* 200 + noofkids \* 100;

}

void enter\_tour()

{

cout<<"Enter value for travel code: ";

cin.getline(tcode, 20);

cout<<"Enter no. of adults : ";

cin>>noofadults;

cout<<"Enter no. kids : ";

cin>>noofkids;

cout<<"Enter kilometres : ";

cin>>kilometres;

assignfare();

}

void show\_tour()

{

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PASSENGER INFORMATION\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<endl;

cout<<"\t\t\tTravel code : "<<tcode<<endl;

cout<<"\t\t\tNumber of adults : "<<noofadults<<endl;

cout<<"\t\t\tNumber of children : "<<noofkids<<endl;

cout<<"\t\t\tDistance : "<<kilometres<<endl;

cout<<"\t\t\tTotal fare : "<<tot\_fare<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

}

};

void main()

{

clrscr();

tour t1;

t1.enter\_tour();

clrscr();

t1.show\_tour();

getch();

}

**6. Program to calculate wage of a worker using classes.**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

class worker

{

private:

char wname[30];

float hrwrk, wgrate, totwage;

float calcw()

{

totwage = hrwrk \* wgrate;

return totwage;

}

public:

void in\_data()

{

cout<<"Enter name : ";

cin.getline(wname,30);

cout<<"Enter work hours: ";

cin>>hrwrk;

cout<<"Enter wage rate : ";

cin>>wgrate;

}

void out\_data()

{

cout<<"Name : "<<wname;

cout<<"\nWork Hours : "<<hrwrk<<" hours";

cout<<"\nWage Rate : Rs. "<<wgrate<<" per hour";

cout<<"\nTotal wage : Rs. "<<calcw();

}

};

void main()

{

clrscr();

worker w1;

w1.in\_data();

clrscr();

w1.out\_data();

getch();

}

**7. Insertion in an array.**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

class worker

{

private:

char wname[30];

float hrwrk, wgrate, totwage;

float calcw()

{

totwage = hrwrk \* wgrate;

return totwage;

}

public:

void in\_data()

{

cout<<"Enter name : ";

cin.getline(wname,30);

cout<<"Enter work hours: ";

cin>>hrwrk;

cout<<"Enter wage rate : ";

cin>>wgrate;

}

void out\_data()

{

cout<<"Name : "<<wname;

cout<<"\nWork Hours : "<<hrwrk<<" hours";

cout<<"\nWage Rate : Rs. "<<wgrate<<" per hour";

cout<<"\nTotal wage : Rs. "<<calcw();

}

};

void main()

{

clrscr();

worker w1;

w1.in\_data();

clrscr();

w1.out\_data();

getch();

}

**8. Binary Search in an array.**

#include<iostream.h>

#include<conio.h>

int Bsearch(int[], int, int);

void main()

{

clrscr();

int AR[50], ITEM, N, index;

cout<<"Enter desired array: ";

cin>>N;

cout<<"Enter array elements (in asc order): \n";

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

cin>>AR[i];

cout<<"Enter element to be searched for: ";

cin>>ITEM;

index = Bsearch(AR, N, ITEM);

if (index == -1)

cout<<"\nSorry!! Given element could not be found\n";

else

cout<<"Element found at index: "<<index<<", Position: "<<index+1<<endl;

getch();

}

int Bsearch(int AR[], int size, int item)

{

int beg, last, mid;

beg = 0;

last = size-1;

while (beg <= last)

{

mid = (beg + last)/2;

if (item == AR[mid])

return mid;

else if (item > AR[mid])

beg = mid + 1;

else last = mid -1;

}

return -1;

}

**9. Selection sort in an array.**

#include<iostream.h>

#include<conio.h>

void sort(int [], int);

void main()

{

int AR[10], N, ITEM;

clrscr();

cout<<"Enter number of elements: ";

cin>>N;

cout<<"Enter elements of array: \n";

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

cin>>AR[i];

sort(AR, N);

cout<<"\n\nThe sorted array is: \n";

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

cout<<AR[i]<<" ";

cout<<endl;

getch();

}

void sort(int AR[], int N)

{

int small, pos, tmp;

for(int i = 0; i < N-1; i++)

{

small = AR[i];

pos = i;

for (int j = i+1; j < N; j++)

{

if (AR[j] < small)

{

small = AR[j];

pos = j;

}

}

tmp = AR[i];

AR[i] = AR[pos];

AR[pos] = tmp;

cout<<"\nArray after passing "<<i+1<<" is: ";

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

{

cout<<AR[j]<<" ";

}

}

getch();

}

**10. Program to add, display, modify and delete data using data file handling and classes.**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<stdio.h>

class student

{

private:

char name[50];

int rno;

public:

void indata()

{

cout<<"\nEnter Name : ";

gets(name);

cout<<"\nEnter Roll No.: ";

cin>>rno;

}

void outdata()

{

cout<<"\nName : "<<name;

cout<<"\nRoll No.: "<<rno;

}

int rollno()

{

return rno;

}

void modify()

{

char name2[50];

int rno2;

cout<<"\nEnter name : ";

gets(name2);

cout<<"\nEnter Roll No.: ";

cin>>rno2;

}

};

void main()

{

char ans;

int choice, pos, rn, found=0, deleted=0, n=0;

student s1, s2;

ofstream file1;

fstream file2, tempfile;

while(choice!=6)

{

clrscr();

cout<<"1. Add\n";

cout<<"2. Display\n";

cout<<"3. Search\n";

cout<<"4. Modify\n";

cout<<"5. Delete\n";

cout<<"6. Exit\n";

cout<<"Enter Choice: ";

cin>>choice;

switch(choice)

{

case 1: clrscr();

file1.open("stud", ios::app | ios::binary);

s1.indata();

file1.write((char \*)&s1, sizeof(s1));

file1.close();

cout<<"\n\nRecord Successfully added";

cout<<"\n\nPress any key to continue";

getch();

break;

case 2: clrscr();

file2.open("stud", ios::in | ios::binary);

while (file2)

{

file2.read((char \*)&s1, sizeof (s1));

if(!file2.eof())

s1.outdata();

cout<<"\n-----------------------------\n";

n++;

if (n>3)

{

cout<<"\nPress any key to continue!!";

getch();

n=0;

}

}

file2.close();

cout<<"\nPress any key to continue....";

getch();

break;

case 3: clrscr();

file2.open("stud", ios::in |ios::binary);

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

cin>>rn;

while(!file2.eof())

{

file2.read((char\*)&s2, sizeof(s2));

if(s2.rollno()==rn)

{

s2.outdata();

found=1;

}

}

if(found==0)

cout<<"\nRecord not found!";

file2.close();

found=0;

cout<<"\nPress any key to continue";

getch();

break;

case 4: clrscr();

file2.open("stud" , ios::in | ios::out | ios::binary);

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

cin>>rn;

while(!file2.eof())

{

pos = file2.tellg();

file2.read((char\*)&s2, sizeof (s2));

if (s2.rollno() == rn)

{

s2.outdata();

cout<<"\nDo you want to edit this record(Y/N): ";

cin>>ans;

if (ans == 'y' || ans == 'Y')

{

cout<<"\nEnter details to modify";

s2.modify();

file2.seekg(pos);

file2.write((char \*)&s2, sizeof(s2));

cout<<"\nRecord successfully added";

}

found = 1;

break;

}

}

if (found == 0)

cout<<"\nRoll no. not found";

file2.close();

found = 0;

cout<<"\nPress any key to continue...";

getch();

break;

case 5: clrscr();

file2.open("stud", ios::in | ios::binary);

tempfile.open("temp", ios::out | ios::binary);

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

cin>>rn;

while(!file2.eof())

{

file2.read((char \*)&s2, sizeof(s2));

if (!file2.eof())

{

if (s2.rollno() == rn)

{

s2.outdata();

cout<<"\nDo you want to delete this record?(Y/N): ";

cin>>ans;

if (ans == 'n' || ans == 'N')

{

tempfile.write((char \*)&s2, sizeof (s2));

deleted=1;

}

found = 1;

}

else

tempfile.write((char \*)&s2, sizeof (s2));

}

}

if (found == 0)

cout<<"\nRecord not found!";

else if (deleted == 0)

cout<<"\nRecord deleted successfully";

file2.close();

tempfile.close();

remove("stud");

rename("temp" , "stud");

found = 0;

deleted = 0;

cout<<"\nPress any key to continue...";

getch();

break;

case 6: clrscr();

cout<<"\n\n\n\t\tTHANK YOU";

getch();

break;

default: clrscr();

cout<<"\nInvalid choice";

getch();

}

}

getch();

}

**11. Program to add flight details using data file handling and classes.**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<stdio.h>

#include<string.h>

class flight

{

int fl\_no;

char destination[30];

public:

void getdata()

{

cout<<"\nEnter Flight Number: ";

cin>>fl\_no;

cout<<"\nEnter Destination of flight: ";

gets(destination);

}

void putdata()

{

cout<<"\nFlight Number: "<<fl\_no;

cout<<"\nDestination : "<<destination;

}

int code()

{ return fl\_no;}

}f1,f2;

void main()

{

clrscr();

ofstream ok("flight.txt", ios::app|ios::binary);

char want='y';

while(want=='y')

{

f1.getdata();

ok.write((char\*)&f1, sizeof(f1));

cout<<"\nRecord added\n";

cout<<"\nWant to record more data?(y/n)";

cin>>want;

}

ok.close();

clrscr();

int x;

cout<<"\n----------------Menu-------------------\n";

cout<<"Choose your option: ";

cout<<"\n 1. Insert data";

cout<<"\n 2. Display data";

cout<<"\n 3. Modify data";

cout<<"\n 4. Delete data";

cout<<"\n 5. Search data";

cin>>x;

if(x=='1')

{

ifstream ko("flight.txt", ios::in | ios::binary);

char last='l';

clrscr();

cout<<"\nEnter details of flight to be inserted\n";

f1.getdata();

while(!ko.eof())

{

ok.write((char\*)&f1, sizeof(f1));

if(f1.code()<=f2.code())

{

ok.write((char\*)&f1, sizeof(f1));

last='m';

break;

}

else

ok.write((char\*)&f2, sizeof(f2));

}

if(last=='l')

ok.write((char\*)&f2, sizeof(f1));

else if(!ko.eof())

{

while(!ko.eof())

{

ko.read((char\*)&f2,sizeof(f2));

ok.write((char\*)&f2,sizeof(f2));

}

}

ok.close();

ko.close();

}

if(x=='2')

{

ifstream ko("flight.txt", ios::in | ios::binary);

ko.open("flight.txt", ios::in);

cout<<"\nFile now contains\n";

while(!ko.eof())

{

ko.read((char\*)&f2, sizeof(f2));

if(ko.eof())

break;

f2.putdata();

}

ko.close();

}

if(x=='3')

{

fstream ko("flight.txt", ios::in | ios::out | ios::binary);

int fno;

}

getch();

}

**12. Program to convert number to words.**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

void main()

{

clrscr();

long int z, c, o, o1, t, t1, t2, t3, h, h1, h2, h3, h4;

long int T, T1, T2, T3, T4, T5, T6, tT, tT1, tT2, tT3, tT4, tT5, tT6, tT7, tT8;

long int L, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10;

long int x[10];

char \*zero[10] ={"Zero"};

char \*teen[10] ={"","Eleven","Twelve","Thirteen","Fourteen","Fifteen","Sixteen","Seventeen","Eighteen","Nineteen"};

char \*ones[10] ={"","One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

char \*tens[10] ={"","Ten", "Twenty", "Thirty", "Fourty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

char \*hund[2] ={"Hundred"};

char \*thou[2] ={"Thousand"};

char \*lakh[2] ={"Lakh"};

cout<<"Enter any number between 0 to 999999: ";

cin>>z;

clrscr();

if ((z<10) && (z!=0))

{

o1 = z%10;

cout<<"\nValue in words: "<<\*(ones+o1);

}

else if (z==0)

{

cout<<"\nValue in words: "<<\*(zero);

}

if (z==10)

{

cout<<"\nValue in words: "<<\*(tens+1);

}

else if ((z>10) && (z<20))

{

t3 = z%10;

cout<<"\nValue in words: "<<\*(teen+t3);

}

else if ((z<100) && (z>=20))

{

t1 = z%10;

t2 = z/10;

cout<<"\nValue in words: "<<\*(tens+t2)<<" "<<\*(ones+t1);

}

if ((z>=100) && (z<1000))

{

h1 = z%100;

h2 = z/100;

h3 = h1/10;

h4 = h1%10;

if((h1>10) && (h1<20))

{

cout<<"\nValue in words: "<<\*(ones+h2)<<" "<<\*(hund)<<" "<<\*(teen+h3);

}

else

{

cout<<"\nValue in words: "<<\*(ones+h2)<<" "<<\*(hund)<<" "<<\*(tens+h3)<<" "<<\*(ones+h4);

}

}

if((z>=1000) && (z<10000))

{

T1=z/1000;

T2=z%1000;

T3=T2/100;

T4=T2%100;

T5=T4/10;

T6=T4%10;

if(T3==0)

{

if ((T4>10) && (T4<20))

{

cout<<"\nValue in words: "<<\*(ones+T1)<<" "<<\*(thou)<<" "<<\*(teen+T5);

}

else

{

cout<<"\nValue in words: "<<\*(ones+T1)<<" "<<\*(thou)<<" "<<\*(tens+T5)<<" "<<\*(ones+T6);

}

}

else if(T3!=0)

{

if ((T4>10) && (T4<20))

{

cout<<"\nValue in words: "<<\*(ones+T1)<<" "<<\*(thou)<<" "<<\*(ones+T3)<<" "<<\*(hund)<<" "<<\*(teen+T5);

}

else

{

cout<<"\nValue in words: "<<\*(ones+T1)<<" "<<\*(thou)<<" "<<\*(ones+T3)<<" "<<\*(hund)<<" "<<\*(tens+T5)<<" "<<\*(ones+T6);

}

}

}

if((z>=10000) && (z<100000))

{

tT1=z/10000;

tT2=z%10000;

tT3=tT2/1000;

tT4=tT2%1000;

tT5=tT4/100;

tT6=tT4%100;

tT7=tT6/10;

tT8=tT6%10;

if (tT3==0)

{

if (tT5==0)

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(thou)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(thou)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

else

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

}

else

{

if(tT1==1)

{

if (tT5==0)

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(teen+tT3)<<" "<<\*(thou)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(teen+tT3)<<" "<<\*(thou)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

else

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(teen+tT3)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(teen+tT3)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

}

else

{

if (tT5==0)

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(ones+tT3)<<" "<<\*(thou)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(ones+tT3)<<" "<<\*(thou)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

else

{

if ((tT6>10) && (tT6<20))

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(ones+tT3)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(teen+tT7);

}

else

{

cout<<"\nValue in words: "<<\*(tens+tT1)<<" "<<\*(ones+tT3)<<" "<<\*(thou)<<" "<<\*(ones+tT5)<<" "<<\*(hund)<<" "<<\*(tens+tT7)<<" "<<\*(ones+tT8);

}

}

}

}

}

if((z>=100000) && (z<1000000))

{

L1 = z/100000;

L2 = z%100000;

L3 = L2/10000;

L4 = L2%10000;

L5 = L4/1000;

L6 = L4%1000;

L7 = L6/100;

L8 = L6%100;

L9 = L8/10;

L10= L8%10;

if (L3==0)

{

if (L5==0)

{

if (L7==0)

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

else

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

}

else

{

if (L7==0)

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

else

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

}

}

else

{

if (L5==0)

{

if (L7==0)

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(thou)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(thou)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

else

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

}

else

{

if (L3==1)

{

if (L7==0)

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(teen+L5)<<" "<<\*(thou)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(teen+L5)<<" "<<\*(thou)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

else

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(teen+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(teen+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

}

else

{

if (L7==0)

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

else

{

if ((L8>10) && (L8<20))

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(teen+L9);

}

else

{

cout<<"\nValue in words: "<<\*(ones+L1)<<" "<<\*(lakh)<<" "<<\*(tens+L3)<<" "<<\*(ones+L5)<<" "<<\*(thou)<<" "<<\*(ones+L7)<<" "<<\*(hund)<<" "<<\*(tens+L9)<<" "<<\*(ones+L10);

}

}

}

}

}

}

getch();

}

**13. Program to add objects to the beginning of a linked list.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct node

{

int info;

node \*next;

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

node \*create\_new\_node( int );

void insert\_beg ( node\* );

void display (node\*);

void main()

{

clrscr();

start = NULL;

int inf;

char ch='y';

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

{

system("cls");

cout<<"Enter information for the new node: ";

cin>>inf;

cout<<"Creating new node...\n";

system("pause");

newptr = create\_new\_node(inf);

if (newptr != NULL)

{

cout<<"\n\nNew node created successfully...\n";

system("pause");

}

else

{

cout<<"\n\nCouldn't create new node! Aborting.\n" ;

system("pause");

exit(1);

}

cout<<"\n\nNow inserting this node in the beginning of list...\n";

system("pause");

insert\_beg(newptr);

cout<<"\nNow the list is: \n";

display(start);

cout<<"\nPress Y to enter more nodes, N to exit...\n";

cin>>ch;

}

getch();

}

node \*create\_new\_node(int n)

{

ptr = new node;

ptr -> info = n;

ptr -> next = NULL;

return ptr;

}

void insert\_beg(node\* np)

{

if (start == NULL)

start = np;

else

{

save = start;

start = np;

np -> next = save;

}

}

void display( node\* np)

{

while (np != NULL)

{

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

np= np -> next;

}

}

**14. Linear search in an array.**

#include<iostream.h>

#include<conio.h>

int LSearch(int[], int, int);

void main()

{

int ar[10], size, index, item;

cout<<"Enter size of array(Max 10): ";

cin>>size;

cout<<"Enter elements: ";

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

{

cin>>ar[i];

}

cout<<"Enter element to search for: ";

cin>>item;

index=LSearch(ar, size, item);

if (index == -1)

{

cout<<"The Element doesn't exist in the array";

getch();

}

else

{

cout<<"Element found at index: "<<index<<" and position: "<<index+1;

}

getch();

}

int LSearch(int ar[], int size, int item)

{

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

{

if (ar[i] == item)

{

return i;

}

return -1;

}

}

**15. Deletion in an array.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

int Lsearch(int [], int, int);

void main()

{

clrscr();

int ar[10], item, size, index;

cout<<"Enter number of elements for array(max 10): ";

cin>>size;

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

{

cin>>ar[i};

}

char ch='y';

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

{

cout<<"Enter element to be deleted: ";

cin>>item;

if (size==0)

{

cout<<"UNDERFLOW!"

getch();

}

index = Lsearch(ar, size, item);

if (index != -1)

{

ar[index]=0;

}

else

{

cout<<"Sorry! No such element found.";

getch();

}

cout<<"\nNow the array is: ";

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

{

cout<<ar[i];

}

cout<<"\n";

for(i=index; i<size; i++)

{

ar[i] = ar[i+1];

}

size -=1;

cout<<"Want to delete more elements? (y/n): ";

cin>>ch;

}

cout<<"\nThe final array is: ";

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

{

cout<<ar[i]<<" ";

}

cout<<endl;

getch();

}

int Lsearch(int ar[], int size, int item)

{

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

if (ar[i] == item)

{

return i;

}

return -1;

}

**16. Bubble Sort in array.**

#include<iostream.h>

#include<conio.h>

void BubbleSort(int[], int);

void main()

{

int ar[10], item, size;

cout<<"Enter size of array(max 10): ";

cin>>size;

cout<<"Enter elements for array: ";

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

{

cin>>ar[i];

}

BubbleSort(ar, size);

cout<<"The sorted array is: ";

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

{

cout<<ar[i]<<" ";

}

cout<<endl;

getch();

}

BubbleSort(int ar[], int size)

{

int tmp, ctr=0;

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

{

for (int j=0; j<(size-1)-i; j++)

{

if (ar[j]>ar[j+1])

{

tmp=ar[j];

ar[j]=ar[j+1];

ar[j+1]=tmp;

}

}

cout<<"Array after iteration - "<<++ctr<<" - is: ";

for(int k=0; K<size; k++)

{

cout<<ar[k]<<" ";

}

}

}

**17. Insertion Sort in array.**

#include<iostream.h>

#include<conio.h>

#include<limits.h>

void InsSort(int [], int);

void main()

{

int ar[50], size;

cout<<"Enter number of elements for array(max 10): ";

cin>>size;

cout<<"Enter elements for array: ";

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

{

cin>>ar[i];

}

InsSort(ar, size);

cout<<"The sorted array is: ";

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

{

cout<<ar[i]<<" ";

}

cout<<endl;

getch();

}

void InsSort(int ar[], int size)

{

int tmp, j;

ar[0]=INT\_MIN;

for(int i=1; i<=size; i++)

{

tmp=ar[i];

j=i-1;

while(tmp<ar[j])

{

ar[j+1] = ar[j];

j--;

}

ar[j+1] = tmp;

cout<<"Array after pass - "<<i<<" - is: ";

for(int k=1; k<=size; k++)

cout<<ar[k]<<" ";

cout<<endl;

}

}

**18. Merging two array in which one array is in ascending order and the other is in descending order. Resultant array is in ascending order.**

#include<iostream.h>

#include<conio.h>

void merge(int[], int, int[], int, int[]);

void main()

{

int a[10], b[10], c[20], mn=0, m, n;

cout<<"Enter number of elements for first array(max 10): ";

cin>>m;

cout<<"Enter elements for first array(ascending): \n";

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

{

cin>>a[i];

}

cout<<"Enter number of elements for second array(max 10): ";

cin>>n;

mn= m + n;

cout<<"Enter elements for second array(descending): \n";

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

{

cin>>b[i];

}

merge(a, m, b, n, c);

cout<<"The merged array is: ";

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

{

cout<<c[i]<<" ";

}

getch();

}

void merge(int A[], int m, int B[], int n, int C[])

{

int a,b,c;

for(a=0, b=n-1, c=0; a<m && b>=0;)

{

if (A[a] <= B[b])

{

C[c++]=A[a++];

}

else

{

C[c++]=B[b--];

}

if (a>m-1)

{

while (b>=0)

{

C[c++]=B[b--];

}

}

if (b<0)

{

while (a<=m-1)

{

C[c++]=A[a++];

}

}

}

}

**19. Insertion in the beginning of a list.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct Node

{

int info;

Node \* next;

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

Node \* Create\_New\_Node(int);

void Insert\_Beg(Node \*);

void Display(Node \*);

void main()

{

start=NULL;

int inf;

char ch='y';

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

{

clrscr();

cout<<"Enter information for he new node: ";

cin>>inf;

cout<<"Creating new node!";

system("pause");

newptr = Create\_New\_Node(inf);

if (newptr != NULL)

{

cout<<"New node created successfully!";

system("pause");

}

else

{

cout<<"Cannot create new node. Aborting.";

getch();

}

cout<<"\nNow inserting this node in the beginning of list.";

system("pause");

Insert\_Beg(newptr);

cout<<"\nNow the list is: ";

Display(start);

cou<<"Press Y to enter more nodes, N to exit: ";

cin>>ch;

}

getch();

}

Node \* Create\_New\_Node(int n)

{

ptr = new Node;

ptr -> info = n;

ptr -> next = NULL;

return ptr;

}

void Insert\_Beg(Node \* np)

{

if ( start == NULL)

start = np;

else

{

save = start;

start = np;

np -> next = save;

}

}

void Display(Node \* np)

{

while (np != NULL)

{

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

np = np -> next;

}

cout<<"!!!\n";

}

**20. Insertion in the end of list.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct Node

{

int info;

Node \* next;

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

Node \* Create\_New\_Node(int);

void Insert\_End(Node \*);

void Display(Node \*);

void main()

{

start = rear = NULL;

int inf;

char ch='y';

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

{

clrscr();

cout<<"Enter information for the new node: ";

cin>>inf;

cout<<"Creating new node.";

system("pause");

newptr = Create\_New\_Node(inf);

if (newptr != NULL)

{

cout<<"Node created successfully!";

system("pause");

}

else

{

cout<<"Cannot create new node. Aborting!";

getch();

}

cout<<"Inserting new node in the end of list.";

system("pause");

Insert\_End(newptr);

cout<<"Now the list is: ";

Display(start);

cout<<"Press Y to enter more node, N to exit.";

cin>>ch;

}

getch();

}

Node \* Create\_New\_Node(int n)

{

ptr = new Node;

ptr -> info = n;

ptr -> next = NULL;

return ptr;

}

void Insert\_End(Node \* np)

{

if (start == NULL)

{

start = rear = np;

}

else

{

rear -> next = np;

rear = np;

}

}

void Display (Node \* np)

{

while (np!=NULL)

{

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

np = np -> next;

}

cout<<"!!!\n";

}

**21. Pushing in Stack - Array.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

int Push(int[], int&, int);

void Display(int[], int);

const int size =10;

void main()

{

int stack[size], item, top=-1, res;

char ch='y';

clrscr();

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

{

cout<<"Enter item for insertion: ";

cin>>item;

res = Push(stack, top, item);

if (res==-1)

{

cout<<"OVERFLOW!! Aborting!";

getch();

}

cout<<"The stack is now: ";

Display(stack, top);

cout<<"Wnat to enter more elements?(y/n)..";

cin>>ch;

}

getch();

}

int Push(int stack[], int & top, int ele)

{

if (top==size-1)

return -1;

else

{

top++;

stack[top]=ele;

}

return 0;

}

void Display(int stack[], int top)

{

cout<<stack[top]<<" <-- "<<endl;

for(int i=top-1; i>=0; i--)

cout<<stack[i]<<endl;

}

**22. Pushing in stack – Linked List.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct Node

{

int info;

Node \* next;

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

Node \* Create\_New\_Node(int);

void Push(Node \*);

void Display(Node \*);

void main()

{

int inf;

char ch='y';

top = NULL;

clrscr();

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

{

cout<<"Enter information for new node: ";

cin>>inf;

newptr = Create\_New\_Ptr(inf);

if(newptr == NULL)

{

cout<<"Cannot create new node. Aborting!";

getch();

}

Push(newptr);

cout<<"Now the linked list is: ";

Display(top);

cout<<"Press Y to enter more nodes, N to exit: ";

cin>ch;

}

getch();

}

Node \* Create\_New\_Node(int n)

{

ptr = new Node;

ptr -> info = n;

ptr -> next = NULL;

return ptr;

}

void Push(Node \* np)

{

if (top==NULL)

{

top = np;

}

else

{

save = top;

top = np;

np -> next = save;

}

}

void Display(Node \* np)

{

while ( np != NULL)

{

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

np = np -> next;

}

cout<<"!!!\n";

}

**23. Insertion and deletion in queue.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

const int size = 50;

int a[size], front=-1, back=-1;

int insert(int[],int);

int remove(int[]);

void display(int[], int, int);

void main()

{

int x, r, choice;

lb:

clrscr();

cout<<"---------Menu----------"<<endl;

cout<<"1. Insert elements in queue"<<endl;

cout<<"2. Delete elements in queue"<<endl;

cout<<"3. Exit"<<endl;

cout<<"Enter choice: "; cin>>choice;

switch (choice)

{

case 1:

char ch = 'y';

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

{

cout<<"\nEnter element for queue: ";

cin>>x;

r = insert (a, x);

if (r==-1)

{

cout<<"\nOVERFLOW!!\n";

exit(1);

}

cout<<"\nQueue is: \n";

display(a, front, back);

cout<<"\nWant to enter another element?(y/n)";

cin>>ch;

}

goto lb;

case 2:

ch = 'y';

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

{

r = remove(a);

if(r==-1)

{

cout<<"\nUNDERFLOW!!\n";

exit(1);

}

display(a, front, back);

cout<<"\nWant to delete another element? (y/n)";

cin>>ch;

}

goto lb;

default: exit(1);

}

getch();

}

int insert(int queue[], int object)

{

if (back==size-1)

return -1;

else if (back==-1)

{

front=back=0;

queue[back]=object;

}

else

{

back++;

queue[back]=object;

}

return 0;

}

int remove(int queue[])

{

int res;

if(front==-1)

return -1;

else

{

res = queue[front];

if (front == back)

front = back= -1;

else

front++;

}

return res;

}

void display(int queue[], int front, int back)

{

if (front==-1)

return;

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

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

cout<<queue[back]<<endl;

}

**24. Insertion and deletion in circular queue.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

#include<stdlib.h>

int insert (int[], int);

int del (int[]);

void display(int[], int, int);

const int size=7;

int a[size], front=-1, back=-1;

void main()

{

start:

clrscr();

int obj, res, choice;

char ch='y';

cout<<"---------Menu-------- "<<endl;

cout<<"1.Insert" <<endl;

cout<<"2.Delete" <<endl;

cout<<"3.Display"<<endl;

cout<<"4.Exit" <<endl;

cout<<"Enter choice: ";cin>>choice;

switch(choice)

{

case 1: while(ch=='y'||ch=='Y')

{

cout<<"\nEnter element for Circular queue: ";cin>>obj;

res = insert(a, obj);

if(res==-1)

cout<<"\nOverflow!!\n";

else

{

cout<<"\nThe circular queue is: \n";

display(a, front, back);

system("pause");

}

clrscr();

cout<<"Do you want to add another element?(y/n)";

cin>>ch;

}

system("pause");

goto start;

case 2: while(ch=='y'||ch=='Y')

{

obj = del(a);

cout<<"Element deleted is: "<<obj<<endl;

display(a, front, back);

system("pause");

clrscr();

cout<<"\nDo you want to delete another element? (y/n)";

cin>>ch;

}

system("pause");

goto start;

case 3: display(a, front, back);

system("pause");

goto start;

default: exit(1);

}while(choice!=4);

getch();

}

int insert(int cque[], int el)

{

if((front==0&&back==size-1) || (back==front-1))

return -1;

else if(back==-1)

front = back = 0;

else if(back == size -1)

back = 0;

else

back++;

cque[back] = el;

return 0;

}

int del(int cque[])

{

int re;

if(front==-1)

return -1;

else

{

re = cque[front];

if(front==back)

front = back = -1;

else if(front==size-1)

front = 0;

else

front++;

}

return re;

}

void display(int cque[], int front, int back)

{

cout<<"\nThe circular queue: "<<"Front(-->), Back(<--)\n";

if(front==-1)

return;

if(back>=front)

{

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

cout<<"-";

cout<<"-->";

for(i=front; i<back; i++)

cout<<cque[i]<<"<-";

cout<<cque[back]<<"<--"<<endl;

}

else

{

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

cout<<cque[i]<<"<-";

cout<<cque[back]<<"<--";

for(; i<front; i++)

cout<<"-";

cout<<"-->";

for(i=front; i<back; i++)

cout<<cque[i]<<"<-";

}

}

**25. Popping from stack – Linked.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct Node

{

int info;

Node \* next;

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

Node \* Create\_New-Node(int);

void Push(Node \*);

void Display(Node \*);

void Pop();

void main()

{

top = NULL;

int inf;

char ch='y';

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

{

cout<<"Enter information for the new node: ";

cin>>inf;

newptr = Create\_New\_Node(inf);

if (newptr == NULL)

{

cout<<"Cannot create new node. Aborting!";

getch();

}

Push(newptr);

cout<<"Press enter to enter more nodes, N to exit: ";

cin>>ch;

}

clrscr();

do

{

cout<<"The stack now is: ";

Display(top);

clrscr();

cout<<"Want to pop an element?(y/n): ";

cin>>ch;

if (ch=='y'||ch=='Y')

{

Pop();

}

}while(ch=='y'||ch=='Y');

getch();

}

Node \* Create\_New\_Node(int n);

{

ptr = new Node;

ptr -> info = n;

ptr -> next = NULL;

return ptr;

}

void Push(Node \* np)

{

if (top==NULL)

{

top=np;

}

else

{

save=top;

top=np;

np -> next = save;

}

}

void Pop()

{

if (top==NULL)

{

cout<<"Underflow!";

}

else

{

ptr = top;

top = top -> next;

delete ptr;

}

}

void Display(Node \* np)

{

while (np != NULL)

{

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

np = np -> next;

}

cout<<"!!!\n";

}

SQL PROGRAMS

1.