**Shree Swami Bachchu Baba**

**Senior Secondary School**

**Keshavpuram, Jalesar Road**

**Firozabad**

**Session 2015-2016**

**Practical File**→**Computer Science**

****

**Submitted By: - Submitted To:-**

**Name:-Chaklesh Yadav Mrs. Mitali Mukherji**

**Roll No:- 24**

**Class-12th “A”**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.no** | **Programs** | **Page No.** | **Remarks** |
| 1 | Program to illustrate Sin Series |  |  |
| 2 | Program to illustrate class employee |  |  |
| 3 | Program to illustrate array implementation |  |  |
| 4 | Program to illustrate class “large” |  |  |
| 5 | Program to illustrate bubble array |  |  |
| 6 | Program to illustrate guessing game using random. |  |  |
| 7 | Program to illustrate binary search |  |  |
| 8 | Program to illustrate insertion in array |  |  |
| 9 | Program to illustrate link list implementation of queue |  |  |
| 10 | Program to illustrate swap using call by value. |  |  |
| 11 | Program to illustrate linear search in array |  |  |
| 12 | Program to illustrate merging in array |  |  |
| 13 | Program to illustrate use of object arrays |  |  |
| 14 | Program to illustrate push stack in array |  |  |
| 15 | Program to illustrate complex algebra |  |  |
| 16 | Program to illustrate WAP stack |  |  |
| 17 | Program to illustrate sequential search technique |  |  |
| 18 | Program to illustrate sequential sort technique |  |  |
| 19 | Program to illustrate selection sort technique |  |  |
| 20 | Program to illustrate traversal in array |  |  |
| 21 | SQL commands |  |  |

**PROGRAMS**

**1. PROGRAM TO ILLUSTRATE THE**

**SIN SERIES**

#include<iostream.h>

#include<conio.h>

#include<math.h>

void main()

{

int i,N;

float X,T,S=0;

cout<<"Enter X and N\n";

cin>>X>>N;

T=X;S=T;

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

{

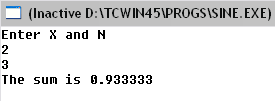
T=T\*(-1)\*X\*X/((2\*i-1)\*(2\*i-2));

S=S+T;

}

cout<<"The sum is "<<S;

}

**1. OUTPUT** 

**2.PROGRAM TO ILLUSTRATE CLASS**

**“EMPLOYEE”**

#include<iostream.h>

#include<stdio.h>

class employee

{ char name[20]; char desig[20]; float basesal; float da()

{float d=(basesal\*3)/5; return d; }

float cal\_salary()

{ float d=da(); float t; t=basesal+d; return t; }

public:

void input()

{ gets(name); gets(desig); cin>>basesal; }

void output()

{ float t;

cout<<endl<<name<<endl<<desig<<endl;t=cal\_salary();

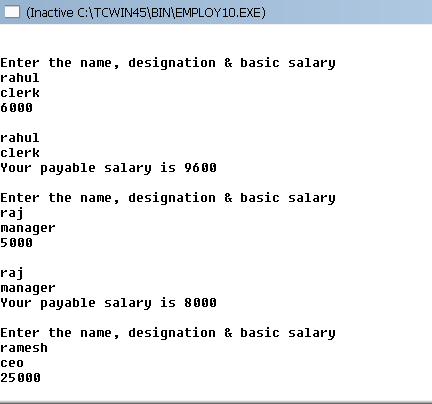
cout<<"Your payable salary is "<<t;}};

void main()

{employee E[5];

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

{cout<<"\nEnter the name, designation & basic salary\n"; E[i].input(); E[i].output();} }

**2.OUTPUT**

**3.PROGRAM TO ILLUSTRATE ARRAY**

**IMPLEMENTATION**

#include<fstream.H>

#include<string.H>

#include<stdio.H>

#include<conio.H>

const N=5;

class queue

{

float A[N]; int rear ;

public:

queue(){rear=-1;}

void addq(); void delq(); void dispq();

};

void queue::addq() {

if(rear==N-1)

cout<<"overflow,addition not possible";

else

{

rear++;

cin>>A[rear];}}

void queue::delq()

{

if(rear==-1)

cout<<"Underflow! queue is empty";

else

{

rear--;

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

A[i]=A[i+1];}}

void queue::dispq() {

if (rear==-1)

cout<<"queue is empty";

else

{

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

cout<<A[i]<<endl;}}

void main()

{

int choice;

queue q;

do{

cout<<"1.ADD\n";

cout<<"2.DELETE\n";

cout<<"3.DISPLAY \n";

cout<<"4.QUIT\n";

choice=getch();

switch(choice)

{

case '1': q.addq(); break;

case '2': q.delq(); break;

case '3': q.dispq(); break;

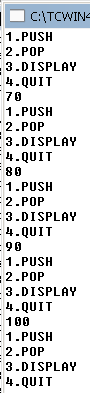
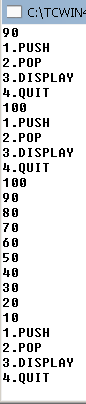
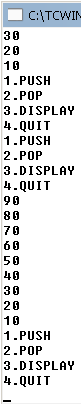
case '4': break;

default :cout<<"\nWrong Choice Entered !!\n\n";}

}while(choice!='4');

}

**3.OUTPUT**

****   

**4.PROGRAM TO ILLUSTRATE CLASS**

**”LARGE”**

#include<iostream.h>

#include<conio.h>

class ITEM

{

int itemcode[50]; float it\_price[50];

public:

void initialize(void); float largest(void); float sum(void);

void display\_item(void); };

void ITEM::initialize(void)

{ for(int i=0;i<50;i++)

{ cout<<"\n"<<"item no:"<<(i+1);

cout<<"\n"<<"enter item code:";

cin>>it\_price[i];

cout<<"\n"; }}

float ITEM::largest(void)

{ float large=it\_price[0];

for(int i=1;i<50;i++)

{ if(large<it\_price[i])

large=it\_price[i]; }

return large; }

float ITEM::sum(void)

{ float sum=0;

for(int i=0;i<50;i++)

sum+=it\_price[i];

return sum; }

void ITEM::display\_item(void)

{ cout<<"\ncode price\n";

for(int i=0;i<50;i++)

{cout<<"\n"<<itemcode[i];

cout<<" "<<it\_price[i];}

cout<<"\n";}

int main()

{ ITEM order;

order.initialize();

float total,biggest;

int ch=0; clrscr();

do

{ cout<<"\nmain menu.\n";

cout<<"\n1.display largest price.";cout<<"\n2.display sum of price.";

cout<<"\n3.display item list.";

cout<<"\nenter your choice(1-3):";

cin>>ch;

switch(ch)

{ case 1:biggest=order.largest();

cout<<"the largest price is"<<biggest<<"\n"; break;

case 2:total=order.sum();

cout<<"the sum of price is"<<total<<"\n"; break;

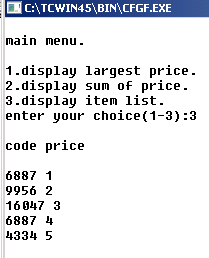
case 3:order.display\_item(); break;

default: cout<<"\nwrong choice!\n"; break; } }

while(ch>=1&&ch<=3); return 0;

}

**4.OUTPUT**



**5.PROGRAM TO ILLUSTRATE BUBBLE**

**IN ARRAY**

#include<iostream.h>

void bubsort(float a[],int n)

{ float t;

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

{for(int j=0;j<n-i-1;j++) {if(a[j]>a[j+1]){t=a[j]; a[j]=a[j+1]; a[j+1]=t;}}}

for(int k=0;k<n;k++)

{cout<<a[k]<<endl;}}

void main()

{

float x[10];

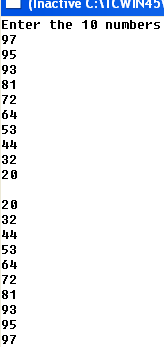
cout<<"Enter the 10 numbers\n";

for(int i=0;i<10;i++)

{cin>>x[i]; }

cout<<endl; bubsort(x,10);}

**5.OUTPUT**

****

**6. PROGRAM TO ILLUSTRATE GUESSING GAME USING RANDOM().**

#include<iostream.h>

#include<stdlib.h>

void main()

{

int x,n[5];

cout<<"Guess the number between 1-15 within 5 chances\n";

randomize();

x=random(15)+1;

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

{

cin>>n[i];

if(n[i]==x)

{cout<<"Correct!!\n";break; }

else

{

if(n[i]>x)

cout<<"Too High\n";

else

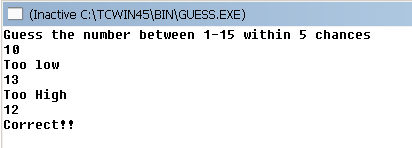
cout<<"Too low\n";

}

}

}

**6. OUTPUT**

****

**7. PROGRAM TO ILLUSTRATE BINARY**

**SEARCH**

#include<iostream.h>

Int (float a[],int n, float data)

{ int l=0,u=n-1,m,found=0;

while(found==0&&u>l)

{

m=(l+u)/2; if(a[m]==data)

{ found=1;}

else if(a[m]>data) {u=m-1;}

else {l=m+1;}

}

return found;

}

void main()

{ float x[10] ; float d; int f;

cout<<"Enter the 10 numbers \n";

for(int j=0;j<10;j++) { cin>>x[j]; }

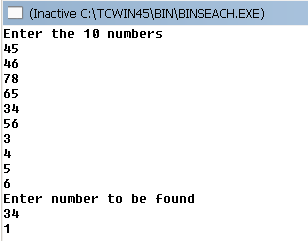
cout<<"Enter number to be found\n";

cin>>d; f=binasearch(x,10,d);

cout<<f;

}

**7.OUTPUT**

****

**8. PROGRAM TO ILLUSTRATE**

**INSERTION IN ARRAY**

#include<iostream.h>

#include<process.h>

int findpos(int[],int,int);

int main()

{

int AR[50],item,n,index;

cout<<"how many elements do you want to crerate with? (max.50)";

cin>>n;

cout<<"\n enter array elements(must be sorted in asc order)\n";

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

cin>>AR[i];

char ch='y';

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

{

cout<<"\n enter element to be inserted";

cin>>item;

if(n==50)

{

cout<<"overflow!!\n";

exit(1);

}

index=findpos(AR,n,item);

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

{

AR[i]=AR[i-1];

}

AR[index]=item;

n+=1;

cout<<"\n want to insert more elements?(y\n)";

cin>>ch;

}

cout<<"the array now is as shown below\n";

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

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

cout<<endl;

return 0;

}

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

{

int pos;

if(item <AR[0]) pos=0;

else

{

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

{

if(AR[i]==item&&item<AR[i+1])

{

pos=i+1; break;

}}

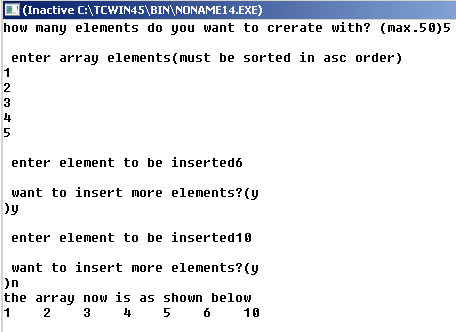
if(i==size-1)

pos=size;

} return pos;

}

**8.OUTPUT**

****

**9.PROGRAM TO ILLUSTRATE LINKED**

**LIST IMPLEMENTATION OF QUEUE**

#include<iostream.h>

#include<stdio.h>

struct node

{ char name[20];

int marks;

node \*next;

};

class queue

{ node \*front;

node \*rear;

public:

queue(){front=rear=NULL;}

void addq(); void delq(); void display();

~queue();

};

void queue::addq()

{ node \*temp; temp=new node; gets(temp->name);

cin>>temp->marks; temp->next=NULL;

if(front!=NULL)

{ rear->next=temp;

rear=temp; }

else

front=rear=temp; }

void queue::delq()

{ if(front==NULL)

cout<<"Underflow!!Queue is empty\n";

else { node \*temp;

temp=front;

front=front->next; delete(temp);

if(front==NULL)rear=NULL; } }

void queue::display()

{ if(front==NULL)

cout<<"Queue is empty\n";

else

{ node \*temp; temp=front;

while(temp!=NULL)

{cout<<temp->name;

cout<<temp->marks;

temp=temp->next; } } }

queue::~queue()

{ node \*temp;

while(front!=NULL)

{ temp=front; front=front->next; delete(temp); } }

void main()

{ queue q; int ch;

do

{ cout<<"1.ADD\n2.DEL\n3.DISPLAY\n4.QUIT\n";

cin>>ch;

switch(ch)

{ case 1: q.addq();break;

case 2: q.delq();break;

case 3: q.display();break;

case 4: break; } }while(ch!=4) }

**9. OUTPUT**

** **

**10. PROGRAM TO ILLUSTRATE**

**SWAP USING CALL BY VALUE.**

#include<iostream.h>

void swap(int a,int b)

{

int c=a;

a=b;

b=c;

}

void main()

{

int x,y;

cout<<"Enter x and y\n";

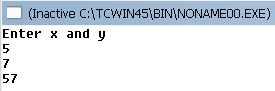
cin>>x>>y;

swap(x,y) ;

cout<<x<<y;

}

**10. OUTPUT**

****

**11. PROGRAM TO ILLUSTRATE**

**LINEAR SEARCH**

#include<iostream.h>

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

int main()

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

cout<<"enter array";

cin>>N;

cout<<"enter array element";

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

{cin>>AR[i]; }

cout<<"enter element search";

cin>>ITEM;

index=Lsearch(AR,N,ITEM);

if(index==-1)

cout<<"element not found";

else

cout<<"elementfound at index:"<<index<<",position:"<<index+1<<endl;

return 0;

}int Lsearch(int AR[],int size,int item)

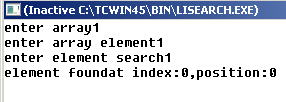
{ for(int i=0;i<size;i++)

{ if(AR[i]==item)

return i;

}return-1;

}

**11.OUTPUT **

**12.PROGRAM TO ILLUSTRATE**

**MERGING IN ARRAY**

#include<iostream.h>

void merge(float a[],int m,float b[],int n,float c[]){ int i=0,j=0,k=0;

while(i<m&&j<n)

{ if(a[i]<b[j])

{ c[k]=a[i]; k++; i++; }

Else { c[k]=b[j]; k++; j++; } }

while(i<m)

{ c[k]=a[i]; k++; i++; }

while(j<n)

{ c[k]=b[j]; k++; j++; }

for(int p=0;p<m+n;p++)

cout<<c[p]<<endl;}

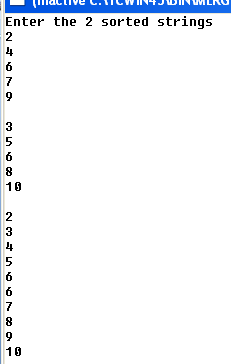
void main()

{ float x[5],y[5],z[10];

cout<<"Enter the 2 sorted strings\n";for(int f=0;f<5;f++) cin>>x[f]; cout<<endl;

for(int g=0;g<5;g++)cin>>y[g]; cout<<endl;merge(x,5,y,5,z);}

**12. OUTPUT**

****

**13. PROGRAM TO ILLUSTRATE USE**

**OF OBJECT ARRAYS**

#include<iostream.h>

class Item { int itemno;

float price ;

public :

void getdata(int i,float j)

{ itemno=i; price=j;

} void putdata(void

{cout<<"itemno: <<itemno; cout<<"price: <<price;}};

const int size=10;

Item order[size];

int main()

{ int ino ; float cost;

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

{ cout<<"Enter itemno & price for item"<<a+1;

cin>>ino>>cost;

order[a].getdata(ino,cost); }

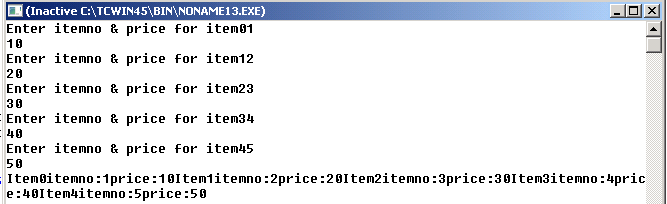
for(a=0;a<size;a++)

{ cout<<"Item"<<a+1;

order[a].putdata();

} return 0; }

**13.OUTPUT**

****

**14.PROGRAM TO ILLUSTRATE STACK**

**PUSH**

#include<iostream.h>

#include<conio.h>

#include<process.h>

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

void Display(int[],int);

const int size=50;

void main()

{

int Stack[size],Item,top=-1,res;

char ch='y';

clrscr();

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

{cout<<"\n Enter ITEM for insertion:";

cin>>Item;

res=Push(Stack,top,Item);

if(res==-1)

{cout<<"OVERFLOW!!!Aborting!!\n";

exit(1);

}

cout<<"\n The Stack now is :\n";

Display(Stack,top);

cout<<"\n Want to insert more element?(y/n)...";

cin>>ch;

}

}

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)

{

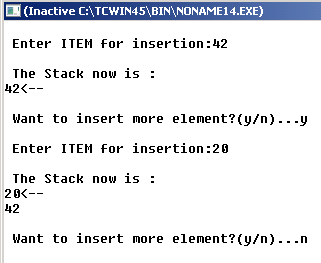
cout<<Stack[top] <<"<--"<<endl;

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

cout<<Stack[i]<<endl;

}

**14.OUTPUT**

****

**15.PROGRAM TO ILLUSTRATE**

**COMPLEX ALGEBRA**

#include<iostream.h>

void main()

{

int a,b,c,d,cho;

cout<<"Enter the real and imagimary parts of the 2 complex numbers:\n";

cin>>a>>b>>c>>d;

cout<<"Choose the operation\n"<<"1.Add\n2.Difference\n3.Multiply\n\n";

cin>>cho;

switch(cho)

{

case 1:cout<<a+c<<"+"<<b+d<<"i";

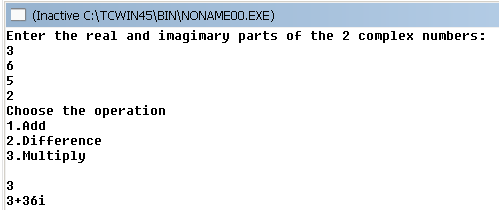
break;

case 2:cout<<a-c<<"+"<<b-d<<"i";

break;

case3:cout<<(a\*c)-(b\*d)<<"+"<<(a\*d)+(b\*c)<<"i"; } }

**15.OUTPUT**

****

**16.PROGRAM TO ILLUSTRATE WAP**

**STACK**

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

struct node

{ char name[20];

int marks;

node \*next;

};

class stack

{ node \*top;

public:

stack()

{top=NULL;} void push(); void pop();void display();

~stack(); };

void stack::push()

{

node \*temp;

temp=new node;

gets(temp->name);

cin>>temp->marks;

temp->next=top;

top=temp; }

void stack::pop()

{ if(top==NULL)

cout<<"Underflow!!Stack is empty\n";

else

{ node \*temp; temp=top;

top=top->next;

delete(temp); } }

void stack::display()

{ if(top==NULL)

cout<<"Stack is empty\n";

else

{

node \*temp;

temp=top;

while(temp!=NULL)

{ cout<<temp->name;

cout<<temp->marks;

temp=temp->next; } } }

stack::~stack()

{ node \*temp;

while(top!=NULL)

{ temp=top; top=top->next; delete(temp); } }

void main()

{ stack s; int ch;

do

{ cout<<"1.PUSH\n2.POP\n3.DISPLAY\n4.QUIT\n";

ch=getch();

switch(ch)

{ case 1: s.push();break;

case 2: s.pop();break;

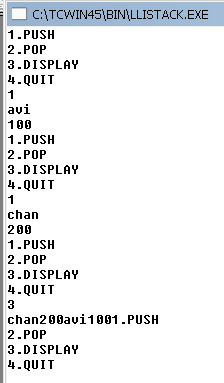
case 3: s.display();break;

case 4: break;

default:cout<<"Wrong Choice";} }

while(ch!=4); }

**16.OUTPUT**

****

**17. PROGRAM TO ILLUSTRATE**

**SEQUENTIAL SEARCH TECHNIQUE**

#include<iostream.h>

int seqsearch(float a[],int n,float data)

{ int found =0; for(int i=0;i<n;i++) { if(a[i]==data) found=1; }

return found; }

void main()

{ float x[10] ; float d; int f;

cout<<"Enter the 10 numbers \n";

for(int j=0;j<10;j++)

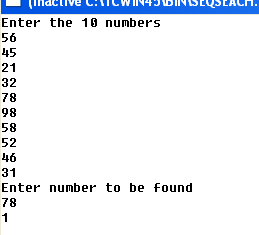
{cin>>x[j];}

cout<<"Enter number to be found\n";

cin>>d; f=seqsearch(x,10,d); cout<<f;

}

**17. OUTPUT**

****

**18.PROGRAM TO ILLUSTRATE**

**SEQUENTIAL SORT TECHNIQUE**

#include<iostream.h>

void seqsort(float a[],int n)

{ float temp;

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

{ for(int j=i+1;j<n;j++)

{ if(a[i]>a[j])

{ temp=a[i]; a[i]=a[j]; a[j]=temp; } } }

for(int k=0;k<n;k++)

{ cout<<a[k]<<endl;} }

void main()

{ float x[10];

cout<<”Enter the 10 numbers\n”;

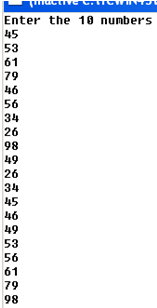
for(int i=0;i<10;i++)

{ cin>>x[i]; }

seqsort(x,10);

}

**18. OUTPUT**



**19.PROGRAM TO ILLUSTRATE**

**SELECTION SORT TECHNIQUE**

#include<iostream.h>

void sel\_sort(float A[], int N)

{ int i,j,c,M; float Temp;

cout<<"Original Array : "<<endl;

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

cout<<A[c]<<'\t'; cout<<endl<<endl;

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

{ M=i;

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

{ if(A[M]>A[j])

M=j; } if(M!=i)

{ Temp=A[M]; A[M]=A[i]; A[i]=Temp; }

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

cout<<A[c]<<'\t';

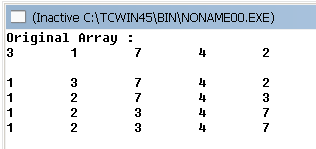
cout<<endl; } }

void main()

{ float X[5]={3,1,7,4,2};

sel\_sort(X,5);}

**19.OUTPUT**



**20.PROGRAM TO ILUSTRATE TRAVERSAL IN ARRAY**

#include<iostream.h>

int main()

{

int ar[50],item,n,index;

cout<<"how many elements enter";

cin>>n;

cout<<"enter array elements";

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

cin>>ar[i];

cout<<"array with doubled elements as follows";

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

{

ar[i]\*=2;

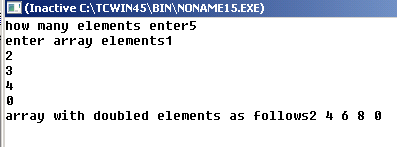
cout<<ar[i]<<" ";}

cout<<endl;

return 0;

}

**20.OUTPUT**

****

**21.SQL TABLES**

**SQL : 1**

**Table: INTERIORS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **ITEMNAME** | **TYPE** | **DATEOFSTOCK** | **PRICE** | **DISCOUNT** |
| 1 | Red rose | Double Bed | 23/02/11 | 32000 | 15 |
| 2 | Soft touch | Baby cot | 20/01/11 | 9000 | 10 |
| 3 | Jerry’s home | Baby cot | 19/02/11 | 8500 | 10 |
| 4 | Rough wood | Office table | 01/01/11 | 20000 | 20 |
| 5 | Comfort zone | Double bed | 12/01/11 | 15000 | 20 |
| 6 | Jerry look | Baby cot | 24/02/11 | 7000 | 19 |
| 7 | Lion king | Office table | 20/02/11 | 16000 | 20 |
| 8 | Royal tiger | Sofa | 22/02/11 | 30000 | 25 |
| 9 | Park sitting | Sofa | 13/12/11 | 9000 | 15 |
| 10 | Dine Paradise | Dining Table | 19/02/11 | 11000 | 15 |

1. To show all information about the Sofa ’s from the INTERIORS table.

**Select \* from interiors where type=”sofa”;**

1. To list the item name which are priced at more than 10000 from the INTERIORS table

**Select itemname from interiors where price>10000;**

1. To display item name and date of stock of those items, in which the discount percentage is more than 15 from INTERIORS table.

**Select itemname, dateofstock from interiors where discount >15;**

1. To list item name and type of items, in which date of stock is before 22/01/11 from the INTERIORS table in descending order of item name.

**Select itemname,type from interiors where dateofstock<22/01/11**

**Order by itemname desc;**

1. To count the number of items of each type from INTERIORS table.

**Select count(distinct type) from interiors;**

1. To insert new row in INTERIORS table with following data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 11 | White wood | Double Bed | 23/02/06 | 20000 | 20 |

**Insert into interiors**

**Field(NO, ITEMNAME, TYPE, DATEOFSTOCK, PRICE, DISCOUNT)**

**Values(11,” White wood”,” Double Bed”,” 23/02/06”, 20000, 20);**

1. To count distinct types from INTERIORS table.

**Select count(distinct type) from interiors;**

1. To find average discount from INTERIORS for each type of Interiors.

**Select avg(discount distinct type) from interiors;**

1. To calculate sum of price from INTERIORS where date of stock is before 12/02/11.

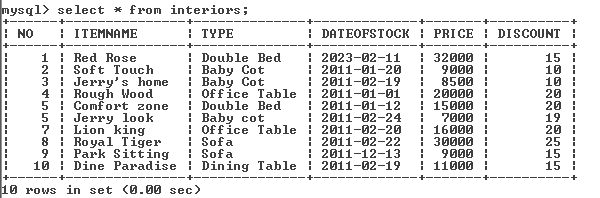
**Select sum(price) from interiors where dateofstock<12/02/11;**

1. To increase price of all Office table by 3000 from INTERIORS table.

**Update table interiors**

**Set price= price+3000 where type=”office table”;**

**OUTPUT**

**

**SQL : 2**

**Table: BOOKS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BOOK\_ID** | **BOOK\_NAME** | **AUTHOR\_NAME** | **PUBLISHERS** | **PRICE** | **TYPE** | **QUANTITY** |
| C0001 | Fast Cook | Lata Kapoor | EPB | 355 | Cookery | 5 |
| F0001 | The Tears | William Hopkins | First Publishers | 650 | Fiction | 20 |
| T0001 | My First C++ | Brain & Brroke | EPB | 350 | Text | 10 |
| T0002 | C++ Brainworks | A.M.Rossaine | TDH | 350 | Text | 15 |
| F0002 | Thunderbolts | Anna Roberts | First Publishers | 750 | Fiction | 50 |

**Table: ISSUED**

|  |  |
| --- | --- |
| **BOOK\_ID** | **QUANTITY\_ISSUED** |
| T0001 | 4 |
| C0001 | 5 |
| F0001 | 2 |

To show book name, author name and price of books of First Publishers.

**Select book\_name,author\_name,price from books where publishers=”First Publishers”;**

1. To list the names from books of Text type.

**Select book\_name from books where type=”Text”;**

1. To display the names and price from books in ascending order of their price.

**Select book\_name,price from books order by price;**

1. To increase the price of all books of EPB publishers by 10%.

**Update books set price=price+(price/10) where publishers=”EPB”;**

1. To display the book id, book name and quantity issued for all books which have been issued.

**Select book\_id,book\_name,quantity\_issued from books, issued**

**Where issued.book\_id=books.book\_id;**

1. To insert new row in the table issued having the following data:

‘F3003’,1

**Insert into issued**

**Field(book\_id,quantity\_issued)**

**Values(“F3003”,1);**

1. To display the total no. of books (quantity) of each type.

**Select count(distinct type) from books;**

1. To find the maximum price from books which have quantity more than 15.

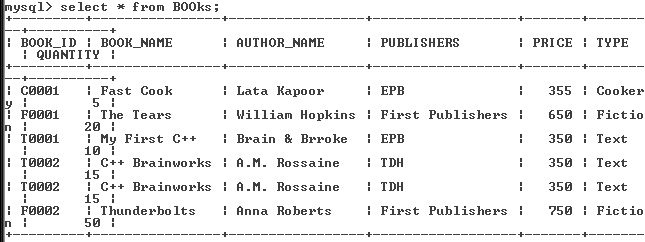
**Select max(price) from books where quantity>15;**

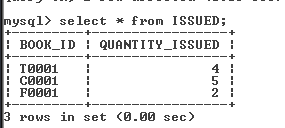
1. To count unique publishers from books & cost less than 400.

**Select count(distinct publishers) from books where price<400;**

1. To display the no. of titles available of each publishers.

**OUTPUT**

**

**

**SQL : 3**

**Table: GAMES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GCODE** | **GAMENAME** | **NUMBER** | **PRIZEMONEY** | **SCHEDULEDATE** |
| 101 | Carom Board | 2 | 5000 | 23-Jan-2011 |
| 102 | Badminton | 2 | 12000 | 12-Dec-2010 |
| 103 | Table Tennis | 4 | 8000 | 14-FEB-2011 |
| 105 | Chess | 2 | 9000 | 01-Jan-2011 |
| 108 | Lawn Tennis | 4 | 25000 | 19-Nov-2010 |

**Table: PLAYER**

|  |  |  |
| --- | --- | --- |
| **PCODE** | **NAME** | **GCODE** |
| 1 | Nabi Ahmad | 101 |
| 2 | Ravi Sahai | 108 |
| 3 | Jatin | 101 |
| 4 | Nazneen | 103 |

1. To display the names of all games where schedule date is after 31-Dec-2010.

**Select gamename from games where scheduledate>”31-Dec-2010”;**

1. To display the name of the players in the table player with their gamesname.
2. To display the content of the games table in ascending order of schedule date.

**Select \* from games order by scheduledate;**

1. To display the sum of price money received by the players in the table player.

**Select sum(prizemoney) from games;**

1. To add a new column (attribute) ADDRESS of type Character (20) in the table PLAYER

**Alter table player**

**Add**

**(Address char(20));**

1. To display maximum and min. schedule date.

**Select max(scheduledate),min(scheduledate) from games;**

1. To change the prize money for Chess from 9000 to 10000.

**Update table games**

**Set prizemoney=10000;**

1. To display unique gcodes from players.

**Select distinct gcode from players;**

1. To delete all rows from games table where schedule date was before 01-Jan-2011.

**Delete \* from games where scheduledate<01-Jan-2011;**

1. To insert a new record in the player table as follows:

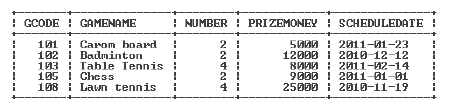
|  |  |  |
| --- | --- | --- |
| 5 | Anand | 105 |

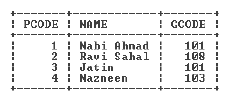
**Insert into player**

**Field(Pcode,Name,Gcode)**

**Values(5,”Anand”,105);**

**OUTPUT**



**

**Bibliography**

All the information in the project has been gathered from internet.

**Websites used:**

* Google
* Free encyclopedia on Computer Science
* iCBSE.com
* cppforschool.blogspot.in