#include<iostream>

using namespace std;

class myarray{

public:

void writearray(int arr[]){

cout<<"Enter the arrray\n";

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

//cout<<"Enter the arrray\n";

cin>>arr[i];

}

}

void printarray(int arr[],int x){

int loc=0;

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

if(arr[i]==x){

loc=i;

}

}

cout<<"value is searched is:"<<loc;

}

};

int main(){

int \*arr=new int[5];

myarray\*am=new myarray();

int n;

am->writearray(arr);

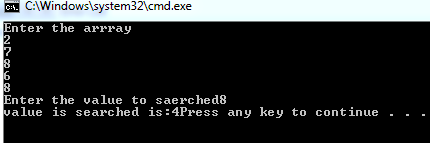
cout<<"Enter the value to saerched";

cin>>n;

am->printarray(arr,n);

return 0;

}



#include<iostream>

using namespace std;

class programme{

public:

int arr[2][2];

void writearray(){

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

for(int j=0;j<2;j++){

cin>>arr[i][j];

cout<<"\t";

cout<<"\n";

}

}

}

void printarray(){

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

for(int j=0;j<2;j++){

cout<<arr[i][j]<<"\t";

}

}

cout<<"\n";

}

};

int main(){

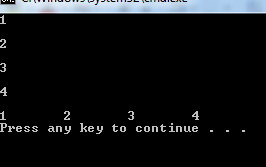
programme am;

am.writearray();

am.printarray();

return 0;

}



#include<iostream>

using namespace std;

void insertion(int A[],int N){

int ptr;

int temp;

for(int k=0;k<N;k++){

temp=A[k];

ptr=k-1;

while(temp<A[ptr]){

A[ptr+1]=A[ptr];

ptr=ptr-1;

}

A[ptr+1]=temp;

}

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

cout<<A[j]<<endl;

}

}

int main(){

const int N=6;

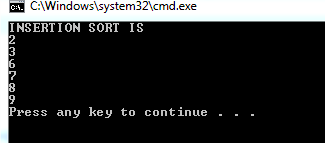
int A[N]={7,8,3,6,2,9};

cout<<"INSERTION SORT IS\n";

insertion(A,N);

return 0;

}



#include<iostream>

using namespace std;

void push(int);

void pop(int);

void display();

int top=0,stack[6];

int main(){

int flag=0;

int data=0;

do{

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

cout<<"\nEnter the 1 for push \n Enter the 2 for pop \n Enter the 3 for display\n Enter the 4 for Exit"<<endl;

cin>>flag;

if(flag==1){

cin>>flag;

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

push(data);

}

else if(flag==2){

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

pop(data);

}

else if(flag==3){

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

display();

}

}while(flag!=4);

return 0;

}

void push(int data){

if(top==5)

{cout<<"OVERFLOW";

}

stack[top]=data;

top=top+1;

cout<<top;

}

void pop(int data){

if(top==0){

cout<<"UNDERFLOW";

}

else{

data=stack[top];

top=top-1;

cout<<top;

}

}

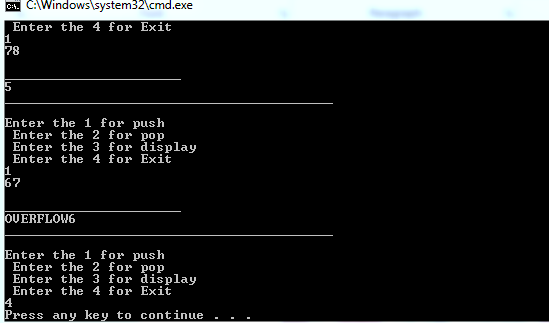
void display(){

for(int j=0;j<top;j++){

cout<<"\nstack["<<j<<"]="<<stack[j]<<endl;

}

}



#include<iostream>

using namespace std;

void enqueue(int);

void dequeue();

void display();

const int n=6;

int queue[n];

int rear=0;

int front=0;

int main(){

int flag=0;

int data=0;

do{

cout<<"\n Enter the 1 for enqueue \nEnter the 2 for dequeue \n Enter the 3 for display\n Enter the 4 for Exit"<<endl;

cin>>flag;

if(flag==1){

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

enqueue(data);

}

else if(flag==2){

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

dequeue();

}

else if(flag==3){

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

display();

}

}

while(flag!=4);

return 0;

}

void enqueue(int data){

if(rear==n){

cout<<"queue is full";

}

else{

cout<<"Enter the value is to inserted ";

cin>>data;

queue[rear]=data;

rear++;

}

}

void dequeue(){

int temp=0;

if(front==n){

cout<<"\nfront ison maximum queue";

}

else if(front==rear){

cout<<"front can not cross the rear";

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"<<endl;

}

else{

cout<<queue[front]<<"Delete from queue";

queue[front]=temp;

front++;

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\n"<<front<<endl;

}

}

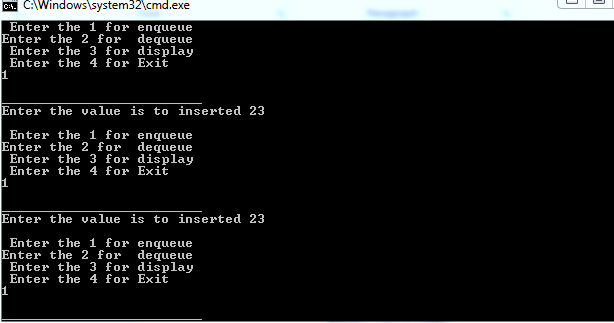
void display(){

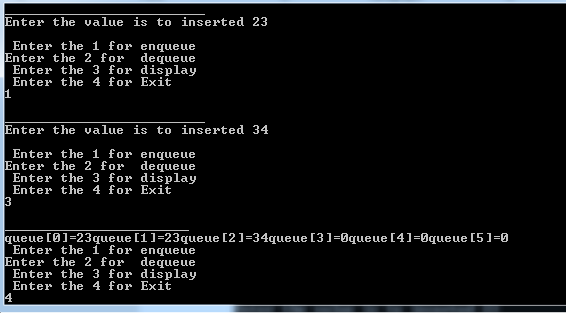
for(int j=0; j<n;j++){

cout<<"queue["<<j<<"]="<<queue[j];

}

}





#include<iostream>

using namespace std;

void qinsert(int);

void qdelete();

void qdisplay();

const int n=5;

int queue[n],rear=0,front=0;

int main(){

int flag=0;

int data=0;

cout<<"Queue wil be of"<<n;

do{

cout<<"\nEnter the 1 for insertion"<<endl;

cout<<"Enter the 2 deletion"<<endl;

cout<<"Enter the 3 for display"<<endl;

cout<<"Enter the 4 for Exit"<<endl;

cin>>flag;

if(flag==1){

qinsert(data);

}

else if(flag==2){

qdelete();

}

else if(flag==3){

qdisplay();

}

}while(flag!=4);

return 0;

}

void qinsert(int item){

if(rear==n){

cout<<"\n no space"<<endl;

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

}

else{

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

cout<<"rear value:"<<rear<<endl;

cout<<"\nEnter the value to insert queue:";

cin>>item;

queue[rear]=item;

rear++;

cout<<"There is rear:"<<rear<<endl;

}

}

void qdelete(){

int temp=0;

if(front==n){

cout<<"\n front on maximum"<<endl;

}

else if(front==rear){

cout<<"\n No value ahead"<<endl;

}

else{

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"<<endl;

cout<<"front value is:"<<front<<endl;

cout<<queue[front]<<"delete from queue";

queue[front]=temp;

front++;

cout<<"\n Now front is on:"<<front<<endl;

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"<<endl;

}

}

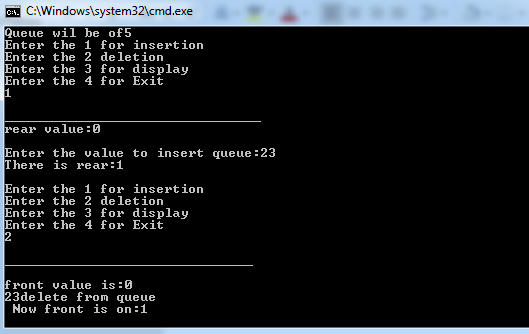
void qdisplay(){

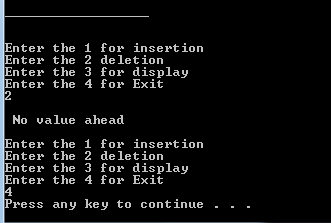
for(int i=0;i<n;i++){

cout<<queue[i];

}

}





#include<iostream>

using namespace std;

void qinsert(int);

void qdelete();

void qdisplay();

const int n=5;

int queue[n],rear=0,front=0;

int main(){

int flag=0;

int data=0;

cout<<"Queue wil be of"<<n;

do{

cout<<"\nEnter the 1 for insertion"<<endl;

cout<<"Enter the 2 deletion"<<endl;

cout<<"Enter the 3 for display"<<endl;

cout<<"Enter the 4 for Exit"<<endl;

cin>>flag;

if(flag==1){

qinsert(data);

}

else if(flag==2){

qdelete();

}

else if(flag==3){

qdisplay();

}

}while(flag!=4);

return 0;

}

void qinsert(int item){

if(rear>=n){

if(front>0){

rear=0;

}

}

if(queue[rear]==0){

cout<<"rear value"<<rear<<endl;

cout<<"\n Enter value to insert queue:";

cout<<item;

cout<<endl;

queue[rear]=item;

rear++;

cout<<"Now rear is on"<<rear<<endl;

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

}

else{

cout<<"Queue is full-value!!"<<endl;

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

}

}

void qdelete(){

int temp=0;

if(front>=n){

if(rear>0){

front=0;

}

}

if(queue[front]!=0){

cout<<"front value"<<front<<endl;

cout<<queue[front]<<"Delete from queue"<<endl;

queue[front]=temp;

front++;

cout<<"\n front is on"<<front<<endl;

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

}

else{

cout<<"Front cannot cross rear"<<endl;

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

}

}

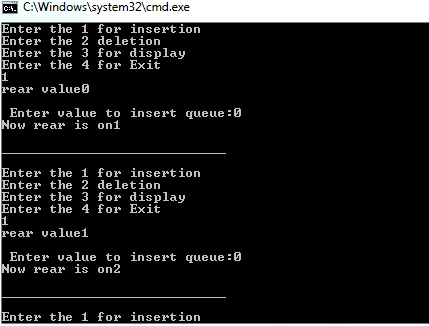
void qdisplay(){

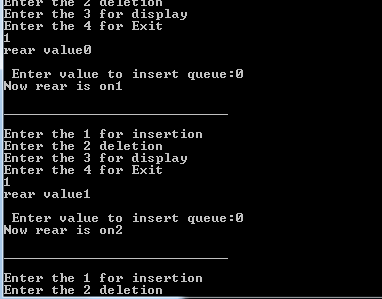
for(int i=0;i<n;i++){

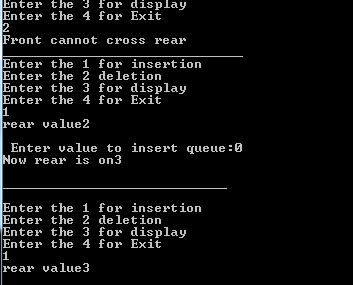
cout<<queue[i];

}

}







#include<iostream>

using namespace std;

struct node{

int data;

node\*next;

};

class linklist{

private:

node\*head;

public:

linklist(){

head=NULL;

}

void insert(int a){

node\*temp;

node\*newnode=new node();

newnode->data=a;

newnode->next=NULL;

if(head==NULL){

head=newnode;

}

else{

temp=head;

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=newnode;

}

}

void Search(int search){

int loc=0;

node\*temp=head;

while(temp!=NULL){

loc++;

if(search==temp->data){

cout<<"\n item is:"<<search;

cout<<"\n location is:"<<loc;

}

temp=temp->next;

}

}

void display(){

node\*temp=head;

cout<<"\n list:";

while(temp!=NULL){

cout<<temp->data<<"";

temp=temp->next;

}

}

};

int main(){

linklist l1;

int no,value,search;

cout<<"Enter the length of node";

cin>>no;

for(int i=0;i<no;i++){

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

cout<<"\nEnter the to insert:";

cin>>value;

l1.insert(value);

cout<<endl;

l1.display();

}

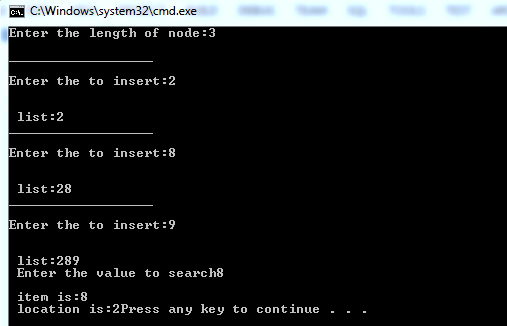
cout<<"\n Enter the value to search";

cin>>search;

l1.Search(search);

return 0;

}



#include<iostream>

using namespace std;

struct node{

int data;

node\*next;

};

class linklist{

private:

node\*head;

public:

linklist(){

head=NULL;

}

void list(int a){

node\*temp;

node\*newnode=new node();

newnode->data=a;

newnode->next=NULL;

if(head==NULL){

head=newnode;

}

else{

temp=head;

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=newnode;

}

}

void sort(){

node\*i;

node\*j;

int temp=0;

for(i=head;i!=NULL;i=i->next){

for(j=i->next;j!=NULL;j=j->next){

if(i->data > j->data){

temp=i->data;

i->data=j->data;

j->data=temp;

}

}

}

}

void display(){

node\*temp=head;

cout<<"\n After sort list is:";

while(temp!=NULL){

cout<<temp->data<<" ";

temp=temp->next;

}

}

};

int main(){

linklist l1;

int value, no;

cout<<"\n Enter the length of node:";

cin>>no;

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

for(int i=0;i<no;i++){

cout<<"\n Enter the insert the value:";

cin>>value;

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

l1.list(value);

l1.sort();

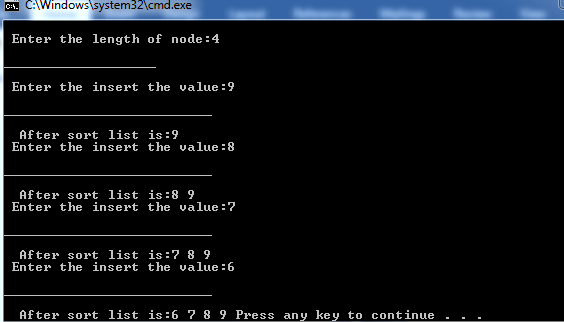
//cout<<"\n after sort is:"<<endl;

l1.display();

}

return 0;

}



#include<iostream>

#include<string>

using namespace std;

struct node{

int serialno;

int rolnum;

string name;

float gpa;

char grade;

node\*next;

};

class studentprofile{

private:

node\*head;

public:

studentprofile(){

head=NULL;}

void profilelist(int sno,int rno,string n,float GPA,char g){

node\*newnode=new node();

//node\*temp=head;

if(head==NULL){

newnode->serialno=sno;

newnode->rolnum=rno;

newnode->name=n;

newnode->gpa=GPA;

newnode->grade=g;

newnode->next=NULL;

}

else{

newnode->serialno=sno;

newnode->rolnum=rno;

newnode->name=n;

newnode->gpa=GPA;

newnode->grade=g;

newnode->next=head;

}

head=newnode;

}

void print(){

node\*temp=head;

cout<<"\nserial no |\t Rollnumber |\tName |\tGpa |\tGrade";

cout<<"\n |\t |\t |\t |\t ";

while(temp!=NULL){

cout<<endl<<""<<temp->serialno<<""<<temp->rolnum<<""<<temp->name<<""<<temp->gpa<<""<<temp->grade;

temp=temp->next;

}

}

};

int main(){

studentprofile s1;

int ser,rool,no;

string nme;

char grde;

float gps;

cout<<"ENTER THE LIST OF AS YOU WANT:";

cin>>no;

for(int i=0;i<no;i++){

cout<<"Enter the serial number:";

cin>>ser;

cout<<"Enter the Rollnumber:";

cin>>rool;

cout<<"Enter the Name of student:";

cin>>nme;

cout<<"Enter the Gpa of student:";

cin>>gps;

cout<<"Enter the grade of student:";

cin>>grde;

s1.profilelist(ser,rool,nme,gps,grde);

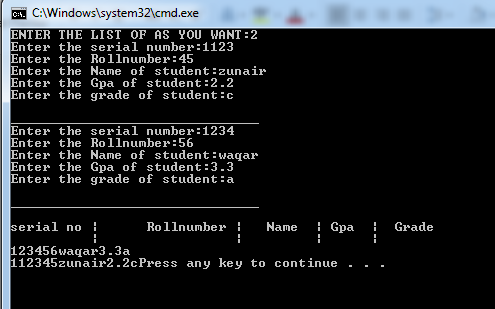
cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

s1.print();

return 0;

}



#include<iostream>

using namespace std;

struct node{

int data;

node\*next;

};

class linklist{

private:

node\*head;

public:

linklist(){

head=NULL;

}

void insertbeg(int a){

node\*newnode=new node();

newnode->data=a;

if(head=NULL){

newnode->next=NULL;

}

else{

newnode->next=head;

}

head=newnode;

}

void insertmid(int a,int n){

node\*temp=head,\*pre=NULL;

node\*newnode=new node();

newnode->data=a;

if(n==1){

insertbeg(a);

}

else{

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

temp=temp->next;

}

newnode->next=temp->next;

temp->next=newnode;

}

}

void insertend(int a){

node\*temp=head,\*pre;

node\*newnode=new node();

newnode->data=a;

if(head==NULL){

insertbeg(a);

}

else{

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=newnode;

}

}

void print(){

node\*temp=head;

cout<<"list is:";

while(temp!=NULL){

cout<< ""<<temp->data;

temp=temp->next;

}

}

};

int main(){

linklist l1;

int flag=0,k,l;

do {

cout<<"Enter the 1 for insert begning:"<<endl;

cout<<"Enter the 2 for insert mid:"<<endl;

cout<<"Enter the 3 for insert end:"<<endl;

cout<<"Enter the 4 print"<<endl;

cin>>flag;

if(flag==1)

{

l1.insertbeg(k);

}

if(flag==2){

l1.insertmid(k,l);

}

if(flag==3){

l1.insertend(k);

}

if(flag==4){

l1.print();

}

}while(flag!=5);

return 0;

}