Radix sort

#include<iostream>  
using namespace std;  
struct node  
{  
        int data;  
        struct node \*link;  
}\*first=NULL;  
void display(struct node\*);  
void radix(struct node \*);  
int main()  
{  
        struct node \*newnode,\*temp;  
        char ans;  
        do  
        {  
                cout<<"enter data";  
                newnode=new node;  
        newnode->link=NULL;  
        cin>>newnode->data;  
        if(first==NULL)  
        {  
                first=newnode;  
                temp=newnode;  
        }  
        else  
        {  
            temp->link=newnode;  
            temp=newnode;  
        }  
        cout<<"do you want to continue(y/n)";  
        cin>>ans;  
        }while(ans!='n');  
        cout<<"\n inserted data";  
        display(first);  
        cout<<"\n sorting using radix sort";  
        radix(first);  
        return 0;  
}  
void display(struct node \*temp)  
{  
    if(temp==NULL)  
    {  
        cout<<"linked list is empty";  
    }  
    else  
    {  
        while(temp!=NULL)  
        {  
                cout<<temp->data<<"-->";  
                temp=temp->link;  
        }  
    }  
}  
void radix(struct node \*first)  
{  
        int j,m=0,c,i,q,k,p=0,l,d,h;  
     struct node\*temp=first,\*t[10],\*b[10],\*next,\*r,\*prev;  
     for(temp=first;temp!=NULL;temp=temp->link)  
     {  
         q=temp->data;  
         c=0;  
         while(q!=0)  
         {  
                 q=q/10;  
                 c++;  
         }  
         if(c>m)  
         {  
                 m=c;  
         }  
     }  
     for(j=1;j<=m;j++)  
     {  
         for(i=0;i<=9;i++)  
         {  
                 t[i]=b[i]=NULL;  
         }  
         r=first;  
         while(r!=NULL)  
         {  
                 h=r->data;  
                 for(k=0;k<j;k++)  
                 {  
                         d=h%10;  
                         h=h/10;  
                 }  
                 next=r->link;  
                 if(t[d]==NULL)  
                 {  
                         b[d]=t[d]=r;  
                 }  
                 else  
                 {  
                         t[d]->link=r;  
                         t[d]=r;  
                 }  
                 r->link=NULL;  
                 r=next;  
         }  
         while(b[p]==NULL)  
         {  
                 p=p+1;  
         }  
         first=b[p];  
          for(l=p+1;l<=9;l++)  
          {  
                  prev=t[l-1];  
                  if(t[l]!=NULL)  
                  {  
                          prev->link=b[l];  
                  }  
                  else  
                  {  
                          t[l]=prev;  
                  }  
          }  
     }  
     cout<<"\nsorted data is";  
     display(first);  
}