//Insertion sort

#include<stdio.h>

insertionSort(int\* arr,int n)

{

int temp,i,j;

for(i=1;i<n;i++)

{

temp=arr[i];

j=i-1;

while(j>=0 && arr[j]>temp)

{

arr[j+1]=arr[j];

j--;

}

arr[j+1]=temp;

}

}

void main()

{

int arr[100],n,i;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

insertionSort(arr,n);

printf("After sorting:");

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

{

printf("%d\t",arr[i]);

}

}

//Bubble sort

#include<stdio.h>

#include<stdlib.h>

bubbleSort(int arr[],int n)

{

int temp,i,j,swap;

for(i=1;i<n;i++)

{

for(j=0;j<n-1;j++)

{

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

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

swap=1;

}

}

if(swap==0)

{

break;

}

}

}

void main()

{

int arr[100],n,i,\*ptr;

printf("Enter the number of elements:");

scanf("%d",&n);

ptr=(int\*) malloc(n\*sizeof(int));

printf("Enter the array elements:");

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

{

scanf("%d",(ptr+i));

}

bubbleSort(ptr,n);

printf("After sorting:");

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

{

printf("%d\t",\*(ptr+i));

}

}

//Selection sort

#include<stdio.h>

#include<stdlib.h>

selectionSort(int arr[],int n)

{

int temp,i,j;

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

{

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

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

}

void main()

{

int arr[100],n,i,\*ptr;

printf("Enter the number of elements:");

scanf("%d",&n);

ptr=(int\*) malloc(n\*sizeof(int));

printf("Enter the array elements:");

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

{

scanf("%d",(ptr+i));

}

selectionSort(ptr,n);

printf("After sorting:");

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

{

printf("%d\t",\*(ptr+i));

}

}

// Merge sort

#include<stdio.h>

#include<stdlib.h>

void mergeSort(int arr[],int lb,int ub)

{

int mid;

if(lb<ub) //atleast 2 elements

{

mid=(lb+ub)/2;

mergeSort(arr,lb,mid);

mergeSort(arr,mid+1,ub);

merge(arr,lb,mid,ub);

}

}

void merge(int arr[], int lb,int mid, int ub)

{

int i,j,k,b[50];

i=lb,j=mid+1,k=lb;

while(i<=mid && j<=ub)

{

if(arr[i]<=arr[j])

{

b[k]=arr[i];

i++;

k++;

}

else

{

b[k]=arr[j];

j++;

k++;

}

}

if(i>mid)

{

while(j<=ub)

{

b[k]=arr[j];

j++;

k++;

}

}

else if(j>ub)

{

while(i<=mid)

{

b[k]=arr[i];

i++;

k++;

}

}

for(k=lb;k<=ub;k++)

{

arr[k]=b[k];

}

}

void main()

{

int arr[100],n,i;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

int lb=0;

int ub=n-1;

//int mid=(0+(n-1))/2;

mergeSort(arr,lb,ub);

printf("After sorting:");

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

{

printf("%d\t",arr[i]);

}

}

//Quick sort

#include<stdio.h>

#include<stdlib.h>

quickSort(int arr[],int lb,int ub)

{

int loc;

if(lb<ub)

{

loc=partition(arr,lb,ub);

quickSort(arr,lb,loc-1);

quickSort(arr,loc+1,ub);

}

}

int partition(int arr[],int lb,int ub)

{

int temp;

int pivot=arr[lb]; //zero

int start=lb;

int end=ub;

while(start<end)

{

while(arr[start]<=pivot)

{

start++;

}

while(arr[end]>pivot)

{

end--;

}

if(start<end)

{

//swap(arr[start],arr[end]);

temp=arr[start];

arr[start]=arr[end];

arr[end]=temp;

}

}

//swap(arr[lb],arr[end]);

temp=arr[lb];

arr[lb]=arr[end];

arr[end]=temp;

return end;

}

void main()

{

int arr[100],n,i;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

int lb=0;

int ub=n-1;

//int mid=(0+(n-1))/2;

quickSort(arr,lb,ub);

printf("After sorting:");

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

{

printf("%d\t",arr[i]);

}

}

// Linked list

#include<stdio.h>

#include<stdlib.h>

struct node{

int data;

struct node \*next;

};

struct node\* createNode()

{

struct node\* newnode;

int data;

newnode=(struct node\*) malloc(sizeof(struct node));

if(newnode==NULL)

{

printf("Memory not allocated...");

return;

}

else

{

printf("\nEnter data:");

scanf("%d",&newnode->data);

newnode->next=NULL;

return newnode;

}

}

void createLinkedList(struct node\*\* head)

{

struct node\* newnode=NULL;

struct node\* travnode=\*head;

newnode=createNode();

if(\*head==NULL)

{

\*head=newnode;

}

else

{

while(travnode->next!=NULL)

{

travnode=travnode->next;

}

travnode->next=newnode;

}

}

void displayLinkedList(struct node\* head)

{

while(head!=NULL)

{

printf("%d ->",head->data);

head=head->next;

}

}

void insertAtFirst(struct node\*\* head)

{

struct node\* newnode=NULL;

newnode=createNode();

newnode->next=\*head;

\*head=newnode;

}

void insertAtPos(struct node\*\* head)

{

struct node\* newnode=NULL;

struct node\*travnode=\*head;

int pos,i;

printf("Enter position:");

scanf("%d",&pos);

int n=countNode(\*head);

if(pos==1)

{

insertAtFirst(head); //address of first

}

else if(pos==n+1)

{

createLinkedList(head);

}

else if(pos<1||pos>n+1)

{

printf("Invalid position...");

}

else

{

newnode=createNode();

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

{

travnode=travnode->next;

}

newnode->next=travnode->next;

travnode->next=newnode;

}

}

int countNode(struct node\* head)

{

int count=0;

while(head->next!=NULL)

{

count++;

head=head->next;

}

return count;

}

void deleteAtFirst(struct node\*\* head)

{

if(\*head==NULL)

{

printf("Linked list is not available...");

return;

}

else

{

struct node\* tempnode=\*head;

\*head=(\*head)->next;

free(tempnode);

}

}

void deleteAtLast(struct node\*\* head)

{

struct node\* travnode=\*head;

if(\*head==NULL)

{

printf("Linked list is not available...");

}

else if((\*head)->next==NULL) //If only one node

{

free(\*head);

\*head=NULL;

}

else //If more than 2 nodes present

{

while(travnode->next->next!=NULL)

{

travnode=travnode->next;

}

free(travnode->next);

travnode->next=NULL;

}

}

void deleteAtPos(struct node\*\* head)

{

int i,pos,n;

struct node\* travnode1=\*head;

struct node\* travnode2=NULL;

printf("Enter the position:");

scanf("%d",&pos);

n=countNode(\*head);

if(pos==1)

{

deleteAtFirst(head);

}

else if(pos==n)

{

deleteAtLast(head);

}

else if(pos<1||pos>n)

{

printf("Invalid position...");

}

else

{

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

{

travnode1=travnode1->next;

}

travnode2=travnode1->next;

travnode1->next=travnode2->next;

free(travnode2);

}

}

void main()

{

struct node\* firstnode;

int choice;

do{

printf("\n1. Create Linked List");

printf("\n2. Display Linked List");

printf("\n3. Insert at first");

printf("\n4. Insert at position");

printf("\n5. Insert at last");

printf("\n6. Delete at first");

printf("\n7. Delete at position");

printf("\n8. Delete at last");

printf("\n Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: createLinkedList(&firstnode);

break;

case 2: displayLinkedList(firstnode);

break;

case 3:insertAtFirst(&firstnode);

break;

case 4:insertAtPos(&firstnode);

break;

case 5:createLinkedList(&firstnode);

break;

case 6:deleteAtFirst(&firstnode);

break;

case 7:deleteAtPos(&firstnode);

break;

case 8:deleteAtLast(&firstnode);

break;

default:printf("Invalid choice...");

}

}while(choice!=0);

}

// Pallindrome element location

#include<stdio.h>

void main()

{

int arr[100],i,j,n,temp,rem,rev;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

printf("Pallindrome element position:");

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

{

temp=arr[i];

rev=0;

while(temp>0)

{

rem=temp%10;

rev=rev\*10+rem;

temp=temp/10;

}

if(arr[i]==rev)

{

printf("%d\t ",i);

}

}

}

//Half ascending half descending

#include<stdio.h>

void main()

{

int arr[100],i,j,temp,n;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

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

{

for(j=0;j<n/2;j++)

{

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

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

for(j=n/2;j<n-1;j++)

{

if(arr[j]<arr[j+1])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

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

{

printf("%d\t", arr[i]);

}

}

//Separate odd and even elements in same array

#include<stdio.h>

void main()

{

int arr[100],i,j,temp,n;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

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

{

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

{

if(arr[i]%2==0&&arr[j]%2!=0)

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

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

{

printf("%d\t",arr[i]);

}

}

//Print unique elements from array

#include<stdio.h>

void main()

{

int arr[100],brr[100],i,j,count=0,n,k=0;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

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

{

count=0;

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

{

if(arr[i]==arr[j])

{

count++;

}

}

if(count==1)

{

brr[k]=arr[i];

k++;

}

}

if(k>0)

{

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

{

printf("%d\t",brr[i]);

}

}

else

{

printf("No unique elements found");

}

}

//Delete element at given position

#include<stdio.h>

void main()

{

int arr[100],brr[100],i,j=0,k,count=0,n,pos;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

printf("Enter the position from where you want to delete element:");

scanf("%d",&pos);

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

{

if(i!=pos)

{

arr[k]=arr[i];

k++;

}

}

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

{

printf("%d\t",arr[j]);

}

}

//Second highest

#include<stdio.h>

void main()

{

int arr[100],brr[100],i,j,n,max,max2;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("Enter the array elements:");

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

{

scanf("%d",&arr[i]);

}

max=arr[0];

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

{

if(arr[i]>max)

{

max=arr[i];

}

}

printf("Largest element: %d",max);

max2=arr[0];

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

{

if(arr[i]>max2)

{

if(max!=arr[i])

{

max2=arr[i];

}

}

}

printf("\nSecond largest element: %d",max2);

}

// Addition of 2 matrics

#include<stdio.h>

void main()

{

int arr[100][100],brr[100][100],crr[100][100],i,j,r,c;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of first matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("Enter the array elements of second matrix:");

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

{

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

{

scanf("%d",&brr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

printf("\nMatrix 2:\n");

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

{

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

{

printf("%d\t",brr[i][j]);

}

printf("\n");

}

printf("\nAddition of 2 matrix: \n");

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

{

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

{

crr[i][j]=arr[i][j]+brr[i][j];

printf("%d\t",crr[i][j]);

}

printf("\n");

}

}

//Upper triangular matrix

#include<stdio.h>

void main()

{

int arr[100][100],i,j,r,c,flag=0;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

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

{

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

{

if(i>j && arr[i][j]!=0)

{

flag=1;

}

}

if(flag==1)

{

printf("Matrix is not upper traingular...");

}

}

if(flag==0)

{

printf("Matrix is upper traingular...");

}

}

//Lower triangular

#include<stdio.h>

void main()

{

int arr[100][100],i,j,r,c,flag=0;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

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

{

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

{

if(i<j && arr[i][j]!=0)

{

flag=1;

}

}

if(flag==1)

{

printf("Matrix is not lower traingular...");

}

}

if(flag==0)

{

printf("Matrix is lower traingular...");

}

}

// Matrix is identical or not

#include<stdio.h>

void main()

{

int arr[100][100],i,j,r,c,flag=0;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

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

{

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

{

if(i==j && arr[i][j]!=1)

{

flag=1;

break;

}

else if(i>j && arr[i][j]!=0)

{

flag=1;

break;

}

else if(j>i && arr[i][j]!=0)

{

flag=1;

break;

}

}

if(flag==1)

{

printf("Matrix is not identical...");

break;

}

}

if(flag==0)

{

printf("Matrix is identical...");

}

}

//Symmetric matrix

#include<stdio.h>

void main()

{

int arr[100][100],i,j,r,c,flag=0;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

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

{

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

{

if(arr[i][j]!=arr[j][i])

{

flag=1;

}

}

if(flag==1)

{

printf("Matrix is not symmetric...");

break;

}

}

if(flag==0)

{

printf("Matrix is symmetric...");

}

}

// Trnaspose of matrix

#include<stdio.h>

void main()

{

int arr[100][100],arr2[100][100],i,j,r,c,flag=0;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

printf("\nTranspose is:\n");

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

{

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

{

arr2[i][j]=arr[j][i];

printf("%d\t",arr2[i][j]);

}

printf("\n");

}

}

// Multiplication of two matrix

#include<stdio.h>

void main()

{

int arr[100][100],brr[100][100],crr[100][100],i,j,r,k,c;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of first matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("Enter the array elements of second matrix:");

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

{

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

{

scanf("%d",&brr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

printf("\nMatrix 2:\n");

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

{

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

{

printf("%d\t",brr[i][j]);

}

printf("\n");

}

printf("\nMultiplication of 2 matrix: \n");

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

{

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

{

crr[i][j]=0;

for(k=0;k<r;k++)

{

crr[i][j]+=arr[i][k]\*brr[k][j];

}

}

}

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

{

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

{

printf("%d\t",crr[i][j]);

}

printf("\n");

}

}

//Diagonal of matrix

#include<stdio.h>

void main()

{

int arr[100][100],brr[100][100],crr[100][100],i,j,r,k,c;

printf("Enter the number of rows:");

scanf("%d",&r);

printf("Enter the number of rows:");

scanf("%d",&c);

printf("Enter the array elements of first matrix:");

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

{

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

{

scanf("%d",&arr[i][j]);

}

}

printf("\nMatrix 1:\n");

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

{

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

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

printf("Diagonal of matrics:\n");

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

{

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

{

if(i==j)

{

printf("%d\t",arr[i][j]);

}

else

{

printf("\t");

}

}

printf("\n");

}

}

//Insert substring into mainstring

#include<stdio.h>

#include<string.h>

void main()

{

char str1[100],str2[100];

int i=0,pos;

printf("\nEnter the main string:");

fgets(str1,sizeof(str1),stdin);

printf("\nEnter the substring:");

fgets(str2,sizeof(str2),stdin);

printf("Enter position:");

scanf("%d",&pos);

int len1=strlen(str1);

int len2=strlen(str2);

for(i=len1-1;i>=pos;i--)

{

str1[i+len2]=str1[i];

}

for(i=0;i<len2-1;i++)

{

str1[pos+i]=str2[i];

}

printf("%s",str1);

}

// Array in reverse order

#include<stdio.h>

#include<string.h>

void main()

{

int arr[100],rev[100],i,j,n;

printf("Enter number of elements:");

scanf("%d",&n);

printf("Enter array elements:");

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

{

scanf("%d",&arr[i]);

}

j=0;

for(i=n-1;i>=0;i--)

{

rev[j]=arr[i];

j++;

}

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

{

printf("%d\t",rev[i]);

}

}

OR

#include<stdio.h>

#include<string.h>

void main()

{

int arr[100],rev[100],i,j,n;

printf("Enter number of elements:");

scanf("%d",&n);

printf("Enter array elements:");

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

{

scanf("%d",&arr[i]);

}

printf("\nArray is:");

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

{

printf("%d\t",arr[i]);

}

printf("\nArray in reverse order:");

for(i=n-1;i>=0;i--)

{

printf("%d\t",arr[i]);

}

}

// 2 Array merge and then sort

#include<stdio.h>

#include<string.h>

void main()

{

int arr[100],brr[100],i,j,n1,n2,temp;

printf("Enter number of elements:");

scanf("%d",&n1);

printf("Enter array 1 elements:");

for(i=0;i<n1;i++)

{

scanf("%d",&arr[i]);

}

printf("Enter number of elements:");

scanf("%d",&n2);

printf("Enter array 2 elements:");

for(i=0;i<n2;i++)

{

scanf("%d",&brr[i]);

}

j=0;

for(i=n1;i<n1+n2;i++)

{

arr[i]=brr[j];

j++;

}

printf("Array after merge:");

for(i=0;i<n1+n2;i++)

{

printf("%d\t",arr[i]);

}

for(i=0;i<n1+n2;i++)

{

for(j=i+1;j<n1+n2;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

printf("\nAfter sorting merged array:");

for(i=0;i<n1+n2;i++)

{

printf("%d\t",arr[i]);

}

}

// Equilibrium index find

#include<stdio.h>

#include<string.h>

void main()

{

int arr[100],brr[100],i,j,n,right,left;

printf("Enter number of elements:");

scanf("%d",&n);

printf("Enter array elements:");

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

{

scanf("%d",&arr[i]);

}

left=arr[0];

for(i=1;i<n;i++)

{

right=right+arr[i];

}

for(i=1;i<n;i++)

{

if(left==right)

{

printf("Equilibrium index is %d",i);

break;

}

else

{

left=left+arr[i+1];

right=right-arr[i+1];

}

}

}

// Missing number from 0 to n

#include<stdio.h>

#include<string.h>

void main()

{

int arr[100],brr[100],i,j,n,temp,missing;

printf("Enter number of elements:");

scanf("%d",&n);

printf("Enter array elements:");

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

{

scanf("%d",&arr[i]);

}

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

{

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

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

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

{

if(arr[i+1]-arr[i]>1)

{

missing=arr[i]+1;

break;

}

}

printf("Missing number is %d",missing);

}

// String is palindrome or not

#include<stdio.h>

#include<string.h>

void main()

{

char str[100];

int i,len,flag;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

while(str[i]!='\0')

{

i++;

len++;

}

for(i=0;i<len/2;i++)

{

if(str[i]!=str[len-1-i])

{

flag=1;

break;

}

}

if(flag==1)

{

printf("String is not pallindrome...");

}

else

{

printf("String is pallindrome...");

}

}

//Print last word

#include<stdio.h>

#include<string.h>

void main()

{

char str[100];

int i,len;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

len=strlen(str);

i=len-1;

while(i>=0 && str[i]!=' ')

{

i--;

}

printf("Last word: %s",str+i+1);

}

//Insert substring into main string

#include<stdio.h>

#include<string.h>

void main()

{

char str1[100],str2[100];

int i,len1,len2,pos;

printf("Enter the main string:");

fgets(str1,sizeof(str1),stdin);

printf("Enter the sub string:");

fgets(str2,sizeof(str2),stdin);

printf("Enter the position:");

scanf("%d",&pos);

len1=strlen(str1);

len2=strlen(str2);

//Creating space for substring

for(i=len1;i>=pos;i--)

{

str1[i+len2-1]=str1[i];

}

//insert new string

for(i=0;i<len2-1;i++)

{

str1[pos+i]=str2[i];

}

printf("After inserting substring: %s", str1);

}

//Tokenized the string and count no of words

#include<stdio.h>

#include<string.h>

void main()

{

char str1[100];

int i,len1,count=0;

printf("Enter the string:");

fgets(str1,sizeof(str1),stdin);

len1=strlen(str1);

char\* token=strtok(str1," ");

while(token!=NULL)

{

count++;

printf("%s\n",token);

token=strtok(NULL," ");

}

printf("No of tokens: %d", count);

}

//Is and a index

#include<stdio.h>

#include<string.h>

int main()

{

char str[100];

int i,len1,flag;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

len1=strlen(str);

str[strcspn(str,"\n")]='\0';

while(str[i]!='\0')

{

if(i==0)

{

while((str[i]=='i' || str[i]=='I') && (str[i+1]=='s' || str[i+1]=='S') && str[i+2]==' ' || str[i+1]=='\0')

{

printf("\nIs/is: %d",i);

flag=1;

break;

}

}

else

{

while(str[i-1]==' ' && (str[i]=='i' || str[i]=='I') && (str[i+1]=='s' || str[i+1]=='S') && str[i+2]==' ' || str[i+1]=='\0')

{

printf("\nIs/is: %d",i);

flag=1;

break;

}

}

if(i==0)

{

while((str[i]=='a' || str[i]=='A') && str[i+1]==' ' || str[i+1]=='\0')

{

printf("\nA/a: %d",i);

flag=1;

break;

}

}

else

{

while(str[i-1]==' ' && (str[i]=='a' || str[i]=='A') && str[i+1]==' ' || str[i+1]=='\0')

{

printf("\nA/a: %d",i);

flag=1;

break;

}

}

i++;

}

if(flag==0)

{

return -1;

}

}

OR

// IS and A index

#include <stdio.h>

#include<string.h>

int cal(char []);

int cal1(char []);

void main()

{

char str[100];

int index,index1;

printf("Enter string:");

fgets(str,sizeof(str),stdin);

index=cal(str);

printf("Is/Is: %d",index);

index1=cal1(str);

printf("\nA/a: %d",index1);

}

int cal(char str1[])

{

int i,flag=0;

while(str1[i]!='\0')

{

if(i==0)

{

if((str1[i]=='i' || str1[i]=='I') && (str1[i+1]=='s' || str1[i+1]=='S') && (str1[i+2]==' '|| str1[i+2]=='\0'))

{

flag=1;

break;

}

}

else

{

if((str1[i-1]==' ') && (str1[i]=='i' || str1[i]=='I') && (str1[i+1]=='s' || str1[i+1]=='S') && (str1[i+2]==' ' ||str1[i+2]=='\0'))

{

flag=1;

break;

}

}

i++;

}

if(flag==0)

{

return -1;

}

else{

return i;

}

}

int cal1(char str2[])

{

int len=strlen(str2);;

int i,flag=0;

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

{

if(i==0)

{

if((str2[i]=='a' || str2[i]=='A') && (str2[i+1]==' '|| str2[i+1]=='\0'))

{

flag=1;

break;

}

if((str2[i-1]==' ') && (str2[i]=='a' || str2[i]=='A') && (str2[i+1]==' '|| str2[i+1]=='\0'))

{

flag=1;

break;

}

}

if(flag==0)

{

return -1;

}

else{

return i;

}

}

}

// Strings are anagram or not

#include <stdio.h>

#include<string.h>

void main()

{

char str1[100],str2[100],temp;

int i,j,count;

printf("Enter first string:");

fgets(str1,sizeof(str1),stdin);

printf("Enter second string:");

fgets(str2,sizeof(str2),stdin);

str1[strcspn(str1,"\n")]='\0';

str2[strcspn(str2,"\n")]='\0';

int len1=strlen(str1);

int len2=strlen(str2);

if(len1==len2)

{

for(i=0;i<len1;i++)

{

for(j=i+1;j<len1;j++)

{

if(str1[i]>str1[j])

{

temp=str1[i];

str1[i]=str1[j];

str1[j]=temp;

}

}

}

for(i=0;i<len2;i++)

{

for(j=i+1;j<len2;j++)

{

if(str2[i]>str2[j])

{

temp=str2[i];

str2[i]=str2[j];

str2[j]=temp;

}

}

}

for(i=0;i<len1;i++)

{

if(str1[i]==str2[i])

{

count++;

}

}

if(count==len1)

{

printf("Strings are anagram...");

}

else

{

printf("Strings are not anagram...");

}

}

else

{

printf("Strings are not anagram...");

}

}

// No of partitions of upper case characters in left and right

#include <stdio.h>

#include<string.h>

void main()

{

char str[100],temp;

int i,j,k,par=0;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

int len1=strlen(str);

for(i=0;i<len1-2;i++)

{

int countl=0,countr=0;

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

{

if(str[k]>='A' && str[k]<='Z')

{

countl++;

}

}

for(j=i+1;j<len1;j++)

{

if(str[j]>='A'&& str[j]<='Z')

{

countr++;

}

}

if(countl>countr)

{

par++;

}

}

printf("No of partitions: %d",par);

}

// Stack using array

#include<stdio.h>

#define N 5

int stack[N];

int top=-1;

void push()

{

int x;

if(top==N-1)

{

printf("Stack is already full...");

}

else

{

printf("Enter the data:");

scanf("%d",&x);

top=top+1;

stack[top]=x;

}

}

void pop()

{

int item;

if(top==-1)

{

printf("Stack is empty...");

}

else

{

item=stack[top];

top--;

printf("Poped item is %d",item);

}

}

void peek()

{

if(top==-1)

{

printf("Stack is empty...");

}

else

{

printf("Peek item is %d",stack[top]);

}

}

void display()

{

int i;

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

{

printf("%d\n",stack[i]);

}

}

void main()

{

int choice;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stack using array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

printf("\n1. Push \n2. Pop \n3. Peek \n4. Display");

printf("\nEnter the choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: push();

break;

case 2: pop();

break;

case 3: peek();

break;

case 4: display();

break;

default:

printf("Invalid choice...");

}

}while(choice!=0);

}

// Queue using array

#include<stdio.h>

#define N 5

int queue[N];

int front=-1;

int rear=-1;

void enqueue()

{

int x;

printf("\nEnter the data:");

scanf("%d",&x);

if(rear==N-1)

{

printf("Queue is already full...");

}

else if(front=rear=-1) //if no any item present in queue

{

front=rear=0;

queue[rear]=x;

}

else //any item present in queue

{

rear++;

queue[rear]=x;

}

}

void dequeue()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else if(front==rear) //only one item

{

front=rear=-1;

}

else

{

printf("Deleted item is %d",queue[front]);

front++;

}

}

void peek()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else

{

printf("Peek item is %d",queue[front]);

}

}

void display()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else

{

int i;

for(i=front;i<=rear+1;i++)

{

printf("%d<--",queue[i]);

}

}

}

void main()

{

int choice;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Queue using array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

printf("\n1. Enqueue \n2. Dequeue \n3. Peek \n4. Display");

printf("\nEnter the choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: enqueue();

break;

case 2: dequeue();

break;

case 3: peek();

break;

case 4: display();

break;

default:

printf("Invalid choice...");

}

}while(choice!=0);

}

//Stack using linked list

#include <stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\* next;

};

struct node\* top=NULL;

void push()

{

int x;

struct node\* newnode=NULL;

newnode=(struct node\*)malloc(sizeof(struct node));

if(newnode==NULL)

{

printf("Memory not allocated...");

}

else

{

printf("Enter the data:");

scanf("%d",&x);

newnode->data=x;

newnode->next=top;

top=newnode;

}

}

void display()

{

struct node\* temp=top;

if(top==NULL)

{

printf("Stack is empty...");

}

else

{

while(temp!=NULL)

{

printf("%d\n",temp->data);

temp=temp->next;

}

}

}

void peek()

{

if(top==NULL)

{

printf("Stack is empty...");

}

else

{

printf("Peek item is %d",top->data);

}

}

void pop()

{

struct node\* temp=top;

if(top==NULL)

{

printf("Stack is empty...");

}

else

{

printf("Poped item is %d",top->data);

top=top->next;

free(temp);

}

}

int main(){

int choice;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stack using liked list\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do{

printf("\n1. Push\n2. Pop\n3. Peek\n4. Display");

printf("\nEnter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: push();

break;

case 2: pop();

break;

case 3: peek();

break;

case 4: display();

break;

default: printf("Invalid choice...");

}

}while(choice!=0);

return 0;

}

// Find the day of given date

#include<stdio.h>

void main()

{

int date,mon,year,i,s=0;

int month[12]={31,28,31,30,31,30,31,31,30,31,30,31};

char week[7][10]={"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"};

printf("Enter the date (dd/mm/yyyy):");

scanf("%d/%d/%d",&date,&mon,&year);

// Adjusting february days for leap years

if((year%4==0)&&(year%100!=0 || year%400==0))

{

month[1]=29;

}

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

{

s=s+month[i];

}

s=s+(date+year+(year/4)-2);

s=s%7;

printf("\nThe day is %s",week[s]);

}

// Even length word reverse

#include <stdio.h>

#include<string.h>

int main() {

char str[100];

int start,end,i;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

int len=strlen(str);

for(i=0;str[i]!='\0';i++)

{

if(str[i]!=' ')

{

start=i;

while(i<len-1 && str[i]!=' ')

{

i++;

}

end=i-1;

if((end-start+1)%2==0)

{

while(start<end)

{

char temp=str[start];

str[start]=str[end];

str[end]=temp;

start++;

end--;

}

}

}

}

printf("Modified string is %s",str);

}

//Queue using array

#include<stdio.h>

#define N 5

int queue[N];

int front=-1;

int rear=-1;

void enqueue()

{

int x;

printf("\nEnter the data:");

scanf("%d",&x);

if(rear==N-1)

{

printf("Queue is already full...");

}

else if(front==-1&&rear==-1) //if no any item present in queue

{

front=rear=0;

queue[rear]=x;

}

else //any item present in queue

{

rear++;

queue[rear]=x;

}

}

void dequeue()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else if(front==rear) //only one item

{

front=rear=-1;

}

else

{

printf("Deleted item is %d",queue[front]);

front++;

}

}

void peek()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else

{

printf("Peek item is %d",queue[front]);

}

}

void display()

{

if(front==-1&&rear==-1)

{

printf("Queue is empty...");

}

else

{

int i;

for(i=front;i<rear+1;i++)

{

printf("%d<--",queue[i]);

}

}

}

void main()

{

int choice;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Queue using array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

printf("\n1. Enqueue \n2. Dequeue \n3. Peek \n4. Display");

printf("\nEnter the choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: enqueue();

break;

case 2: dequeue();

break;

case 3: peek();

break;

case 4: display();

break;

default:

printf("Invalid choice...");

}

}while(choice!=0);

}

//Armstrong number

#include <stdio.h>

int main() {

int n,temp,count,rem,sum=0;

printf("Enter the number:");

scanf("%d",&n);

temp=n;

while(temp>0)

{

temp=temp/10;

count++;

}

temp=n;

while(temp>0)

{

rem=temp%10;

int ans=1;

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

{

ans=ans\*rem;

}

sum=sum+ans;

temp=temp/10;

}

if(n==sum)

{

printf("Armstrong...");

}

else

{

printf("Not armstrong...");

}

}

//Length of maximum palindrome substring

#include <stdio.h>

#include<string.h>

int logestpal(char\* str)

{

int len=strlen(str);

int start=0,end=1,i;

int low,high;

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

{

//check odd length pallindrome

low=i-1;

high=i;

while(low>=0 && high<len && str[low]==str[high])

{

if(high-low+1>end)

{

start=low;

end=high-low+1;

}

low--;

high++;

}

//check even length pallindrome

low=i-1;

high=i+1;

while(low>=0 && high<len && str[low]==str[high])

{

if(high-low+1>end)

{

start=low;

end=high-low+1;

}

low--;

high++;

}

}

return end;

}

int main() {

char str[100];

printf("Enter the string:::");

fgets(str,sizeof(str),stdin);

printf("Length of maximum pallindrome: %d",logestpal(str));

}

// Display the word at given location in sentence

#include <stdio.h>

#include<string.h>

void main()

{

char str[50];

int i,j,k=0,l=0,len,pos,count=0;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

len=strlen(str);

printf("Enter position:");

scanf("%d",&pos);

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

{

if(str[i]==' ')

{

count++;

if(count==pos-1){

k=i;

break;

}

}

}

for(j=k+1;str[j]!=' ';j++)

{

l=j;

}

for(i=k;i<=l;i++)

{

printf("%c",str[i]);

}

}

//strspan

#include<stdio.h>

#include<string.h>

int Stringspn(char\* str1, char\* str2)

{

int i=0,j,count=0;

while(str1[i]!='\0')

{

j=0;

while(str2[j]!='\0')

{

if(str1[i]==str2[j])

{

count++;

break;

}

j++;

}

if(str2[j]=='\0')

{

break;

}

i++;

}

return count;

}

int main(){

char str1[50];

char str2[30];

int i,j,count=0;

printf("enter the string 1:");

fgets(str1,sizeof(str1),stdin);

printf("enter the string 2:");

fgets(str2,sizeof(str2),stdin);

int res=Stringspn(str1,str2);

printf("Count= %d",res);

return 0;

}

//Sqeeze

// Target character remove from main string

#include <stdio.h>

#include<string.h>

void squuze(char\* str, char target)

{

int len=strlen(str);

int i, j=0;

for(i=0;str[i]!='\0';i++)

{

if(str[i]!=target)

{

str[j]=str[i];

j++;

}

}

str[j]='\0';

}

int main() {

char str[100], target;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

printf("Enter target character:");

scanf("%c",&target);

squuze(str,target);

printf("String is: %s", str);

return 0;

}

//Remove duplicates from string

// Online C compiler to run C program online

#include <stdio.h>

#include<string.h>

char\* duplicate(char\* str)

{

int len=strlen(str);

int i, j=0,count=0,k;

for(i=0;str[i]!='\0';i++)

{

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

{

if(str[i]==str[j])

break;

}

if(j==i)

str[k++]=str[i];

}

str[k]='\0';

return str;

}

int main() {

char str[100], target;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

printf("%s", duplicate(str));

return 0;

}

// Last word

#include <stdio.h>

#include<string.h>

int main() {

char str[100], target;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

int i=strlen(str);

while(i>=0 && str[i]!=' ')

{

i--;

}

printf("Last word: %s", str+i+1);

return 0;

}

//Reverse even length word from string

#include <stdio.h>

#include<string.h>

int main() {

char str[100], start,end;

printf("Enter the string:");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

int i,len1=strlen(str);

for(i=0;i<len1;i++)

{

if(str[i]!=' ')

{

start=i;

while(i<len1 && str[i]!=' ')

{

i++;

}

end=i-1;

if((end-start+1)%2==0)

{

while(start<end)

{

char temp=str[start];

str[start]=str[end];

str[end]=temp;

start++;

end--;

}

}

}

}

printf("Modified string: %s",str);

}

// Date comparison

#include <stdio.h>

struct Date{

int day;

int month;

int year;

};

int compareDates(struct Date date1, struct Date date2)

{

if(date1.year<date2.year)

{

return -1;

}

else if(date1.year>date2.year)

{

return 1;

}

else{

if(date1.month<date2.month)

{

return -1;

}

else if(date1.month>date2.month)

{

return 1;

}

else

{

if(date1.day<date2.day)

{

return -1;

}

else if(date1.day>date2.day)

{

return 1;

}

else

{

return 0;

}

}

}

}

int main()

{

int n,res[n],i;

printf("Enter the number of number:");

scanf("%d",&n);

struct Date date1,date2;

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

{

printf("Enter the first date (day month year):");

scanf("%d%d%d",&date1.day,&date1.month,&date1.year);

printf("Enter the second date (day month year):");

scanf("%d%d%d",&date2.day,&date2.month,&date2.year);

res[i]=compareDates(date1,date2);

}

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

{

if(res[i]>0)

{

printf("Date 1 is later than Date 2");

}

else if(res[i]<0)

{

printf("\n Date 1 is earlier than Date 2");

}

else

{

printf("Date 1 is equal to Date 2");

}

}

return 0;

}