BASICS:

FIBONACCI :

#include <iostream>

using namespace std;

int main()

{

int T;

cin>>T;

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

{

int n1=0,n2=1,n3=0,n;

cin>>n;

cout<<n1<<" "<<n2;

n=n-2;

while(n!=0)

{

n3 = n1 + n2;

n1 = n2;

n2 = n3;

cout<<" "<<n3;

n--;

}//while close

cout<<"\n";

}//for close

return 0;

}//main close

X PATTERN

// Program to make an X shape $ pattern in c++

#include <iostream>

#include<string.h>

using namespace std;

int main()

{

// n denotes the number of lines in which

// we want to make X pattern

char str[50];

cin>>str;

int n = strlen(str);

// i denotes the number of rows

if(n%2==0)

{

cout<<"INVALID";

return 0;

}

else{

// Print all rows one by one

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

{

int j = n -1 - i;

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

{

if (k == i || k == j)

cout << str[k];

else

cout << " ";

}

cout << endl;

}

} //else clsoe

}

POSITION OF A MAN:

import java.util.\*;

public class Main{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int position[]={0,0};

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

{

if(i%5==0)

{

position[0]=position[0]+10;

}

else if(i%5==1)

{

position[1]=position[1]+20;

}

else if(i%5==2)

{

position[0]=position[0]-30;

}

else if(i%5==3)

{

position[1]=position[1]-40;

}

else if(i%5 == 4)

{

position[0]=position[0]+50;

}

}

System.out.println(position[0]+" "+position[1]);

}

}

RIGHT ARROW:

#include<stdio.h>

int main() {

int r, c, rows, r1, c1, rows1;

scanf("%d", &rows1);

for(r1=1; r1<rows1; r1++){

for(c1=1; c1<=(2\*r1-2); c1++){

printf(" ");

}

for(c1=r1; c1<=rows1; c1++){

printf("\*");

}

printf("\n");

}

for(r1=1; r1<=rows1; r1++){

for(c1=1; c1<=(2\*rows1 - 2\*r1); c1++){

printf(" ");

}

for(c1=1; c1<=r1; c1++){

printf("\*");

}

printf("\n");

}

return 0;

}

INVERTED FULL PYRAMID:

#include <stdio.h>

int main() {

int n,i,j;

scanf("%d",&n);

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

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

printf(" ");

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

printf("\* ");

printf("\n");

}

return 0;

}

HALLOW DIAMOND:

#include <iostream>

using namespace std;

int main()

{

int i,j,rows;

cin>>rows;

for(i=1; i<=rows; i++){

for(j=rows; j>i; j--){

cout<<" ";

}

cout<<"\*";

for(j=1; j<(i-1)\*2; j++){

cout<<" ";

}

if(i==1){

cout<<"\n";

}

else{

cout<<"\*"<<"\n";

}

}

for(i=rows-1; i>=1; i--){

for(j=rows; j>i; j--){

cout<<" ";

}

cout<<"\*";

for(j=1; j<(i-1)\*2; j++){

cout<<" ";

}

if(i==1){

cout<<"\n";

}

else{

cout<<"\*";

cout<<"\n";

}

}

return 0;

}

REVERSE OF A STRING

def rev(s):

if(len(s)<=3):

return s

else:

rs=""

rs=rs+s[0]

index = len(s)

for i in range(index-2,0,-1):

rs = rs+s[i]

rs+=s[len(s)-1]

return rs

n=int(input())

a=[]

for i in range(n):

a.append(str(input().strip()))

for i in range(0,len(a)):

b=a[i].split(" ")

for i in range(0,len(b)):

b[i]=rev(b[i])

print(' '.join(b),end=" ")

print()

LEFT ARROW PATTERN

#include<stdio.h>

int main() {

int r, c, rows;

scanf("%d", &rows);

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

{

for(c=1; c<=2\*rows-2\*r; c++)

{

printf("\*");

printf(" ");

}

printf("\n");

}

for(r=1; r<=rows; r++)

{

for(c=1; c<=2\*r; c++)

{

printf("\*");

printf(" ");

}

printf("\n");

}

}

MAX NUMBER IN AN ARRAY:

#include <iostream>

using namespace std;

int main() {

int m;;

cin>>m;//length of array

int a[m];

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

cin>>a[i];

}

int x = a[1];

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

if(x<= a[j])

x=a[j];

}

cout<<x;

return 0;

}

SUM OF ARRAY

#include <iostream>

using namespace std;

int main() {

int m;;

cin>>m;//length of array

int a[m];

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

cin>>a[i];

}

int x = 0;

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

x=x+a[j];

}

cout<<x;

return 0;

}

MERGE TWO SORTED ARRAYS

#include <bits/stdc++.h>

using namespace std;

int main()

{

//cout<<"First String \n";

int n1,n2,temp =0;

cin>> n1;

int a[10000];

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

cin>>a[i];

}//for close a

//cout<<"Second String \n";

cin>>n2;

int b[10000];

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

cin>>b[i];

}//for close b

int n3= n1+n2;

int arr[1000];

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

arr[i] = a[i];

}//for close

int k=n1;

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

arr[i+k]=b[i];

}

//sorting the array

int x = arr[1];

for(int j=0;j<n3;j++){

for(int h =j+1; h<n3;h++){

if(arr[j]>=arr[h]){

temp = arr[j];

arr[j]=arr[h];

arr[h]=temp;

}

}

}//for close brr

//cout<<"Sorted String"<<"\n";

for(int i = 0; i<n3;i++){

cout<<arr[i]<<" ";

}//for close a

}//main close

LARGEST PALINDROME:

import java.util.\*;

public class Main{

public static boolean check(String str)

{

int i=0,j=str.length()-1;

if(str.length()==0)

return false;

while(i<j)

{

if(str.charAt(i)!=str.charAt(j))

return false;

j--;

i++;

}

return true;

}

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

int len=sc.nextInt();

//

int flag=0;

String ans="";

String arr[]=new String[len];

sc.nextLine();

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

{

arr[i]=sc.next();

}

//System.out.print(arr[0]);

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

{

if(check(arr[i]))

{

flag=1;

if(arr[i].length()>ans.length())

ans=arr[i];

}

}

if(flag==0)

System.out.println("-1");

else

System.out.println(ans);

}

}

MIDDLE ELEMENT IN AN ARRAY:

#include<iostream>

using namespace std;

int main(){

int m,n;

int a[100][100];

//cout<<"Enetr rows and coloumns";

cin>>m>>n;

//cout<<"Enter elements";

if(m==n && m%2!=0){

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

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

cin>>a[i][j];

}

}

}

//cout<<"Mid element is";

int x = m/2;

cout<<a[x][x];

}

PRINTING DIAGRNOL OF 2D ARRAY

#include <bits/stdc++.h>

using namespace std;

void printDiagonal(int a[100][100], int n, int m)

{

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

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

// Condition for principal diagonal

if (i == j)

cout << a[i][j] << " ";

}

}

}

int main()

{

int n,m;

cin>>n>>m;

int a[100][100];

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

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

cin>>a[i][j];

}

}

printDiagonal(a, n,m);

return 0;

}

TRANSPOSE OF A MATRIX:

#include <iostream>

using namespace std;

int main() {

int a[10][10], transpose[10][10], row, column, i, j;

// cout << "Enter rows and columns of matrix: ";

cin >> row >> column;

// Storing matrix elements

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

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

cin >> a[i][j];

}

}

// Computing transpose of the matrix

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

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

transpose[j][i] = a[i][j];

}

// Printing the transpose

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

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

cout<< transpose[i][j]<< " ";

/\*if (j == row - 1)

cout << endl << endl;

\*/}

cout<<endl;

}

}//main close