**PROBLEM SOLVING USING ‘C’**

**1.3N+1 PROBLEM:**

**For a given N value it has to print the numbers from 3 to N which takes equal number of iterations for completing the cycle.**

**Solution:**

#include<stdio.h>

int main()

{

int n,count=0,i;

scanf("%d",&n);

int a[n+1];

for(i=3;i<=n;i++)

{

int j=i,count=0;

while(j!=1)

{

if(j%2==0)

j=j/2;

else

j=3\*j+1;

count++;

}

a[i]=count;

}

for(i=3;i<n;i++)

{

count=1;

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

{

if(a[j]==a[i] && a[j]!=0)

{

printf("%d ,",j);

a[j]=0;

count++;

}

}

if(count>1)

{

printf(" %d --> %d\n",i,a[i]);

}

}

}

**2.Find the max no of water units that can be stored between any two values of given array.**

**Solution:**

#include<stdio.h>

int main()

{

int i,j,n,max,sum=0,temp,le=0,mvf;

scanf("%d",&n);

int a[n];

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

{

scanf("%d",&temp);

sum=sum+temp;

a[i]=temp;

}

printf("%d sum",sum);

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

{

j=i+1;

mvf=0;

while(a[j]<a[i]&&j<n)

{

le=1;

if(a[j]<a[j+1] && j<n-1)

mvf=1;

j++;

}

if(le==1)

{

if(j==n)

j--;

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

max=a[i];

else

max=a[j];

while(i<j-1&&mvf==1)

{

i++;

a[i]=max;

}

}

le=0;

}

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

printf("%d ",a[i]);

int sum1=0;

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

sum1+=a[i];

printf("%d sum1",sum1);

printf("%d Units",(sum1-sum));

}

**3.Find the largest water container from the values given in the array and return the indices .**

**Solution:**

#include<stdio.h>

int main()

{

int max=0,n,v1,v2,ind,i;

scanf("%d",&n);

int a[n];

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

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

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

{

if(a[i-1]<=a[i])

v1=a[i-1];

else

v1=a[i];

if(a[i]<=a[i+1])

v2=a[i];

else

v2=a[i+1];

if(v1>max && v1>= v2)

{

max=v1;

ind=i-1;

}

else if(v2>max && v2>v1)

{

max=v2;

ind=i;

}

}

printf("MAX: %d POS: %d %d",max,ind,ind+1);

}

**4.Print the following pattern**

**1**

**3\*2**

**4\*5\*6**

**10\*9\*8\*7**

**11\*12\*13\*14\*15**

**Solution:**

#include <stdio.h>

int main()

{

int n;

scanf("%d",&n);

int j, c = 0;

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

{

if (i%2 != 0)

{

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

printf("%d\*",j);

printf("%d",j++);

printf("\n");

c = j;

}

else

{

c = c+i-1;

for (j=c; j>c-i+1; j--)

printf("%d\*",j);

printf("%d",j);

printf("\n");

}

}

}

**5.For given matrix find if any identical rows exists in it and print the no of times the row is repeated**

**Solution:**

#include<stdio.h>

int main()

{

int m,n,count,k,i,j;

scanf("%d%d",&m,&n);

int mat[m][n],idr[m],idrc[m],dnc[m];

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

{

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

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

idr[i]=0;

idrc[i]=0;

dnc[i]=0;

}

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

{

k=i+1;

while(k<m)

{

count=0;

j=0;

while(mat[i][j]==mat[k][j] && j<n &&dnc[i]!=1)

{

count++;

j++;

}

if(count==n)

{

idr[i]=1;

idrc[i]+=1;

dnc[k]=1;

}

k++;

}

}

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

{

if(idr[i]==1)

{

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

printf("%d ",mat[i][j]);

}

if(idrc[i]!=0)

{

printf("No of times repeated is %d \n",idrc[i]+1);

}

}

}

**6.DailPad Problem:**

**Solution:**

#include<stdio.h>

int main()

{

int a[10][4],i,count=97,j,m,n,ii,ji;

for(i=2;i<10;i++)

{

if(i==7 || i==9)

{ j=0;

while(j<4)

{

a[i][j]=count;

count++;

j++;

}

}

else

{

j=0;

while(j<3)

{

a[i][j]=count;

count++;

j++;

}

}

}

scanf("%d%d",&m,&n);

if(m==7 || m==9)

ii=4;

else

ii=3;

if(n==7 || n==9)

ji=4;

else

ji=3;

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

{

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

{

printf("%c%c ",a[m][i],a[n][j]);

}

printf("\n");

}

}

**7.For a given 3x3 matrix find the 2x2 matrix whose sum is greatest**

**Solution:**

#include<stdio.h>

int main()

{

int m,n,i,j,max=0,indi,indj,sum;

scanf("%d%d",&m,&n);

int mat[m][n];

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

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

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

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

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

{

sum=mat[i][j]+mat[i+1][j]+mat[i][j+1]+mat[i+1][j+1];

if(sum>max)

{

max=sum;

indi=i;

indj=j;

}

}

printf("%d %d\n%d %d",mat[indi][indj],mat[indi][indj+1],mat[indi+1][indj],mat[indi+1][indj+1]);

}

**8. For a given set of given integers print all the missing values in between the maximum and minimum value in the given set.**

**Solution:**

#include<stdio.h>

int main()

{

int n,max,min,i,c=1;

scanf("%d",&n);

int arr[n];

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

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

max=min=arr[0];

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

{

if(arr[i]>max)

max=arr[i];

if(arr[i]<min)

min=arr[i];

}

mergesort(arr, 0, n-1);

for(i=min+1;i<max;i++)

{

if(arr[c]!=i)

printf("%d ",i);

else{

c++;

}

}

}

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

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

L[i] = arr[l + i];

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

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

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergesort(int arr[], int l, int r)

{

if (l < r)

{

int m = l+(r-l)/2;

mergesort(arr, l, m);

mergesort(arr, m+1, r);

merge(arr, l, m, r);

}

}

**9.For a given integer N print the following series till Nth term**

**1 11 21 1112 3112............**

**Solution:**

#include<stdio.h>

int main()

{

int n,arr[999],arr2[10],len,i,ind,j;

arr[0]=1;

len=1;

scanf("%d",&n);

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

arr2[i]=0;

printf("1\t");

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

{

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

{

arr2[arr[j]]++;

}

ind=0;

len=0;

for(j=1;j<10;j++)

{

if(arr2[j]!=0)

{

arr[ind]=j;

arr[ind+1]=arr2[j];

ind+=2;

len+=2;

}

arr2[j]=0;

}

for(j=0;j<len;j+=2)

printf("%d%d",arr[j+1],arr[j]);

printf("\t");

}

}