***Sorting***

***Insertion Sort:***

#include<stdio.h>

int main()

{

int n,array[1000],i,d,t;

printf("Enter number of elements : ");

scanf("%d",&n);

printf("Enter integers : \n");

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

{

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

}

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

{

d=i;

while(d>0&&array[d]<array[d-1])

{

t=array[d];

array[d]=array[d-1];

array[d-1]=t;

d--;

}

}

printf("Sorted list : ");

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

{

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

}

return 0;

}

***OUTPUT:***

Enter number of elements : 5

Enter integers :

9 4 3 7 5

Sorted list : 3, 4, 5, 7, 9,

***Merge Sort:***

#include<stdio.h>

void Merge(int p,int q,int r);

void MergeSort(int p ,int r);

int arr[100];

int main()

{

int n;

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

scanf("%d",&n);

printf("Enter the number : \n");

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

{

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

}

MergeSort(0,n-1);

printf("Sorted List : ");

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

{

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

}

return 0;

}

void Merge(int p,int q,int r)

{

int n1 = q-p+1;

int n2 = r-q;

int left[100],right[100];

for(int i=0; i<n1; i++) left[i] = arr[p+i];

for(int i=0; i<n2; i++) right[i] = arr[q+i+1];

int i=0, j=0;

left[n1] = 100;

right[n2] = 100;

for(int k=p; k<=r; k++)

{

if(left[i]<=right[j])

{

arr[k] = left[i];

i++;

}

else

{

arr[k] = right[j];

j++;

}

}

}

void MergeSort(int p ,int r)

{

if(p<r)

{

int q = (p+r)/2;

MergeSort(p,q);

MergeSort(q+1,r);

Merge(p,q,r);

}

}

***OUTPUT:***

Enter the number of elements : 5

Enter the number :

6 7 9 1 5

Sorted List : 1, 5, 6, 7, 9,

***Quick Sort :***

#include <stdio.h>

void quicksort( int [10], int , int );

int main()

{

int arr[20],size,i;

printf("Enter size of the array: " );

scanf("%d",&size);

printf("Enter elements:\n");

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

{

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

}

quicksort(arr,0,size-1);

printf("Sorted elements: " );

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

{

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

}

return 0;

}

void quicksort( int arr[10], int first, int last)

{

int pivot,j,temp,i;

if (first<last)

{

pivot=first;

i=first;

j=last;

while (i<j)

{

while (arr[i]<=arr[pivot]&&i<last)

i++;

while (arr[j]>arr[pivot])

j--;

if (i<j)

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

temp=arr[pivot];

arr[pivot]=arr[j];

arr[j]=temp;

quicksort(arr,first,j-1);

quicksort(arr,j+1,last);

}

}

***OUTPUT:***

Enter size of the array: 5

Enter elements:

5 7 3 1 9

Sorted elements: 1, 3, 5, 7, 9,

***Counting Sort :***

#include<stdio.h>

int main()

{

int a[100],c[100],b[100],maxv,n,i;

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

scanf("%d",&n);

printf("Enter elements:\n");

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

{

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

}

maxv=a[0];

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

{

if(a[i]>maxv)

{

maxv=a[i];

}

}

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

{

c[i]=0;

}

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

{

c[a[i]]++;

}

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

{

c[i]=c[i-1]+c[i];

}

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

{

b[c[a[i]]]=a[i];

c[a[i]]--;

}

printf("After Counting Sort: ");

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

{

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

}

return 0;

}

***OUTPUT:***

Enter the elements number of array: 5

Enter elements:

6 9 7 1 3

After Counting Sort: 1, 3, 6, 7, 9,

***Radix Sort :***

#include<iostream>

#include<vector>

using namespace std;

int NumberOfDigit(int num)

{

int cnt = 0;

while(num!=0)

{

cnt++;

num/=10;

}

return cnt;

}

int main()

{

int n,arr[100], arr2[100], mx , factor;

cout<<"Enter the limitation of Array : ";

cin>>n;

vector<int>vectorArr[10];

mx = -1000;

cout<<"Enter elements :\n";

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

{

cin>>arr[i];

arr2[i] = arr[i];

if(mx<arr[i]) mx = arr[i];

}

factor = 10;

int cnt = NumberOfDigit(mx);

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

{

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

{

int digit = arr2[i]%10;

vectorArr[digit].push\_back(arr[i]);

}

int l = 0;

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

{

for(int j=0;j<vectorArr[i].size();j++)

{

arr[l] = vectorArr[i][j];

arr2[l] = arr[l]/factor;

l++;

}

vectorArr[i].clear();

}

factor\*=10;

}

cout<<"After Redix Sort : ";

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

{

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

}

}

***OUTPUT:***

Enter the limitation of Array : 5

Enter elements :

123 654 789 369 159

After Redix Sort : 123, 159, 369, 654, 789,

***Heap Sort :***

#include <stdio.h>

int main()

{

int heap[100],size,i,j,c,root,temp;

printf( "Enter the size of array : ");

scanf("%d", &size);

printf("Enter the elements :\n");

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

{

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

}

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

{

c=i;

while(c!=0)

{

root=(c-1)/2;

if(heap[root]<heap[c])

{

temp=heap[root];

heap[root]=heap[c];

heap[c]=temp;

}

c=root;

}

}

for (j=size-1; j>=0; j--)

{

temp=heap[0];

heap[0]=heap[j];

heap[j]=temp;

root=0;

do

{

c=2\*root+1;

if((heap[c]<heap[c+1])&&c<j-1)

{

c++;

}

if (heap[root]<heap[c]&&c<j)

{

temp=heap[root];

heap[root]=heap[c];

heap[c]=temp;

}

root=c;

}

while(c<j);

}

printf("The sorted array is : ");

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

{

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

}

return 0;

}

***OUTPUT:***

Enter the size of array : 5

Enter the elements :

9 7 5 2 3

The sorted array is : 2, 3, 5, 7, 9,

***Searching***

***Binary Search Tree :***

#include<stdio.h>

int main()

{

int arr[100],arrB[100],n,root=1,item,index,i;

for(int i=1; i<=100; i++)

{

arrB[i]=NULL;

}

printf("Enter the Element number : ");

scanf("%d",&n);

printf("Enter the elements :\n");

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

{

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

}

arrB[root]=arr[1];

for(int i=2; i<=n; i++)

{

root=1;

while(arrB[root]!=NULL)

{

if(arr[i]>arrB[root])

{

root=2\*root+1;

}

else if(arr[i]<arrB[root])

{

root=2\*root;

}

}

if(arrB[root]==NULL)

{

arrB[root]=arr[i];

}

index=root;

}

printf("Enter the search item : ");

scanf("%d",&item);

i=1;

while(i<=index)

{

if(item>arrB[i])

{

i=2\*i+1;

}

else if(item<arrB[i])

{

i=2\*i;

}

if(item==arrB[i])

{

printf("Found at position :%d",i);

break;

}

}

if(i>index+1)

{

printf("NOT FOUND");

}

return 0;

}

***OUTPUT:***

Enter the Element number : 5

Enter the elements :

3 5 17 12 19

Enter the search item : 19

Found at position :15

***Dynamic Programming***

***Longest Common Subsequence (LCS) :***

#include<stdio.h>

#include<string.h>

void lcs();

void print(int i,int j);

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

int i,j,len1,len2,c[100][100],countLCS=0;

int main()

{

printf("Enter 1st String : ");

gets(str1);

printf("Enter 2nd String : ");

gets(str2);

printf("The Longest Common Subsequence is : ");

lcs();

print(len1,len2);

printf("\nThe Length of the Subsequence is : %d",countLCS);

return 0;

}

void lcs()

{

len1=strlen(str1);

len2=strlen(str2);

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

{

c[i][0]=0;

}

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

{

c[0][j]=0;

}

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

{

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

{

if(str1[i-1]==str2[j-1])

{

c[i][j]=c[i-1][j-1]+1;

strC[i][j]='c';

}

else if(c[i-1][j]>=c[i][j-1])

{

c[i][j]=c[i-1][j];

strC[i][j]='u';

}

else

{

c[i][j]=c[i][j-1];

strC[i][j]='l';

}

}

}

}

void print(int i,int j)

{

if(i==0 || j==0)

{

return;

}

if(strC[i][j]=='c')

{

print(i-1,j-1);

printf("%c",str1[i-1]);

countLCS++;

}

else if(strC[i][j]=='u')

{

print(i-1,j);

}

else

{

print(i,j-1);

}

}

***OUTPUT:***

Enter 1st String : sabujpaul

Enter 2nd String : apaul

The Longest Common Subsequence is : apaul

The Length of the Subsequence is : 5

***Coin Change :***

#include<stdio.h>

int main()

{

int coin[100],n,a,arr[1000],temp;

printf("Enter the coin number : ");

scanf("%d",&n);

printf("Enter the coins : ");

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

{

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

}

printf("Enter the Amount : ");

scanf("%d",&a);

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

{

if(i==0)

{

arr[i]=1;

}

else

{

arr[i]=0;

}

}

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

{

for(int j=0;j<=a;j++)

{

if(arr[j]!=0)

{

arr[j+coin[i]]=arr[j]+arr[j+coin[i]];

}

}

}

printf("NUMBER OF WAY TO MAKE %d IS : %d",a,arr[a]);

return 0;

}

***OUTPUT:***

Enter the coin number : 3

Enter the coins : 2 4 6

Enter the Amount : 10

NUMBER OF WAY TO MAKE 10 IS : 5

***Dividing coins (Minimum Different):***

#include<stdio.h>

int main()

{

int arr[100],coin[100],sum=0,i,j,n,other[100],diff[100];

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

scanf("%d",&n);

printf("Enter the coins : ");

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

{

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

sum=sum+coin[i];

}

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

{

arr[i]=0;

}

arr[0]=1;

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

{

for(int j=sum; j>=0; j--)

{

if(arr[j]==1)

{

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

}

}

}

int k=0,no;

for(i=1; i<=sum/2; i++)

{

if(arr[i]==1)

{

other[k]=sum-i;

diff[k]=other[k]-i;

no=k++;

}

}

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

{

for(j=i+1; j<=no; j++)

{

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

{

int temp=diff[i];

diff[i]=diff[j];

diff[j]=temp;

}

}

}

printf("Minimum Different : %d",diff[0]);

return 0;

}

***OUTPUT:***

Enter the number of coin : 3

Enter the coins : 2 3 6

Minimum Different : 1

***Graph***

***Breadth First Search (BFS) :***

#include<iostream>

#include<vector>

#include<string.h>

using namespace std;

vector<int>ver,adj[100];

bool flag[200];

void bfs(int source)

{

int front=0, rear=0, u, v;

int queue[100];

flag[source] = true;

queue[rear] = source;

while(front<=rear)

{

u = queue[front];

cout<<u<<" ";

for(int i=0;i<adj[u].size();i++)

{

v = adj[u][i];

if(!flag[v])

{

flag[v] = true;

rear++;

queue[rear] = v;

}

}

front++;

}

cout<<endl;

}

int main()

{

int m, n, u, v, source;

while(true)

{

cout<<"Enter the number of vertex : ";

cin>>m;

memset(flag,false,sizeof(flag));

ver.clear();

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

{

cin>>u;

cin>>n;

ver.push\_back(u);

adj[u].clear();

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

{

cin>>v;

adj[u].push\_back(v);

}

}

cout<<"Enter the Source : ";

cin>>source;

bfs(source);

}

}

***OUTPUT:***

Enter the number of vertex : 6

1 3 2 3 4

2 2 1 5

3 2 1 4

4 4 1 3 5 6

5 3 2 4 6

6 2 4 5

Enter the Source : 3

3 1 4 2 5 6

***Depth First Search (DFS) :***

#include<iostream>

#include<vector>

#include<string.h>

using namespace std;

vector<int>ver,adj[100];

bool flag[200];

void dfs(int source)

{

int stack[100];

int top=0;

stack[top] = source;

flag[source] = true;

while(top>=0)

{

int u = stack[top];

bool isTaken = false;

for(int i=0;i<adj[u].size();i++)

{

int v = adj[u][i];

if(!flag[v])

{

flag[v] = true;

top++;

stack[top] = v;

isTaken = true;

break;

}

}

if(!isTaken)

{

cout<<stack[top]<<" ";

top--;

}

}

cout<<endl;

}

int main()

{

int m, n, u, v, source;

while(true)

{

cout<<"Enter the number of vertex : ";

cin>>m;

memset(flag,false,sizeof(flag));

ver.clear();

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

{

cin>>u;

cin>>n;

ver.push\_back(u);

adj[u].clear();

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

{

cin>>v;

adj[u].push\_back(v);

}

}

cout<<"Enter the Source : ";

cin>>source;

dfs(source);

}

}

***OUTPUT:***

Enter the number of vertex : 6

1 3 2 3 4

2 2 1 5

3 2 1 4

4 4 1 3 5 6

5 3 2 4 6

6 2 4 5

Enter the Source : 3

6 4 5 2 1 3