**1.Write a program to implement the following sorting techniques**

**• Bubble sort**

**• Selection Sort**

**• Insertion Sort**

Program:

#include <stdio.h>

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void bubbleSort(int arr[], int n)

{

int i, j;

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

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

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

swap(&arr[j], &arr[j+1]);

}

void printArray(int arr[], int size)

{

int i;

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

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

printf("\n");

}

void selectionSort(int a[], int n)

{

int i, j,pos;

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

{

pos=i;

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

if(a[pos]>a[j])

pos=j;

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

swap(&a[pos],&a[i-1]);

}

}

void insertionSort(int ar[], int mn)

{

int i, key, j;

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

{

key = ar[i];

j = i - 1;

while (j >= 0 && ar[j] > key)

{

ar[j + 1] = ar[j];

j = j - 1;

}

ar[j + 1] = key;

}

}

int main()

{

int arr[] = {40, 35, 105, 20, 10};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

printf("BUBBLE SORT: \n");

printArray(arr, n);

printf("--------------------------- \n");

int aa[] = {66, 59, 22, 0, 12};

int nn = sizeof(aa)/sizeof(aa[0]);

selectionSort(aa, nn);

printf("SELECTION SORT: \n");

printArray(aa, nn);

printf("--------------------------- \n");

printf("INSERTION SORT: \n");

int ar[] = { 69, 30, 100, 22, 15 };

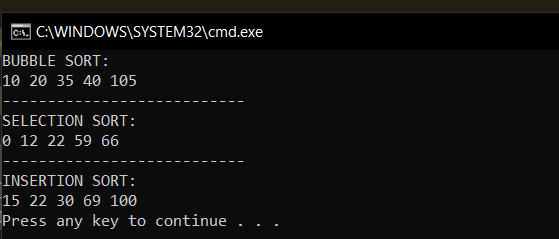
int mn = sizeof(ar) / sizeof(ar[0]);

insertionSort(ar, mn);

printArray(ar, mn);

return 0;

}



2. Write a program to implement heap sort

Program:

#include<stdio.h>

void create(int []);

void down\_adjust(int [],int);

int main()

{

int heap[30],n,i,last,temp;

printf("Enter no. of elements:");

scanf("%d",&n);

printf("\nEnter elements:");

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

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

heap[0]=n;

create(heap);

while(heap[0] > 1)

{

last=heap[0];

temp=heap[1];

heap[1]=heap[last];

heap[last]=temp;

heap[0]--;

down\_adjust(heap,1);

}

printf("\nArray after sorting:\n");

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

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

}

void create(int heap[])

{

int i,n;

n=heap[0]; //no. of elements

for(i=n/2;i>=1;i--)

down\_adjust(heap,i);

}

void down\_adjust(int heap[],int i)

{

int j,temp,n,flag=1;

n=heap[0];

while(2\*i<=n && flag==1)

{

j=2\*i; //j points to left child

if(j+1<=n && heap[j+1] > heap[j])

j=j+1;

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

flag=0;

else

{

temp=heap[i];

heap[i]=heap[j];

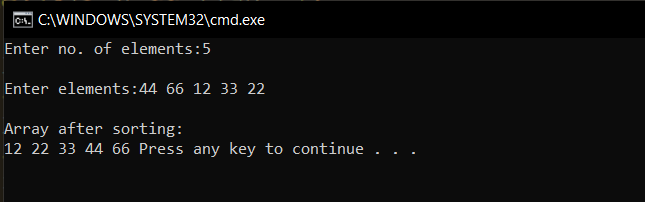
heap[j]=temp;

i=j;

}

}

}



3. Write a program to implement Shell sort

Program:

#include <stdio.h>

void shellsort(int arr[], int num)

{

int i, j, k, tmp;

for (i = num / 2; i > 0; i = i / 2)

{

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

{

for(k = j - i; k >= 0; k = k - i)

{

if (arr[k+i] >= arr[k])

break;

else

{

tmp = arr[k];

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

arr[k+i] = tmp;

}

}

}

}

}

int main()

{

int arr[30];

int k, num;

printf("Enter total no. of elements : ");

scanf("%d", &num);

printf("\nEnter %d numbers: ", num);

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

{

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

}

shellsort(arr, num);

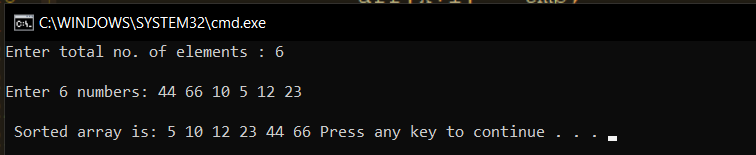
printf("\n Sorted array is: ");

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

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

return 0;

}



4. Write a program to implement Merge sort

Program:

#include <stdio.h>

#define max 10

int a[11] = { 200,123,105,999,569,666,345,261,975,461,241};

int b[10];

void merging(int low, int mid, int high)

{

int l1, l2, i;

for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++)

{

if(a[l1] <= a[l2])

b[i] = a[l1++];

else

b[i] = a[l2++];

}

while(l1 <= mid)

b[i++] = a[l1++];

while(l2 <= high)

b[i++] = a[l2++];

for(i = low; i <= high; i++)

a[i] = b[i];

}

void sort(int low, int high)

{

int mid;

if(low < high)

{

mid = (low + high) / 2;

sort(low, mid);

sort(mid+1, high);

merging(low, mid, high);

}

else

{

return;

}

}

int main()

{

int i;

printf("List before sorting\n");

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

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

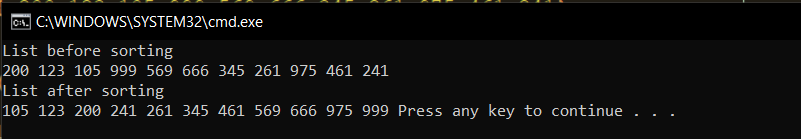
sort(0, max);

printf("\nList after sorting\n");

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

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

}



5. Write a program to implement Radix sort

Program:

#include <stdio.h>

int largest(int a[]);

void radix\_sort(int a[]);

void print(int a[], int n)

{

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

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

}

int main()

{

int a[10]={99,32,105,54,33,23,69,89,44,32};

printf("The array before sorting: \n");

print(a,10);

radix\_sort(a);

printf("\n\nThe sorted array is: \n");

print(a,10);

printf("\n\n");

}

int largest(int a[])

{

int larger=a[0], i;

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

{

if(a[i]>larger)

larger = a[i];

}

return larger;

}

void radix\_sort(int a[])

{

int bucket[10][10], bucket\_count[10];

int i, j, k, remainder, NOP=0, divisor=1, larger, pass;

larger = largest(a);

while(larger>0)

{

NOP++;

larger/=10;

}

for(pass=0;pass<NOP;pass++) // Initialize the buckets

{

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

bucket\_count[i]=0;

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

{

// sort the numbers according to the digit at passth place

remainder = (a[i]/divisor)%10;

bucket[remainder][bucket\_count[remainder]] = a[i];

bucket\_count[remainder] += 1;

}

// collect the numbers after PASS pass

i=0;

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

{

for(j=0;j<bucket\_count[k];j++)

{

a[i] = bucket[k][j];

i++;

}

}

divisor \*= 10;

}

}

