

DSA Lab programs

1. Write a program for the Insertion sort algorithm

Program:

```
#include <stdio.h>

int main()
{
    int n, array[1000], a, b, c, flag = 0;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (a = 0; a < n; a++)
        scanf("%d", &array[a]);

    for (a = 1 ; a <= n - 1; a++) {
        c = array[a];

        for (b = a - 1 ; b >= 0; b--) {
            if (array[b] > c) {
                array[b+1] = array[b];
                flag = 1;
            }
            else
                break;
        }
        if (flag)
```

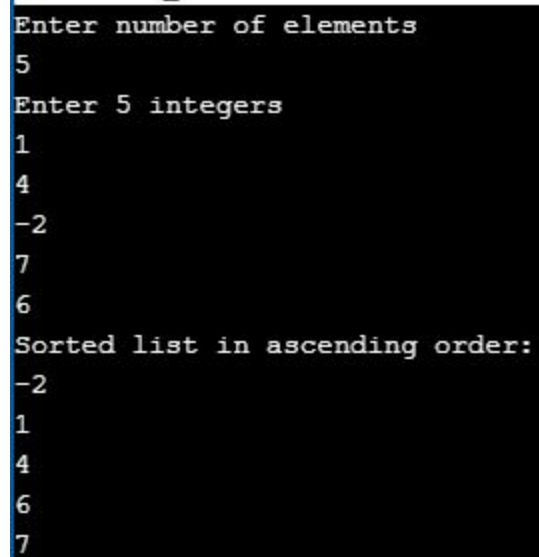
```
        array[b+1] = c;
    }

    printf("Sorted list in ascending order:\n");

    for (a = 0; a <= n - 1; a++) {
        printf("%d\n", array[a]);
    }

    return 0;
}
```

Output:



```
Enter number of elements
5
Enter 5 integers
1
4
-2
7
6
Sorted list in ascending order:
-2
1
4
6
7
```

2. Write a program for the Selection sort algorithm.

Program:

```
#include<stdio.h>
int main(){

    int i, j, count, temp, a[25];

    printf("enter the number of elements: ");
    scanf("%d",&count);

    printf("Enter %d elements: ", count);

    for(i=0;i<count;i++){
        scanf("%d",&a[i]);

    for(i=0;i<count;i++){
        for(j=i+1;j<count;j++){
            if(a[i]>a[j]){
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }

    printf("Sorted elements: ");
    for(i=0;i<count;i++)
        printf(" %d",a[i]);

    return 0;
}
```

Output:

```
enter the number of elements: 4
Enter 4 elements: 5
7
1
3
Sorted elements: 1 3 5 7
Program finished with exit code 0
```

3. Write a program for the Bubble sort algorithm.

program :

```
#include<stdio.h>
```

```
int main(){
```

```
    int count, temp, i, j, number[30];
```

```
    printf("How many numbers are u going to enter?: ");
    scanf("%d",&count);
```

```
    printf("Enter %d numbers: ",count);
```

```
    for(i=0;i<count;i++){
        scanf("%d",&number[i]);
```

```
    /* This is the main logic of bubble sort algorithm
    */
```

```
    for(i=count-2;i>=0;i--){
        for(j=0;j<=i;j++){
            if(number[j]>number[j+1]){
                temp=number[j];
```

```

        number[j]=number[j+1];
        number[j+1]=temp;
    }
}
}

```

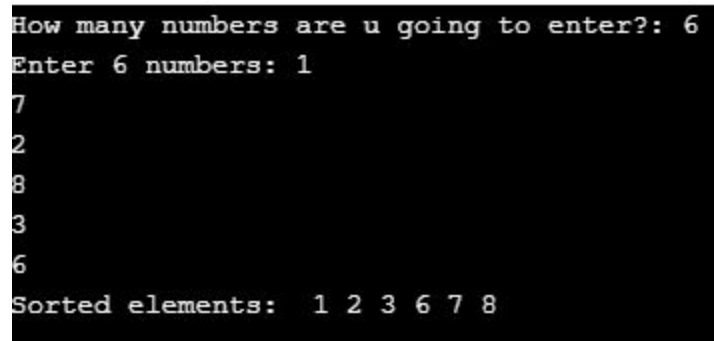
```

printf("Sorted elements: ");
for(i=0;i<count;i++)
    printf(" %d",number[i]);

return 0;

```

Output:



```

How many numbers are u going to enter?: 6
Enter 6 numbers: 1
7
2
8
3
6
Sorted elements:  1 2 3 6 7 8

```

4. Write a program for the Merge sort algorithm

Program:

```

#include <stdio.h>

// function to sort the subsection a[i .. j] of the array a[]
void merge_sort(int i, int j, int a[], int aux[]) {
    if (j <= i) {
        return;    // the subsection is empty or a single element
    }
    int mid = (i + j) / 2;

    // left sub-array is a[i .. mid]

```

```

// right sub-array is a[mid + 1 .. j]

merge_sort(i, mid, a, aux);    // sort the left sub-array recursively
merge_sort(mid + 1, j, a, aux);    // sort the right sub-array recursively

int pointer_left = i;    // pointer_left points to the beginning of the left
sub-array
int pointer_right = mid + 1;    // pointer_right points to the beginning of
the right sub-array
int k;    // k is the loop counter

// we loop from i to j to fill each element of the final merged array
for (k = i; k <= j; k++) {
    if (pointer_left == mid + 1) {    // left pointer has reached the limit
        aux[k] = a[pointer_right];
        pointer_right++;
    } else if (pointer_right == j + 1) {    // right pointer has reached the
limit
        aux[k] = a[pointer_left];
        pointer_left++;
    } else if (a[pointer_left] < a[pointer_right]) {    // pointer left points to
smaller element
        aux[k] = a[pointer_left];
        pointer_left++;
    } else {    // pointer right points to smaller element
        aux[k] = a[pointer_right];
        pointer_right++;
    }
}

for (k = i; k <= j; k++) {    // copy the elements from aux[] to a[]
    a[k] = aux[k];
}

```

```
}
```

```
int main() {  
    int a[100], aux[100], n, i, d, swap;  
  
    printf("Enter number of elements in the array:\n");  
    scanf("%d", &n);  
  
    printf("Enter %d integers\n", n);  
  
    for (i = 0; i < n; i++)  
        scanf("%d", &a[i]);  
  
    merge_sort(0, n - 1, a, aux);  
  
    printf("Printing the sorted array:\n");  
  
    for (i = 0; i < n; i++)  
        printf("%d\n", a[i]);  
  
    return 0;  
}
```

Output:

```
Enter number of elements in the array:
5
Enter 5 integers
4
8
9
6
2
Printing the sorted array:
2
4
6
8
9
```

5) Write a program for the Heapsort algorithm.

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]);
```

```
    //create a heap
```

```
    heap[0]=n;
```



```

create(heap);

//sorting
while(heap[0] > 1)
{
    //swap heap[1] and heap[last]
    last=heap[0];
    temp=heap[1];
    heap[1]=heap[last];
    heap[last]=temp;
    heap[0]--;
    down_adjust(heap,1);
}

//print sorted data
printf("\nArray after sorting:\n");
for(i=1;i<=n;i++)
    printf("%d ",heap[i]);

return 0;
}

```

```

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;
    }
}
}

```

Output:

```

Enter no. of elements:7

Enter elements:5
7
3
4
1
9
3

Array after sorting:
1 3 3 4 5 7 9

```

