# SANGOLA COLLEGE, SANGOLA
## Class-B.Sc(ECS)-II, SEM-IV 2024-25
## Practical Assignments
### Sub- Data Structure using C++-II

# Assignment No- 1

**1) Write a program to implement simple exchange sort method.**

```
#include<iostream.h>
#include<conio.h>
void simple();

void main()
{
   clrscr();
   simple();
   getch();
}

void simple()
{
   int i, j, t, n;
   int x[100];

   cout<<"\nHow many number you want to sort: ";
   cin>>n;

   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }

   for(i = 0; i <= n-1; i++)
   {
      for(j = i+1; j <= n-1; j++)
      {
         if(x[i] > x[j])
         {
            t = x[i];
            x[i] = x[j];
            x[j] = t;
```

```
        }
      }
    }

    cout<<"\nSorted array: ";
    for(i = 0; i <= n-1; i++)
    {
       cout<<"\t"<<x[i];
    }
}
```

o/p=>
How many numbers you want to sort: 5

Enter elements: 23 11 56 4 19

Sorted array:  4  11  19  23  56

## 2) Write a program to implement bubble sort method.

```
#include<iostream.h>
#include<conio.h>
void bubble();

void main()
{
   clrscr();
   bubble();
   getch();
}

void bubble()
{
   int i, j, t, n;
   int x[100];

   cout<<"\nHow many number you want to sort: ";
   cin>>n;

   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }
```

```
   for(i = 0; i <= n-1; i++)
   {
      for(j = 0; j <= n-2; j++)
      {
         if(x[j] > x[j+1])
         {
            t = x[j];
            x[j] = x[j+1];
            x[j+1] = t;
         }
      }
   }

   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
}
```

o/p=>
How many numbers you want to sort: 5

Enter elements: 12 7 5 19 2

Sorted array:  2  5  7  12  19

### 3) Write a program to implement quick sort method.

```
#include<iostream.h>
#include<conio.h>

int split(int[], int, int);
void quick(int[], int, int);

void main()
{
   clrscr();
   int x[50];
   int n, i;

   cout<<"\nHow many number you want to sort: ";
   cin>>n;
```

```
   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }

   quick(x, 0, n-1);
   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }

   getch();
}

void quick(int z[], int lw, int up)
{
   if(up > lw)
   {
      int i = split(z, lw, up);
      quick(z, lw, i-1);
      quick(z, i+1, up);
   }
}

int split(int z[], int lw, int up)
{
   int pivot, upper, lower, t;
   lower = lw;
   upper = up;
   pivot = z[lw];

   while(upper > lower)
   {
      while(z[lower] <= pivot)
      {
         lower++;
      }

      while(z[upper] > pivot)
      {
         upper--;
      }
```

```
      if(upper > lower)
      {
         t = z[lower];
         z[lower] = z[upper];
         z[upper] = t;
      }
   }
   t = z[lw];
   z[lw] = z[upper];
   z[upper] = t;
   return(upper);
}
```

o/p=>
How many numbers you want to sort: 5

Enter elements: 29 12 5 18 7

Sorted array:  5  7  12  18  29

## 4) Write a program to implement shell sort method.

```
#include<iostream.h>
#include<conio.h>
void shell();

void main()
{
   clrscr();
   shell();
   getch();
}

void shell()
{
   int i, j, n, t, gap;
   int x[50];
   cout<<"\nHow many number you want to sort: ";
   cin>>n;
   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }
```

```
   gap = n / 2;
   while(gap != 0)
   {
      for(i = gap; i < n; i++)
      {
         for(j = i-gap; j >= 0; j = j-gap)
         {
            if(a[gap+j] <= a[j])
            {
               t = a[j];
               a[j] = a[gap+j];
               a[gap+j] = t;
            }
         }
      }
      gap = gap / 2;
   }

   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
}
```

o/p=>
How many numbers you want to sort: 6

Enter elements: 32 17 5 9 42 1

Sorted array:  1  5  9  17  32  42

## 5) Write a program to implement selection sort method.

```
#include<iostream.h>
#include<conio.h>
void selection();

void main()
{
   clrscr();
   selection();
   getch();
}
```

```
void selection()
{
   int n, i, j, k, pos, min;
   int x[50];
   cout<<"\nHow many number you want to sort: ";
   cin>>n;
   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }

   for(i = 0; i <= n-1; i++)
   {
      min = x[i];
      for(j = i; j <= n-1; j++)
      {
         if(min > x[j])
         {
            min = x[j];
            pos = j;
         }
      }

      if(x[i] != min)
      {
         k = x[i];
         x[i] = x[pos];
         x[pos] = k;
      }
   }

   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
}
```

o/p=>
How many numbers you want to sort: 5

Enter elements: 15 3 9 1 20

Sorted array:  1  3  9  15  20

## 6) Write a program to implement insertion sort method.

```
#include<iostream.h>
#include<conio.h>
void insert();

void main()
{
   clrscr();
   insert();
   getch();
}

void insert()
{
   int i, j, k, n, t;
   int x[50];

   cout<<"\nHow many number you want to sort: ";
   cin>>n;

   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }

   for(i = 0; i < n; i++)
   {
      for(j = 0; j < i; j++)
      {
         if(x[i] < x[j])
         {
            t = x[j];
            x[j] = x[i];

            for(k = i; k > j; k--)
            {
               x[k] = x[k-1];
            }

            x[k+1] = t;
         }
      }
   }
```

```
   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
}
```

o/p=>
How many numbers you want to sort: 5

Enter elements: 23 4 12 7 1

Sorted array:  1  4  7  12  23

## 7) Write a program to implement merge sort method.

```
#include<iostream.h>
#include<conio.h>
void mergesort(int [], int, int);
void merge(int [], int, int, int, int);

void main()
{
   clrscr();
   int n, i;
   int x[50];

   cout<<"\nHow many number you want to sort: ";
   cin>>n;

   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }

   mergesort(x, 0, n-1);
   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
   getch();
}
```

```
void mergesort(int x[], int i, int j)
{
   int mid;
   if(i < j)
   {
      mid = (i + j) / 2;
      mergesort(x, i, mid);
      mergesort(x, mid+1, j);
      merge(x, i, mid, mid+1, j);
   }
}

void merge(int x[], int p, int q, int r, int s)
{
   int temp[50];
   int i, j, k;
   i = p;
   j = r;
   k = 0;

   while(i <= q && j <= s)
   {
      if(x[i] < x[j])
      {
         temp[k++] = x[i++];
      }
      else
      {
         temp[k++] = x[j++];
      }
   }

   while(i <= q)
   {
      temp[k++] = x[i++];
   }
      while(j <= s)
   {
      temp[k++] = x[j++];
   }
   for(i = p; i <= s; i++)
   {
      x[i] = temp[i-p];
   }
}
```

o/p=>
How many numbers you want to sort: 6

Enter elements: 44 12 5 89 1 32

Sorted array:  1  5  12  32  44  89

## 8) Write a program to implement heap sort method.

```
#include<iostream.h>
#include<conio.h>
void heapsort(int[], int);
void buildheap(int[], int);
void satisfyheap(int[], int, int);

void main()
{
   clrscr();
   int n, i;
   int x[50];
   cout<<"\nHow many number you want to sort: ";
   cin>>n;
   cout<<"\nEnter element: ";
   for(i = 0; i <= n-1; i++)
   {
      cin>>x[i];
   }
   heapsort(x, n);
   getch();
}

void heapsort(int x[], int n)
{
   buildheap(x, n);
   int i, t, m;
   m = n - 1;
   for(i = m; i >= 0; i--)
   {
      t = x[0];
      x[0] = x[m];
      x[m] = t;
      m--;
      satisfyheap(x, 0, m);
   }
```

```
   cout<<"\nSorted array: ";
   for(i = 0; i <= n-1; i++)
   {
      cout<<"\t"<<x[i];
   }
}
void buildheap(x[], int n)
{
   int  m = n - 1;

   for(int i = (n / 2); i >= 0; i--)
   {
      satisfyheap(x, i, m);
   }
}
void satisfyheap(int x[], int i, int m)
{
   int l, r, t, max;
   l = 2 * i;
   r = 2 * i + 1;
   if(l <= m && x[l] > x[i])
   {
      max = l;
   }
   else
   {
      max = i;
   }
   if(r <= m && x[r] > x[max])
   {
      max = r;
   }
   if(max != i)
   {
      t = x[i];
      x[i] = x[max];
      x[max] = t;
      satisfyheap(x, max, m);
   }
}
```

o/p=>

```
How many numbers you want to sort: 6
Enter elements: 45 12 88 23 9 1
Sorted array:    1    9    12    23    45    88
```

---