```
//OS-I Assignment1 SetB-1

  # include<stdio.h>
  # include <stdlib.h>
  # include<sys/types.h>
  # include<unistd.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 = 0; i < n-1; i++)

    // Last i elements are already in place
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}

/* Function to print an array */
void display(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        printf(" %d",arr[i]);
    printf("\n");
```

```
}

void insertionsort(int arr[], int n)
{
   int i, key, j;
   for (i = 1; i < n; i++)
   {
      key = arr[i];
      j = i - 1;

      /* Move elements of arr[0..i-1], that are
      greater than key, to one position ahead
      of their current position */
      while (j >= 0 && arr[j] > key)
      {
         arr[j + 1] = arr[j];
         j = j - 1;
      }
      arr[j + 1] = key;
   }
}

   int main()
   {
      int pid, child_pid;
      int size,i,status;
      int arr[size];
      int pArr[size];
      int cArr[size];

      /*    Input the Integers to be sorted    */
```

```
    printf("Enter the number of Integers to Sort:\t");
    scanf("%d",&size);




    for(i=0;i<size;i++)
    {
            printf("Enter number %d:",(i+1));
      scanf("%d",&arr[i]);
      pArr[i]=arr[i];
      cArr[i]=arr[i];
    }


/*    Display the Enterd Integers    */
    printf("Your Entered Integers for Sorting\n");
    display(arr,size);



/*    Process ID of the Parent    */
      pid=getpid();
      printf("\n Parent Process ID is : %d\n",pid);


/*    Child Process Creation    */
      printf("\n[ Forking Child Process ... ] \n");
      child_pid=fork();    /* This will Create Child Process and
                        Returns Child's PID */
    if( child_pid < 0){

      /* Process Creation Failed ... */
       printf("\nChild Process Creation Failed!!!!\n");
```

```
                exit(-1);
            }
            else if( child_pid==0)
           {
        /* Child Process */
            printf("\nThe Child Process\n");
            printf("\nChild process id is %d",getpid());
            printf("\nChild is sorting the list of Integers by INSERTION SORT:\n");
            insertionsort(cArr,size);
            printf("\nThe sorted List by Child::\n");
            display(cArr,size);
            printf("\nChild Process Completed ...\n");
            sleep(10);
                printf("\nparent of child process is %d",getppid());
            }


            else {
        /* Parent Process */
            printf("\nparent process %d started\n",getpid());
            sleep(30);
            printf("\nThe Parent Process\n");
            printf("\nParent %d is sorting the list of Integers by BUBBLE SORT\n",pid);
            bubblesort(pArr,size);
            printf("\nThe sorted List by Parent::\n");
            display(pArr,size);


               printf("\nParent Process Completed ...\n");
            }


      return 0;
    }
```