

```

#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void display(int arr[], int size)
{
    for (int i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

void bubblesort(int a[], int n)
{
    int i, j, temp;
    for (i = 0; i < n; i++)
        for (j = i + 1; j < n; j++)
            if (a[j] < a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
}

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

        while (j >= 0 && temp <= a[j])
        {
            a[j + 1] = a[j];
            j = j - 1;
        }
        a[j + 1] = temp;
    }
}

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

    printf("Enter no of elements to be sorted : ");
    scanf("%d", &size);

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

```

```

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

printf("Integers for sorting : ");
display(arr, size);

pid = getpid();
printf("\nParent process id : %d\n", pid);
printf("Forking child process\n");
child_pid = fork();

if (child_pid < 0)
{
    printf("Child process creation failed!\n");
    exit(-1);
}

if (child_pid == 0)
{
    printf("\nChild process %d started", child_pid);
    printf("\nChild is sorting the list of integers by insertion sort...");
    insertionsort(pArr, size);
    sleep(10);
    printf("\nThe sorted list by child : ");
    display(pArr, size);
    printf("\nParent of child process is %d\n", getppid());
    sleep(1);
}
else
{
    printf("\nParent process %d started", getpid());
    printf("\nParent process %d sorting the list of integers by bubble sort",
pid);
    bubblesort(cArr, size);
    sleep(10);
    printf("\nThe sorted list by parent : ");
    display(cArr, size);
    sleep(1);
}
}

```