**OS ASSIGNMENT 2B**

**Name:Mansi Mokashi**

**Rollno:87**

**TE IT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#include <stdio.h>

#include <sys/types.h>

#include <stdlib.h> #include <unistd.h>

void bubbleSort(int arr[], int n)

{ for (int i = 0; i < n; i++)

{ for (int j = 1; j < n - i; j++)

{ if (arr[j] < arr[j - 1])

{ int temp = arr[j]; arr[j] = arr[j - 1]; arr[j - 1] = temp;

}

}

} }

void display(int arr[], int n)

{ for (int i = 0; i < n; i++)

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

}

printf("\n");

}

int main(int argc, char \*argv[]) {

// taking the size of the array and storing it inside n variable int n; printf("Enter size of the array: "); scanf("%d", &n);

// creating an array of size n and taking the elements int integerArray[n];

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

{

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

}

char \*newenviron[] = {NULL};

pid\_t pid; pid = fork();

if (pid > 0) {

// inside parent process // sorting the array using bubble sort printf("inside parent process\n"); bubbleSort(integerArray, n); display(integerArray, n);

}

else if (pid == 0) {

// inside child process

// converting the interger array into character array

char \*temp = (char \*)malloc(sizeof(char) \* n); char \*temp[n];

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

{ temp[i] = (integerArray[i]);

}

char \*temp2[] = {NULL, "a", "b", NULL}; temp2[0] = argv[1];

printf("inside child process\n"); execve(argv[1], temp, newenviron);

} return 0;

}