**AIM : 2B.** Implement the C program in which main program accepts an array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an array and passes the sorted array to child process through the command line arguments of EXECVE system call. The child process uses EXECVE system call to load new program which display array in reverse order.

//LAB 2B – main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int compare(const void *a, const void *b) {
         return (*(int *)b - *(int *)a);
}

int main() {
        int n;
        printf("Enter the size of the array: ");
        scanf("%d", &n);

        int arr[n];
        printf("Enter %d elements of the array:\n", n);
        for (int i = 0; i < n; i++) {
                scanf("%d", &arr[i]);
        }

        pid_t child_pid = fork();

        if (child_pid == -1) {
                perror("Fork failed");
                exit(EXIT_FAILURE);
        }

        if (child_pid == 0) {
                // Child process
                char *args[n + 2];
                args[0] = "./display_array";
                args[n + 1] = NULL;

                // Convert integers to strings for execve arguments
                for (int i = 0; i < n; i++) {
```

```
                        char buffer[10];
                        snprintf(buffer, sizeof(buffer), "%d", arr[i]);
                        args[i + 1] = strdup(buffer);
                }

                execve(args[0], args, NULL);
                perror("Exec failed");
                exit(EXIT_FAILURE);
        } else {
                // Parent process
                wait(NULL);
                qsort(arr, n, sizeof(int), compare);

                printf("Sorted array in reverse order: ");
                for (int i = 0; i < n; i++) {
                        printf("%d ", arr[i]);
                }
                printf("\n");
        }

        return 0;
}
```

```c
//LAB 2B – sub.c

/*Program to print array received as commandline arguments*/

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
        printf("Sorted array received from parent in reverse order:\n");
            for (int i = 1; i < argc; i++) {
                    printf("%s ", argv[i]);
            }
        printf("\n");

        return 0;
}
```