**Name-Ujjawal Mandhani**

**Batch-F6**

**Enroll no-9918103237**

**OSSP LAB TEST-1**

**Question 1:**Write a Linux C program in which parent process generates the sequence of

series 3, 22, 7, 45, 15,91, 31……..The number of term of the series is inputted using

command line argument. After generation of series pass the whole array to child

process using inter process communication. Now the series is sorted in the child

process and the summation of series is passed to the parent process. Parent

process prints the summation.

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <sys/wait.h>**

**int main(int argc, char \*\*argv)**

**{**

**if (argc > 2)**

**{**

**printf("More than required arguments\n");**

**exit(0);**

**}**

**int n = atoi(argv[1]);**

**int fd[2];**

**int \*arr = (int \*)malloc(sizeof(int) \* n);**

**pipe(fd);**

**if (fork() > 0)**

**{**

**int i;**

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

**{**

**if (i == 0)**

**{**

**arr[i] = 3;**

**}**

**else if (i % 2 == 0)**

**{**

**arr[i] = arr[i - 2] \* arr[i - 2] + 1;**

**}**

**else**

**{**

**if (i == 1)**

**{**

**arr[i] = 19 + arr[i - 1];**

**}**

**else**

**arr[i] = (19 \* (i + 1) / 2) + arr[i - 1];**

**}**

**}**

**write(fd[1], arr, sizeof(int) \* n);**

**close(fd[1]);**

**wait(NULL);**

**int sum;**

**read(fd[0], &sum, sizeof(int));**

**close(fd[0]);**

**printf("Sum of the array = %d", sum);**

**}**

**else**

**{**

**sleep(n / 10);**

**int i, j, temp, sum = 0;**

**read(fd[0], arr, sizeof(int) \* n);**

**close(fd[0]);**

**printf("Original array: \n");**

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

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

**printf("\n");**

**for (i = 0; i < n - 1; i++)**

**for (j = 0; j < n - i - 1; j++)**

**{**

**if (arr[j] > arr[j + 1])**

**{**

**temp = arr[j];**

**arr[j] = arr[j + 1];**

**arr[j + 1] = temp;**

**}**

**}**

**printf("Sorted array: \n");**

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

**{**

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

**sum += arr[i];**

**}**

**printf("\n");**

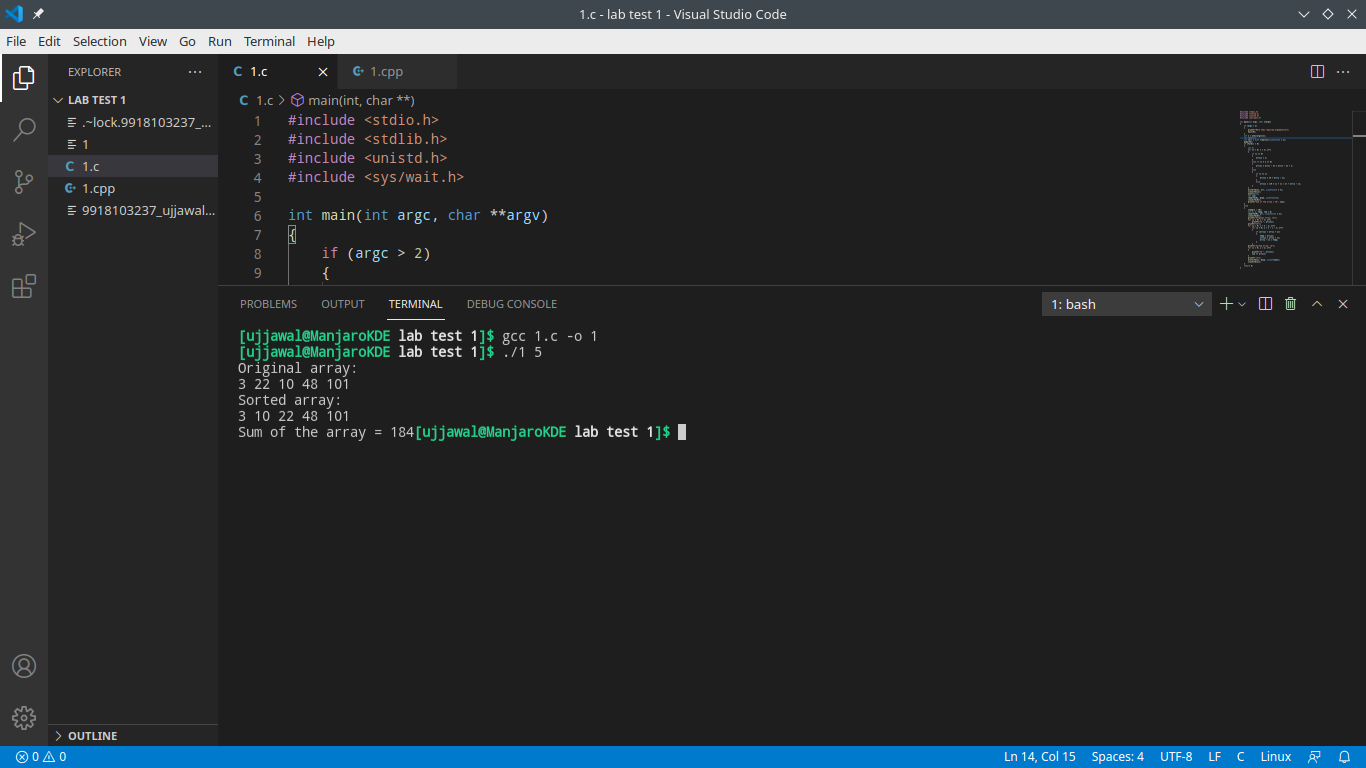
**write(fd[1], &sum, sizeof(sum));**

**close(fd[1]);**

**}**

**return 0;**

**}**



**Question 2:**Write a C Program to compute the average waiting time and average turnaround time if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may Increase.

**Process** **Arrival Time Burst Time**

P1 0.0 1

P2 0.4 4

P3 1.0 3

P4 1.0 3

**#include <iostream>**

**using namespace std;**

**double mat[10][6];**

**void swap(int \*a, int \*b)**

**{**

**double temp = \*a;**

**\*a = \*b;**

**\*b = temp;**

**}**

**void arrangeArrival(int num, double mat[][6])**

**{**

**for (int i = 0; i < num; i++)**

**{**

**for (int j = 0; j < num - i - 1; j++)**

**{**

**if (mat[j][1] > mat[j + 1][1])**

**{**

**for (int k = 0; k < 5; k++)**

**{**

**swap(mat[j][k], mat[j + 1][k]);**

**}**

**}**

**}**

**}**

**}**

**void completionTime(int num, double mat[][6])**

**{**

**double temp;**

**int val;**

**mat[0][3] = mat[0][1] + mat[0][2];**

**mat[0][5] = mat[0][3] - mat[0][1];**

**mat[0][4] = mat[0][5] - mat[0][2];**

**for (int i = 1; i < num; i++)**

**{**

**temp = mat[i - 1][3];**

**double low = mat[i][2];**

**for (int j = i; j < num; j++)**

**{**

**if (temp >= mat[j][1] && low >= mat[j][2])**

**{**

**low = mat[j][2];**

**val = j;**

**}**

**}**

**mat[val][3] = temp + mat[val][2];**

**mat[val][5] = mat[val][3] - mat[val][1];**

**mat[val][4] = mat[val][5] - mat[val][2];**

**for (int k = 0; k < 6; k++)**

**{**

**swap(mat[val][k], mat[i][k]);**

**}**

**}**

**}**

**int main()**

**{**

**int num, temp;**

**cout << "Enter number of Process: ";**

**cin >> num;**

**cout << "...Enter the process ID...\n";**

**for (int i = 0; i < num; i++)**

**{**

**cout << "...Process " << i + 1 << "...\n";**

**cout << "Enter Process Id: ";**

**cin >> mat[i][0];**

**cout << "Enter Arrival Time: ";**

**cin >> mat[i][1];**

**cout << "Enter Burst Time: ";**

**cin >> mat[i][2];**

**}**

**cout << "Before Arrange...\n";**

**cout << "Process ID\tArrival Time\tBurst Time\n";**

**for (int i = 0; i < num; i++)**

**{**

**cout << mat[i][0] << "\t\t" << mat[i][1] << "\t\t" << mat[i][2] << "\n";**

**}**

**arrangeArrival(num, mat);**

**completionTime(num, mat);**

**cout << "Final Result...\n";**

**cout << "Process ID\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n";**

**for (int i = 0; i < num; i++)**

**{**

**cout << mat[i][0] << "\t\t" << mat[i][1] << "\t\t" << mat[i][2] << "\t\t" << mat[i][4] << "\t\t" << mat[i][5] << "\n";**

**}**

**}**

