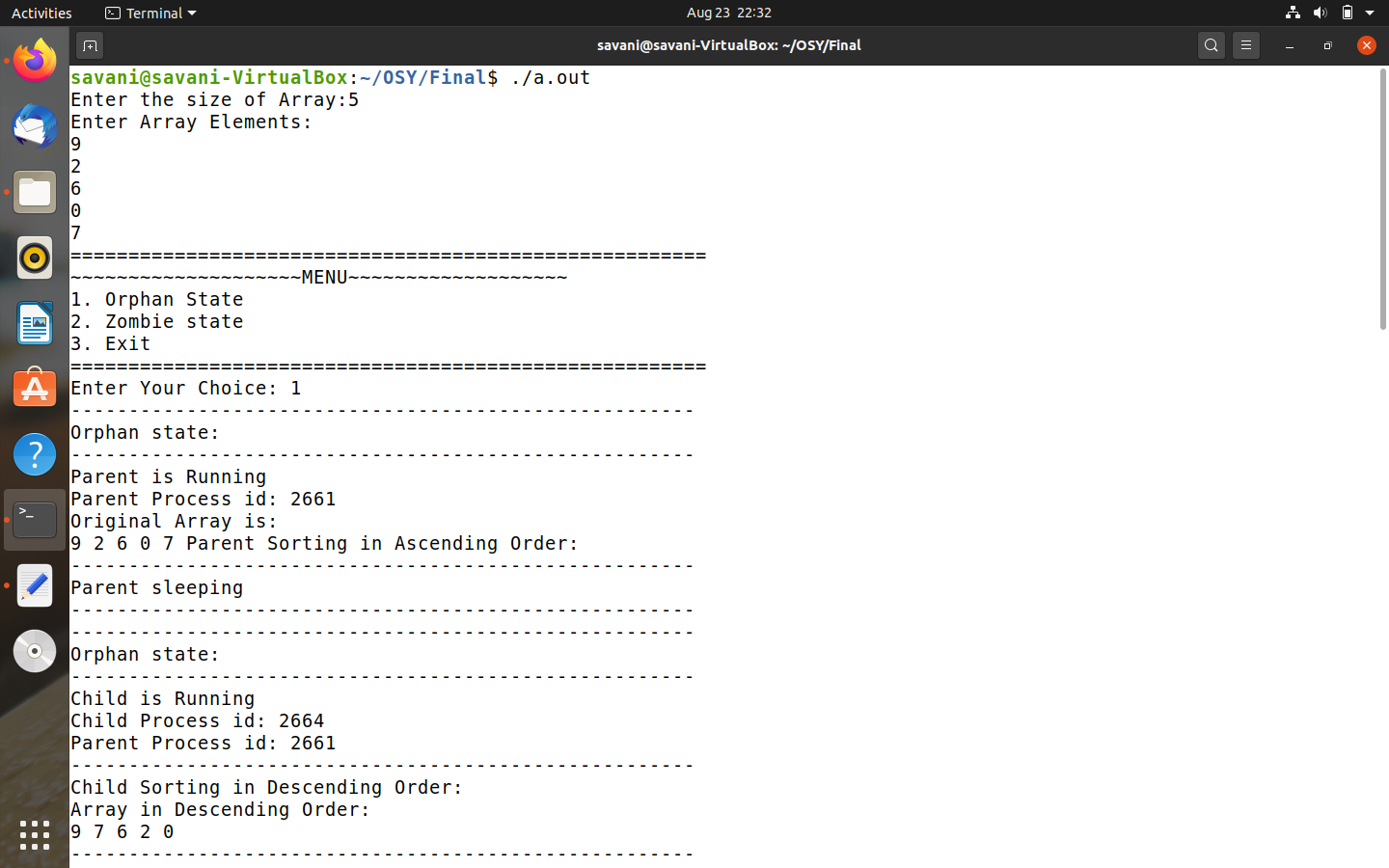
Name: Savani Pravin Kalekar

Roll No: 33320

**Assignment- 2A**

|  |  |  |
| --- | --- | --- |
| CODE:   * #include<stdio.h> #include<stdlib.h> #include<sys/types.h> #include<unistd.h> int main() { //int arr[]={4,1,7,3,2,9,6,5,8}; int i,j,n,arr[20],ch; printf("Enter the size of Array:"); scanf("%d",&n); printf("Enter Array Elements:\n"); for(i=0;i<n;i++) { scanf("%d",&arr[i]); }  do { printf("=======================================================\n"); printf("~~~~~~~~~~~~~~~~~~~~MENU~~~~~~~~~~~~~~~~~~~\n"); printf("1. Orphan State\n"); printf("2. Zombie state\n"); printf("3. Exit\n"); printf("=======================================================\n"); printf("Enter Your Choice: "); scanf("%d",&ch);  int cpid=fork();  switch(ch) { case 1: printf("------------------------------------------------------\n"); printf("Orphan state:\n"); printf("------------------------------------------------------\n"); if(cpid>0) { printf("Parent is Running\n"); printf("Parent Process id: %d \n",getpid());  printf("Original Array is:\n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); }  printf("Parent Sorting in Ascending Order:\n"); for (i = 0; i < n; i++)     {        for (j = 0; j < (n - i - 1); j++)        {            if (arr[j] > arr[j + 1])            {                int temp = arr[j];                arr[j] = arr[j + 1];                arr[j + 1] = temp;            }        } } printf("------------------------------------------------------\n"); printf("Parent sleeping\n"); printf("------------------------------------------------------\n"); sleep(2); printf("------------------------------------------------------\n"); printf("Parent Resumed\n"); printf("\nArray in Ascending Order: \n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); }  printf("\n------------------------------------------------------\n"); printf("\nParent Exiting\n"); } else if(cpid==0) { printf("Child is Running\n"); printf("Child Process id: %d\n",getpid()); printf("Parent Process id: %d\n", getppid()); printf("------------------------------------------------------\n"); printf("Child Sorting in Descending Order:\n"); for(i=0;i<n;i++) { for(j=i+1;j<n; j++) { if(arr[i]< arr[j]) { int t=arr[i]; arr[i]=arr[j]; arr[j]=t; } } } printf("Array in Descending Order:\n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); } printf("\n------------------------------------------------------\n");  printf("\nChild Sleeping\n"); sleep(10); printf("\n------------------------------------------------------\n"); printf("\nchild Resumed\n"); printf("Child Id: %d\n", getpid()); printf("Parent Id: %d\n",getppid()); printf("\nChild Exiting\n"); } break; case 2: printf("------------------------------------------------------\n"); printf("Zombie State:\n "); printf("------------------------------------------------------\n"); if(cpid>0) { printf("Parent is Running....\n"); printf("Id: %d\n",getpid());  printf("Original Array is:\n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); }  printf("Parent Sorting in Ascending Order:\n");  for (i = 0; i < n; i++)     {        for (j = 0; j < (n - i - 1); j++)        {            if (arr[j] > arr[j + 1])            {                int temp = arr[j];                arr[j] = arr[j + 1];                arr[j + 1] = temp;            }        } }  printf("Parent SLeeping....\n"); sleep(10); wait(NULL); printf("------------------------------------------------------\n"); printf("Parent Resumed\n"); printf("\nArray in Ascending Order: \n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); } printf("\n------------------------------------------------------\n"); wait(NULL); printf("Parent Exiting......\n"); exit(1); } else if(cpid==0) { printf("Child is Running....\n"); printf("Child Id: %d\n",getpid()); printf("Parent Id: %d\n",getppid());  printf("------------------------------------------------------\n"); printf("Child Sorting in Descending Order:\n"); for(i=0;i<n;i++) { for(j=i+1;j<n; j++) { if(arr[i]< arr[j]) { int t=arr[i]; arr[i]=arr[j]; arr[j]=t; } } } printf("Array in Descending Order:\n"); for(int i=0;i<n;i++) { printf("%d ",arr[i]); } printf("\n------------------------------------------------------\n");        printf("Child Sleeping:\n"); printf("Child Exiting....\n"); exit(0); } break; case 3: exit(0); } }while(ch!=3); } |  |  |

**OUTPUT:**

