Sriram R   **SRN : PES1UG20CS435**   **Section : H**

Code :

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

#define ARRAY_SIZE 8
#define BUFFER_SIZE 8 // should be greater than expected number of digits in sum
#define READ_END 0
#define WRITE_END 1

int main()
{
        int numArray[] = {1, 2, 3, 4, 5, 6, 7, 8};
        int sumChild = 0;
        int sum = 0;

        char write_msg[BUFFER_SIZE] = "";
        char read_msg[BUFFER_SIZE] = "";
        int fd[2];
        pid_t pid;

        if (pipe(fd) == -1)
        {
                printf("pipe failure\n");
                return 1;
        }

        pid = fork();

        if (pid > 0) // parent process - calculates sum of first half and stores in pipe
        {
                for(int i = 0; i < (ARRAY_SIZE / 2); i++) sum += numArray[i];
                snprintf(write_msg, BUFFER_SIZE, "%d", sum); // converts integer to string
(itoa not supported)
                printf(".... parent process is storing %s in pipe ....\n", write_msg);

                close(fd[READ_END]);
                write(fd[WRITE_END], write_msg, strlen(write_msg) + 1);
                close(fd[WRITE_END]);
                wait(NULL);
```

```
        }

        else if (pid == 0) // child process - calculates sum of second half, adds to the sum in
the pipe and prints
        {
                close(fd[WRITE_END]);
                read(fd[READ_END], read_msg, BUFFER_SIZE);

                for(int i = (ARRAY_SIZE / 2); i < ARRAY_SIZE; i++) sumChild +=
numArray[i];

                sumChild += atoi(read_msg);

                printf("child process summation : %d\n", sumChild);
                close(fd[READ_END]);
        }

        else
        {
                printf("fork failure\n");
                return 1;
        }

        return 0;
}
```

Output (window highlighted in red) :