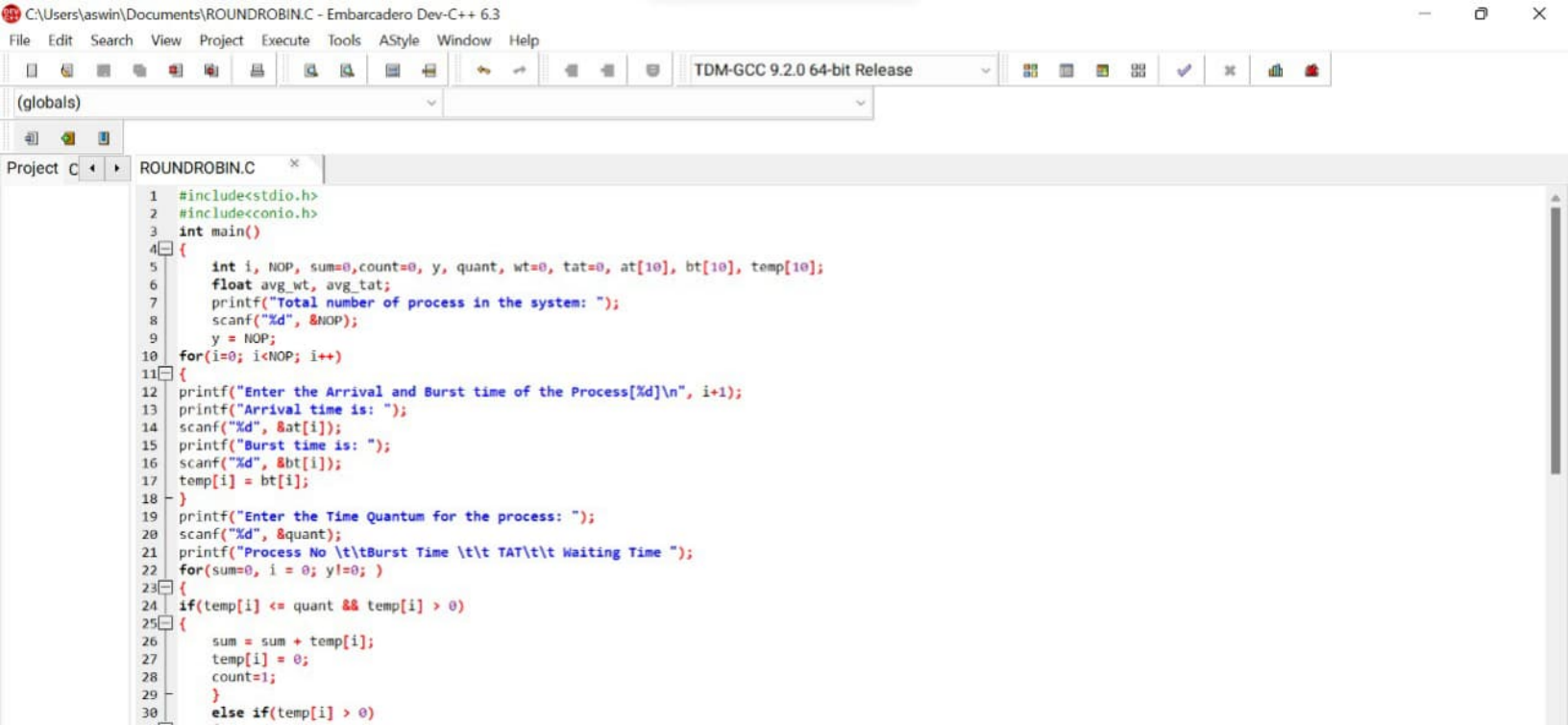

C:\Users\aswin\Documents\ROUNDROBIN.C - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C
ROUNDROBIN.C
#include<stdio.h>
#include<conio.h>
int main()
{
int i, NOP, sum=0,count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];
float avg_wt, avg_tat;
printf("Total number of process in the system: ");
scanf("%d", &NOP);
y = NOP;
for(i=0; i<NOP; i++)
{
printf("Enter the Arrival and Burst time of the Process[%d]\n", i+1);
printf("Arrival time is: ");
scanf("%d", &at[i]);
printf("Burst time is: ");
scanf("%d", &bt[i]);
temp[i] = bt[i];
}
printf("Enter the Time Quantum for the process: ");
scanf("%d", &quant);
printf("Process No \t\tBurst Time \t\t TAT\t\t Waiting Time ");
for(sum=0, i = 0; y!=0;)
{
if(temp[i] <= quant && temp[i] > 0)
{
sum = sum + temp[i];
temp[i] = 0;
count=1;
}
else if(temp[i] > 0)

```
    {
        temp[i] = temp[i] - quant;
        sum = sum + quant;
    }
    if(temp[i]==0 && count==1)
    {
        y--;
        printf("\nProcess No[%d] \t\t %d\t\t\t %d\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);
        wt = wt+sum-at[i]-bt[i];
        tat = tat+sum-at[i];
        count =0;
    }
    if(i==NOP-1)
    {
        i=0;
    }
    else if(at[i+1]<=sum)
    {
        i++;
    }
    else
    {
        i=0;
    }
}
avg_wt = wt * 1.0/NOP;
avg_tat = tat * 1.0/NOP;
printf("\nAverage Turn Around Time: %f", avg_wt);
printf("\nAverage Waiting Time: %f", avg_tat);
return 0;
}
```

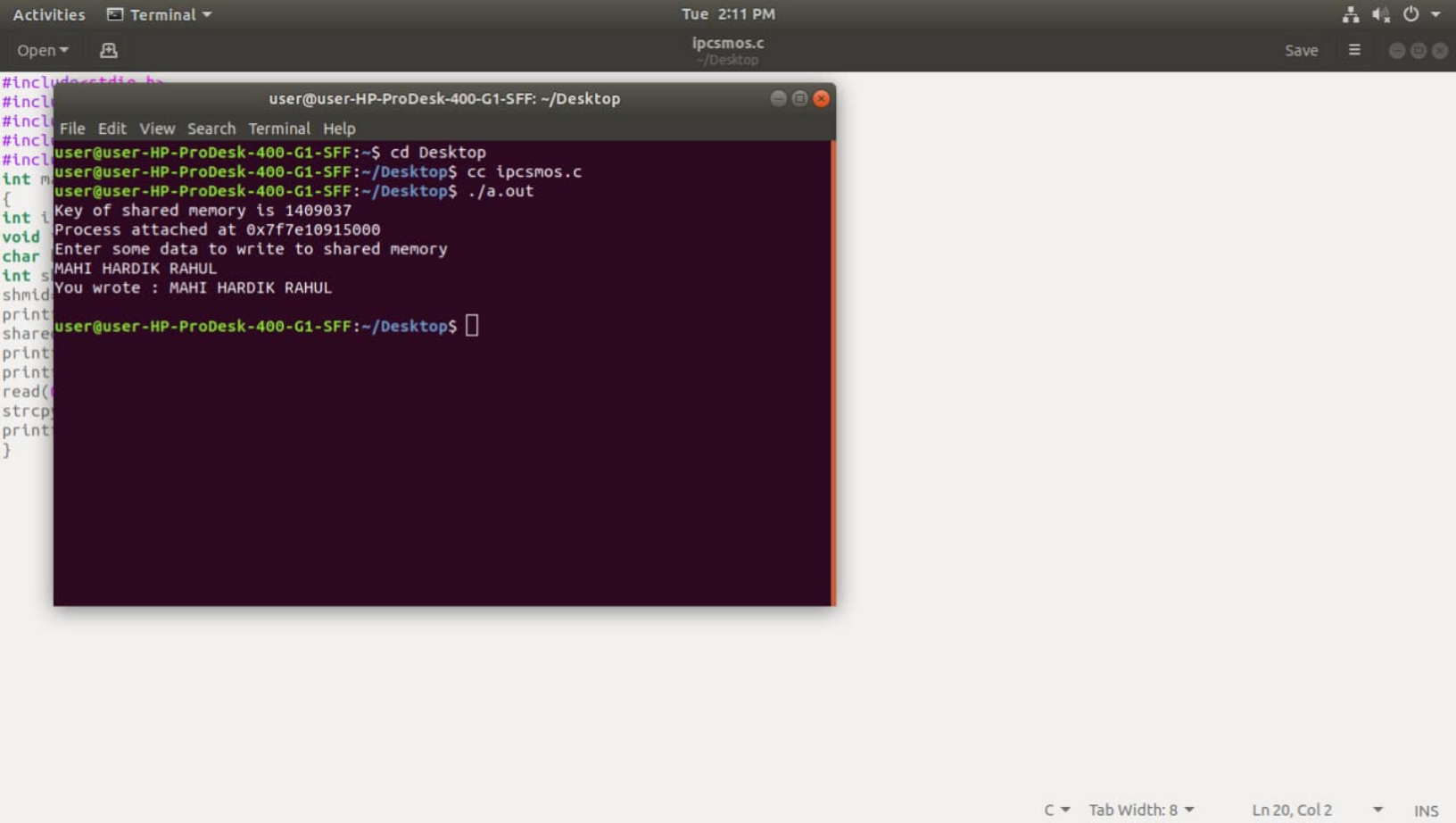
Compiler | Resources | Compile Log | Debug | Find Results | Console

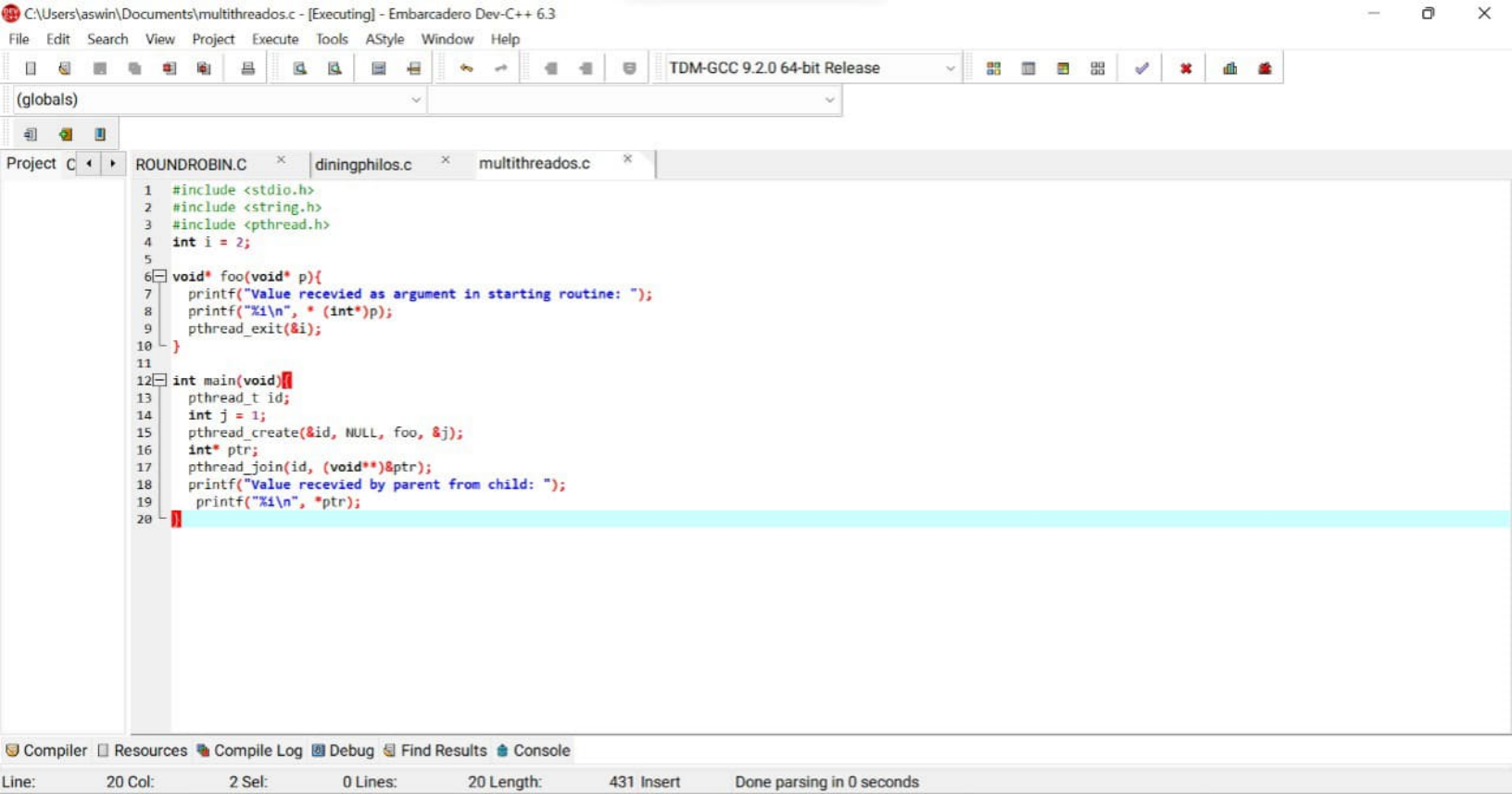
Line: 50 Col: 8 Sel: 0 Lines: 61 Length: 1530 Insert Done parsing in 0 seconds

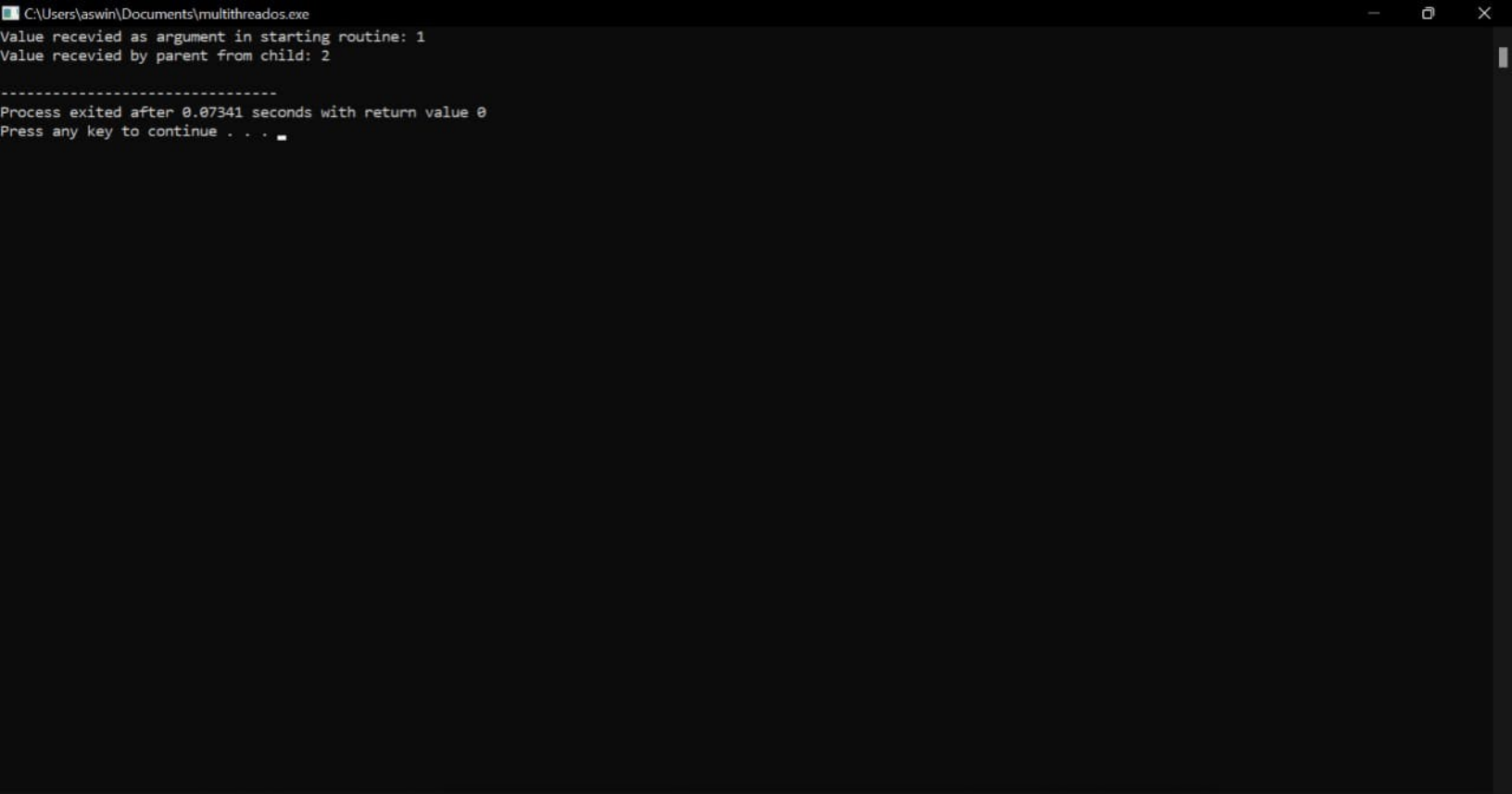
C:\Users\aswin\Documents\ROUNDROBIN.exe

```
Total number of process in the system: 4
Enter the Arrival and Burst time of the Process[1]
Arrival time is: 0
Burst time is: 23
Enter the Arrival and Burst time of the Process[2]
Arrival time is: 8
Burst time is: 31
Enter the Arrival and Burst time of the Process[3]
Arrival time is: 22
Burst time is: 16
Enter the Arrival and Burst time of the Process[4]
Arrival time is: 18
Burst time is: 19
Enter the Time Quantum for the process: 6
Process No              Burst Time              TAT             Waiting Time
Process No[1]            23                     47                      24
Process No[3]            16                     53                      37
Process No[4]            19                     70                      51
Process No[2]            31                     81                      50
Average Turn Around Time: 40.500000
Average Waiting Time: 62.750000
--------------------------------
Process exited after 30.07 seconds with return value 0
Press any key to continue . . .
```

```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
int i;
void *shared_memory;
char buff[100];
int shmid;
shmid=shmget((key_t)2345, 1024, 0666|IPC_CREAT);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Enter some data to write to shared memory\n");
read(0,buff,100);
strcpy(shared_memory,buff);
printf("You wrote : %s\n",(char *)shared_memory);
}
```


Activities
Terminal
Tue 2:11 PM
Open
ipcsmos.c
~/Desktop
Save
user@user-HP-ProDesk-400-G1-SFF: ~/Desktop
File Edit View Search Terminal Help
user@user-HP-ProDesk-400-G1-SFF:~$ cd Desktop
user@user-HP-ProDesk-400-G1-SFF:~/Desktop$ cc ipcsmos.c
user@user-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
Key of shared memory is 1409037
Process attached at 0x7f7e10915000
Enter some data to write to shared memory
MAHI HARDIK RAHUL
You wrote : MAHI HARDIK RAHUL
user@user-HP-ProDesk-400-G1-SFF:~/Desktop$
C
Tab Width: 8
Ln 20, Col 2
INS


C:\Users\aswin\Documents\multithreados.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C
ROUNDROBIN.C
diningphilos.c
multithreados.c
#include <stdio.h>
#include <string.h>
#include <pthread.h>
int i = 2;
void* foo(void* p){
printf("Value recevied as argument in starting routine: ");
printf("%i\n", * (int*)p);
pthread_exit(&i);
}
int main(void){
pthread_t id;
int j = 1;
pthread_create(&id, NULL, foo, &j);
int* ptr;
pthread_join(id, (void**)&ptr);
printf("Value recevied by parent from child: ");
printf("%i\n", *ptr);
}
Compiler Resources Compile Log Debug Find Results Console
Line: 20 Col: 2 Sel: 0 Lines: 20 Length: 431 Insert Done parsing in 0 seconds


C:\Users\aswin\Documents\multithreados.exe
Value recevied as argument in starting routine: 1
Value recevied by parent from child: 2

Process exited after 0.07341 seconds with return value 0
Press any key to continue . . .

C:\Users\aswin\Documents\diningphilos.c - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

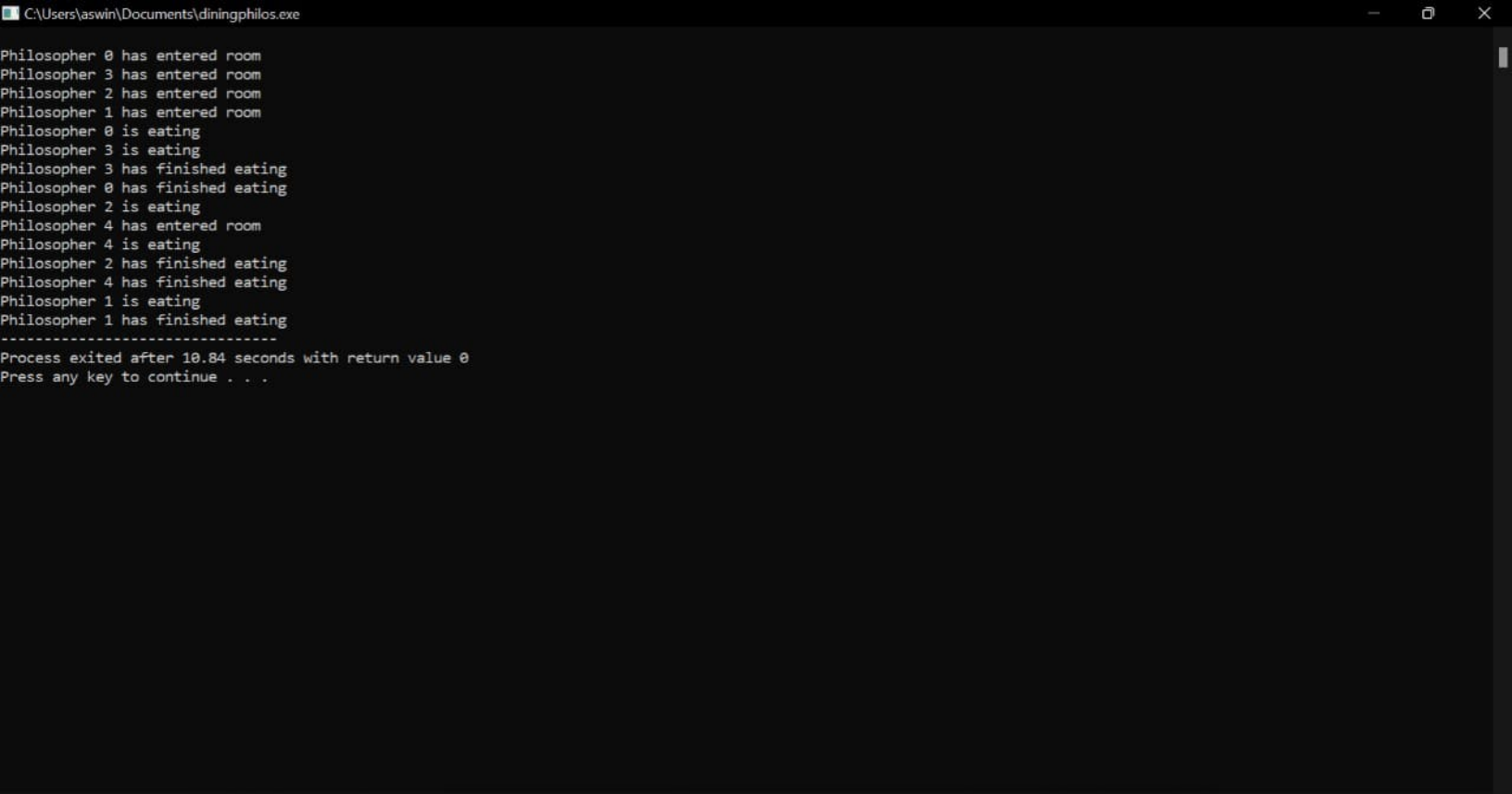
Project C

ROUNDROBIN.C | diningphilos.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>
sem_t room;
sem_t chopstick[5];
void * philosopher(void *);
void eat(int);
int main()
{
    int i,a[5];
    pthread_t tid[5];
    sem_init(&room,0,4);
    for(i=0;i<5;i++)
        sem_init(&chopstick[i],0,1);
    for(i=0;i<5;i++){
        a[i]=i;
        pthread_create(&tid[i],NULL,philosopher,(void *)&a[i]);
    }
    for(i=0;i<5;i++)
        pthread_join(tid[i],NULL);
}
void * philosopher(void * num)
{
    int phil=*(int *)num;
    sem_wait(&room);
    printf("\nPhilosopher %d has entered room",phil);
    sem_wait(&chopstick[phil]);
    sem_wait(&chopstick[(phil+1)%5]);
    eat(phil);
    sleep(2);
    printf("\nPhilosopher %d has finished eating",phil);
    sem_post(&chopstick[(phil+1)%5]);
    sem_post(&chopstick[phil]);
    sem_post(&room);
}
void eat(int phil)
{
    printf("\nPhilosopher %d is eating",phil);
}
```

Compiler | Resources | Compile Log | Debug | Find Results | Console

Line: 41 Col: 2 Sel: 0 Lines: 41 Length: 884 Insert Done parsing in 0.171 seconds


C:\Users\aswin\Documents\diningphilos.exe
Philosopher 0 has entered room
Philosopher 3 has entered room
Philosopher 2 has entered room
Philosopher 1 has entered room
Philosopher 0 is eating
Philosopher 3 is eating
Philosopher 3 has finished eating
Philosopher 0 has finished eating
Philosopher 2 is eating
Philosopher 4 has entered room
Philosopher 4 is eating
Philosopher 2 has finished eating
Philosopher 4 has finished eating
Philosopher 1 is eating
Philosopher 1 has finished eating

Process exited after 10.84 seconds with return value 0
Press any key to continue . . .

C:\Users\aswin\Documents\firstfitos.c - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C

ROUNDROBIN.C | diningphilos.c | multithreados.c | firstfitos.c

```c
#include<stdio.h>
void main()
{
    int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j;
    for(i = 0; i < 10; i++)
    {
        flags[i] = 0;
        allocation[i] = -1;
    }
    printf("Enter no. of blocks: ");
    scanf("%d", &bno);
    printf("Enter size of each block: ");
    for(i = 0; i < bno; i++)
        scanf("%d", &bsize[i]);
    printf("Enter no. of processes: ");
    scanf("%d", &pno);
    printf("Enter size of each process: ");
    for(i = 0; i < pno; i++)
        scanf("%d", &psize[i]);
    for(i = 0; i < pno; i++)
        for(j = 0; j < bno; j++)
            if(flags[j] == 0 && bsize[j] >= psize[i])
            {
                allocation[j] = i;
                flags[j] = 1;
                break;
            }
    printf("\nBlock no.\tsize\t\tprocess no.\t\tsize");
    for(i = 0; i < bno; i++)
    {
        printf("\n%d\t\t%d\t\t", i+1, bsize[i]);
        if(flags[i] == 1)
            printf("%d\t\t\t%d",allocation[i]+1,psize[allocation[i]]);
        else
            printf("Not allocated");
    }
}
```

Compiler | Resources | Compile Log | Debug | Find Results | Console

Line: 15 Col: 40 Sel: 0 Lines: 37 Length: 917 Insert Done parsing in 0 seconds

C:\Users\aswin\Documents\firstfitos.exe

```
Enter no. of blocks: 4
Enter size of each block: 6
8
3
2
Enter no. of processes: 5
Enter size of each process: 8
5
3
9
1

Block no.       size            process no.             size
1               6               2                       5
2               8               1                       8
3               3               3                       3
4               2               5                       1
--------------------------------
Process exited after 13.49 seconds with return value 4
Press any key to continue . . .
```