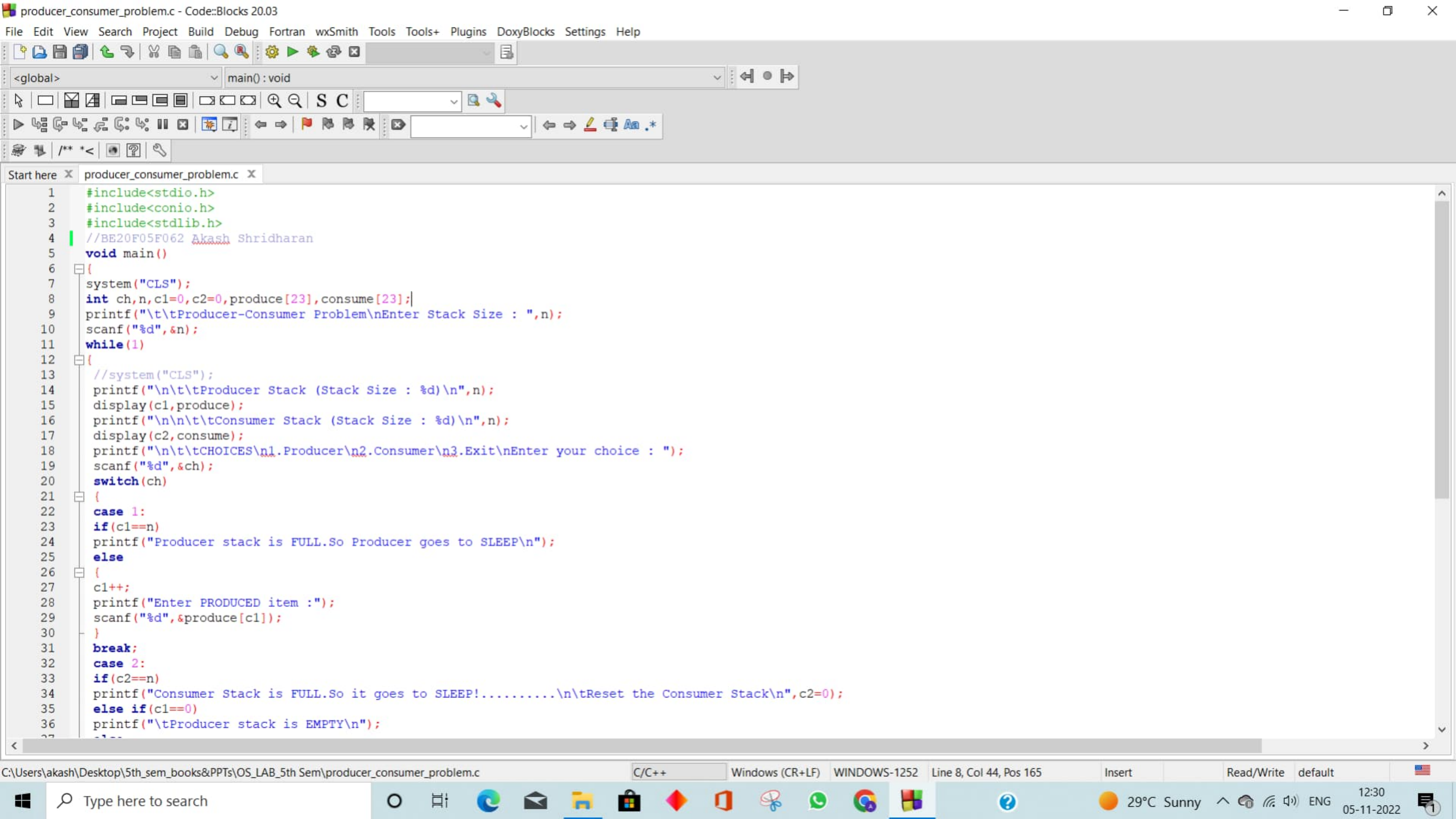
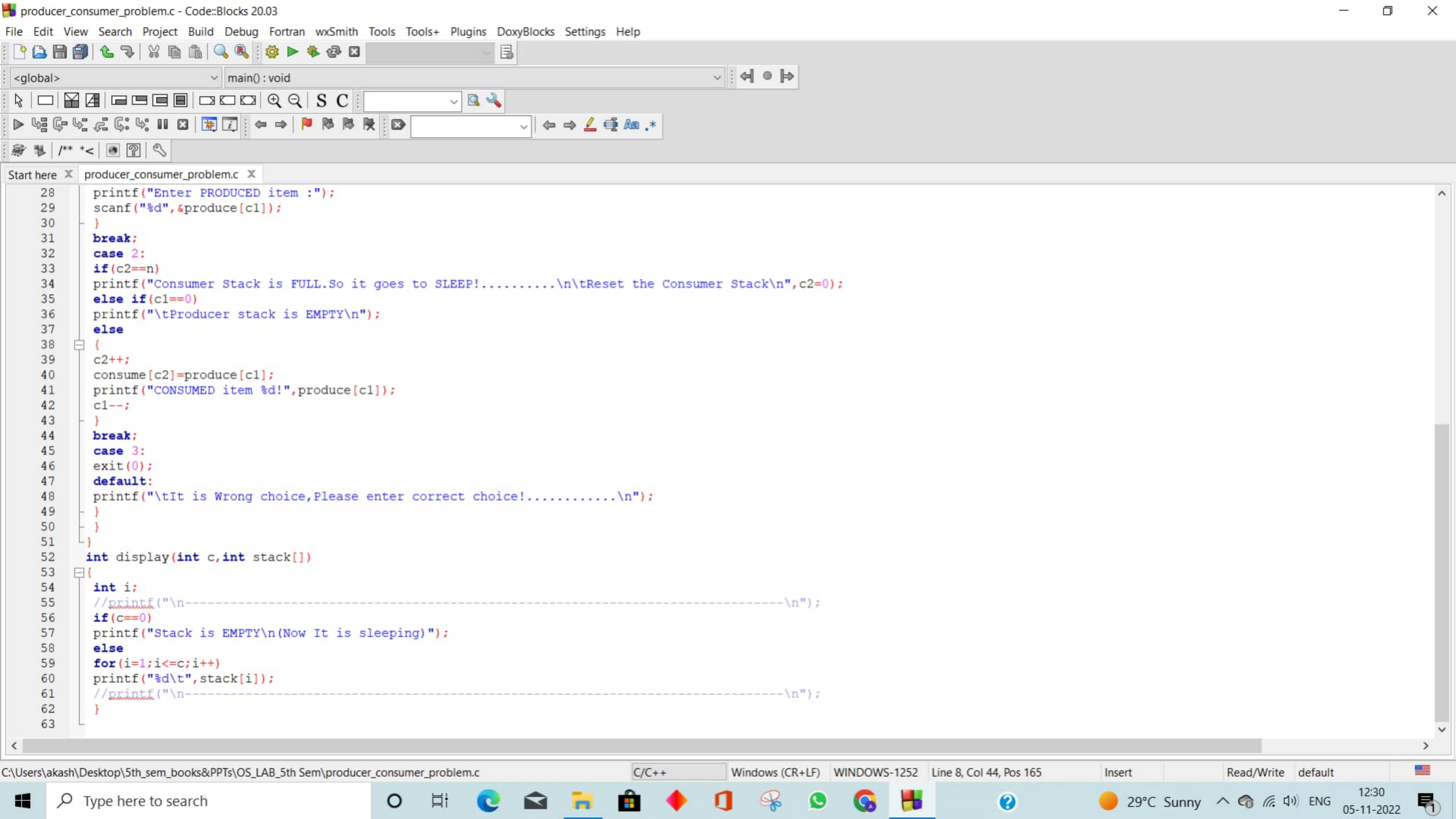
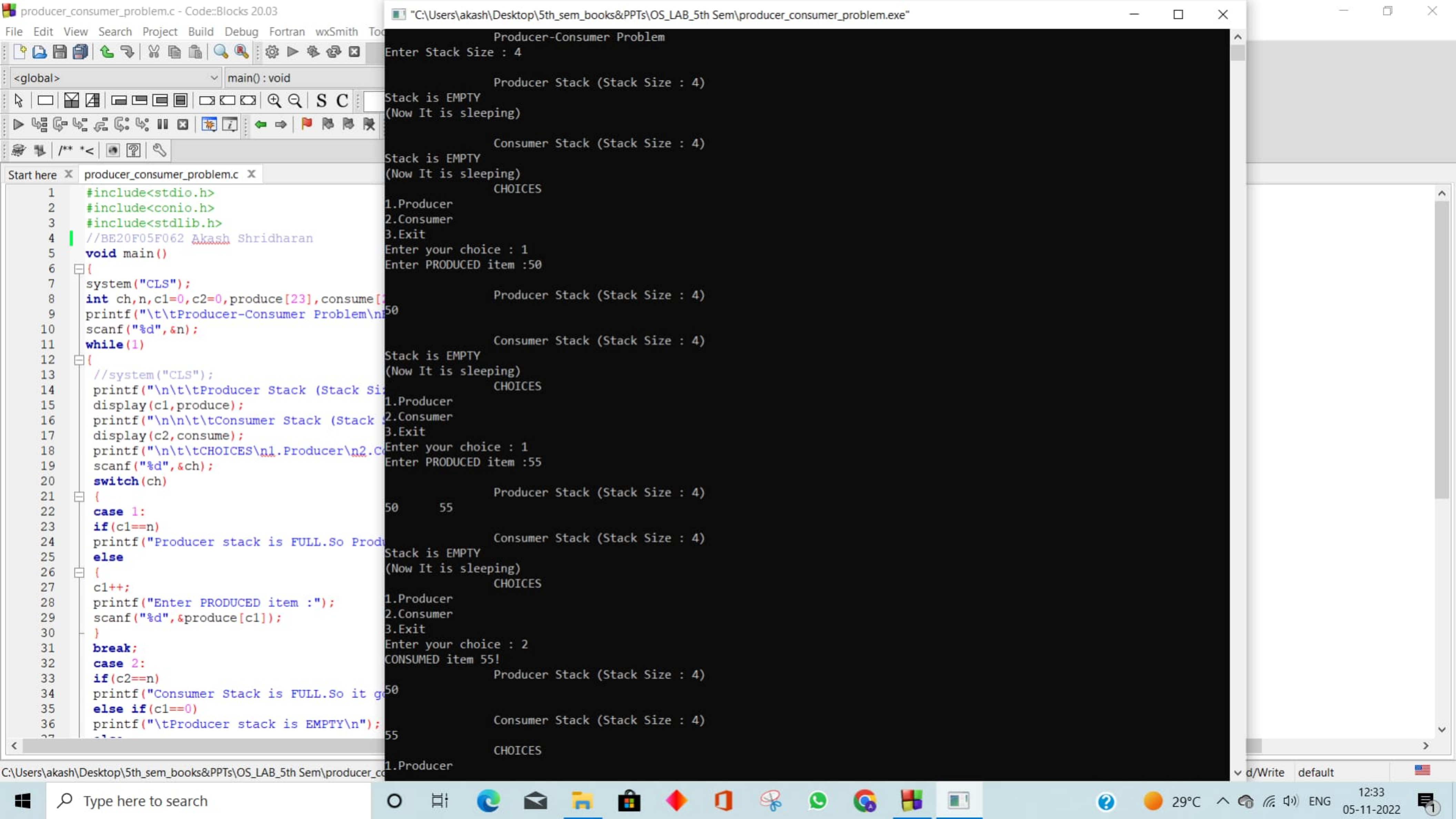

producer_consumer_problem.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : void
Start here
producer_consumer_problem.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\producer_consumer_problem.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 8, Col 44, Pos 165
Insert
Read/Write
default
Type here to search
29°C Sunny
ENG
12:30
05-11-2022

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
//BE20F05F062 Akash Shridharan
void main()
{
system("CLS");
int ch,n,c1=0,c2=0,produce[23],consume[23];
printf("\t\tProducer-Consumer Problem\nEnter Stack Size : ",n);
scanf("%d",&n);
while(1)
{
 //system("CLS");
 printf("\n\t\tProducer Stack (Stack Size : %d)\n",n);
 display(c1,produce);
 printf("\n\n\t\tConsumer Stack (Stack Size : %d)\n",n);
 display(c2,consume);
 printf("\n\t\tCHOICES\n1.Producer\n2.Consumer\n3.Exit\nEnter your choice : ");
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:
 if(c1==n)
 printf("Producer stack is FULL.So Producer goes to SLEEP\n");
 else
 {
 c1++;
 printf("Enter PRODUCED item :");
 scanf("%d",&produce[c1]);
 }
 break;
 case 2:
 if(c2==n)
 printf("Consumer Stack is FULL.So it goes to SLEEP!.........\n\tReset the Consumer Stack\n",c2=0);
 else if(c1==0)
 printf("\tProducer stack is EMPTY\n");
```

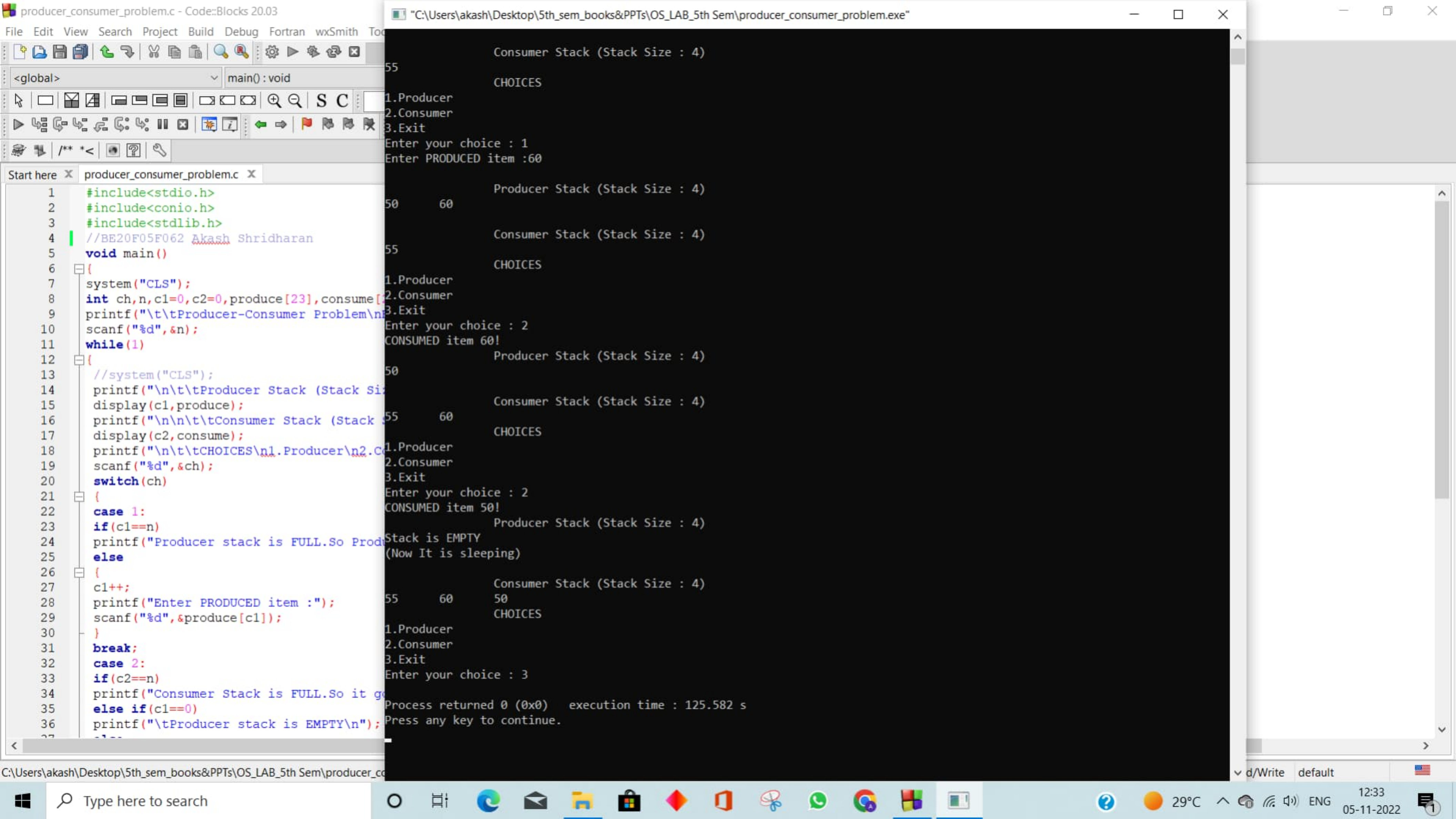

producer_consumer_problem.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : void
Start here
producer_consumer_problem.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\producer_consumer_problem.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 8, Col 44, Pos 165
Insert
Read/Write
default
Type here to search
29°C Sunny
ENG
12:30
05-11-2022

```
 printf("Enter PRODUCED item :");
 scanf("%d",&produce[c1]);
 }
 break;
 case 2:
 if(c2==n)
 printf("Consumer Stack is FULL.So it goes to SLEEP!.........\n\tReset the Consumer Stack\n",c2=0);
 else if(c1==0)
 printf("\tProducer stack is EMPTY\n");
 else
 {
 c2++;
 consume[c2]=produce[c1];
 printf("CONSUMED item %d!",produce[c1]);
 c1--;
 }
 break;
 case 3:
 exit(0);
 default:
 printf("\tIt is Wrong choice,Please enter correct choice!...........\n");
 }
 }
}
int display(int c,int stack[])
{
 int i;
 //printf("\n-------------------------------------------------------------------------------\n");
 if(c==0)
 printf("Stack is EMPTY\n(Now It is sleeping)");
 else
 for(i=1;i<=c;i++)
 printf("%d\t",stack[i]);
 //printf("\n-------------------------------------------------------------------------------\n");
 }
```


producer_consumer_problem.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Too
<global>
main() : void
Start here
producer_consumer_problem.c
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\producer_consumer_problem.exe"
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\producer_co
d/Write
default
Type here to search
29°C
ENG
12:33
05-11-2022

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
//BE20F05F062 Akash Shridharan
void main()
{
system("CLS");
int ch,n,c1=0,c2=0,produce[23],consume[
printf("\t\tProducer-Consumer Problem\n
scanf("%d",&n);
while(1)
{
 //system("CLS");
 printf("\n\t\tProducer Stack (Stack Si
 display(c1,produce);
 printf("\n\n\t\tConsumer Stack (Stack
 display(c2,consume);
 printf("\n\t\tCHOICES\n1.Producer\n2.C
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:
 if(c1==n)
 printf("Producer stack is FULL.So Prod
 else
 {
 c1++;
 printf("Enter PRODUCED item :");
 scanf("%d",&produce[c1]);
 }
 break;
 case 2:
 if(c2==n)
 printf("Consumer Stack is FULL.So it g
 else if(c1==0)
 printf("\tProducer stack is EMPTY\n");
```

```
                Producer-Consumer Problem
Enter Stack Size : 4

                Producer Stack (Stack Size : 4)
Stack is EMPTY
(Now It is sleeping)

                Consumer Stack (Stack Size : 4)
Stack is EMPTY
(Now It is sleeping)
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 1
Enter PRODUCED item :50

                Producer Stack (Stack Size : 4)
50

                Consumer Stack (Stack Size : 4)
Stack is EMPTY
(Now It is sleeping)
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 1
Enter PRODUCED item :55

                Producer Stack (Stack Size : 4)
50      55

                Consumer Stack (Stack Size : 4)
Stack is EMPTY
(Now It is sleeping)
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 2
CONSUMED item 55!
                Producer Stack (Stack Size : 4)
50

                Consumer Stack (Stack Size : 4)
55

                CHOICES
1.Producer
```

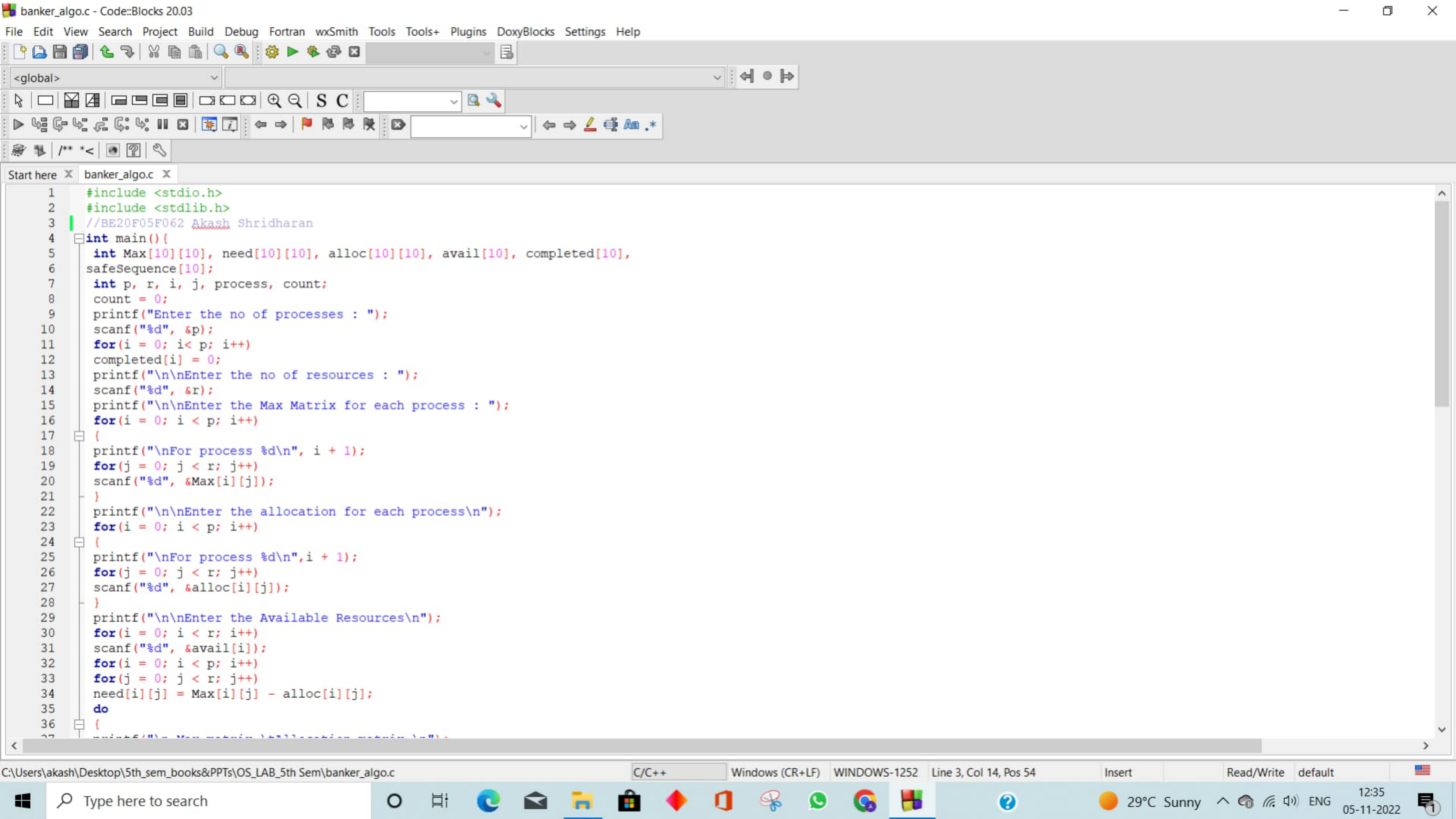


producer_consumer_problem.c - Code::Blocks 20.03

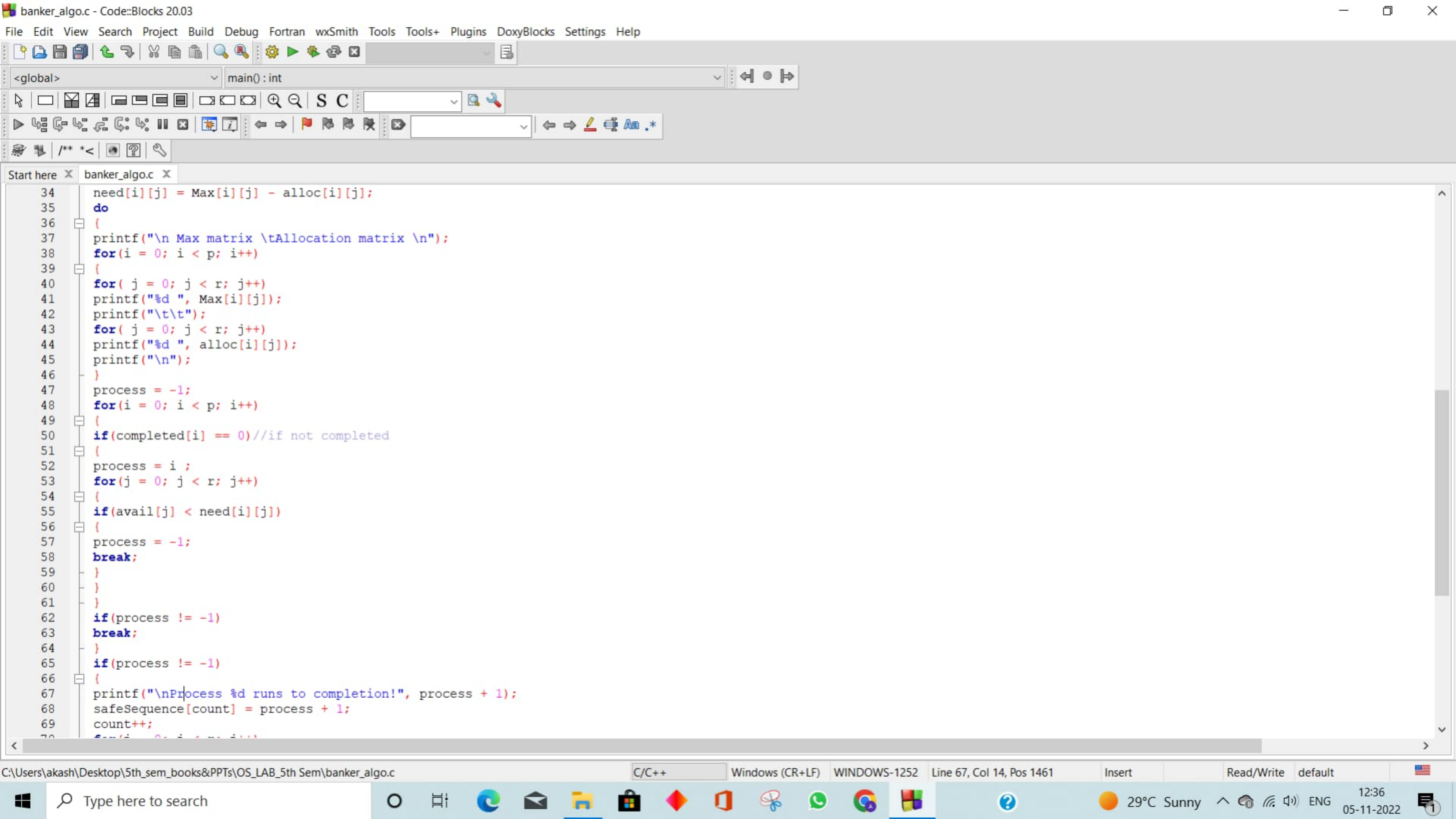
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\producer_consumer_problem.exe"

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
//BE20F05F062 Akash Shridharan
void main()
{
system("CLS");
int ch,n,c1=0,c2=0,produce[23],consume[
printf("\t\tProducer-Consumer Problem\n
scanf("%d",&n);
while(1)
{
 //system("CLS");
 printf("\n\t\tProducer Stack (Stack Si
 display(c1,produce);
 printf("\n\n\t\tConsumer Stack (Stack 
 display(c2,consume);
 printf("\n\t\tCHOICES\n1.Producer\n2.C
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:
 if(c1==n)
 printf("Producer stack is FULL.So Prod
 else
 {
 c1++;
 printf("Enter PRODUCED item :");
 scanf("%d",&produce[c1]);
 }
 break;
 case 2:
 if(c2==n)
 printf("Consumer Stack is FULL.So it g
 else if(c1==0)
 printf("\tProducer stack is EMPTY\n");
```

```
                Consumer Stack (Stack Size : 4)
55
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 1
Enter PRODUCED item :60

                Producer Stack (Stack Size : 4)
50      60

                Consumer Stack (Stack Size : 4)
55
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 2
CONSUMED item 60!
                Producer Stack (Stack Size : 4)
50

                Consumer Stack (Stack Size : 4)
55      60
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 2
CONSUMED item 50!
                Producer Stack (Stack Size : 4)
Stack is EMPTY
(Now It is sleeping)

                Consumer Stack (Stack Size : 4)
55      60      50
                CHOICES
1.Producer
2.Consumer
3.Exit
Enter your choice : 3

Process returned 0 (0x0)   execution time : 125.582 s
Press any key to continue.
```


banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
banker_algo.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 3, Col 14, Pos 54
Insert
Read/Write
default
Type here to search
29°C Sunny
ENG
12:35
05-11-2022

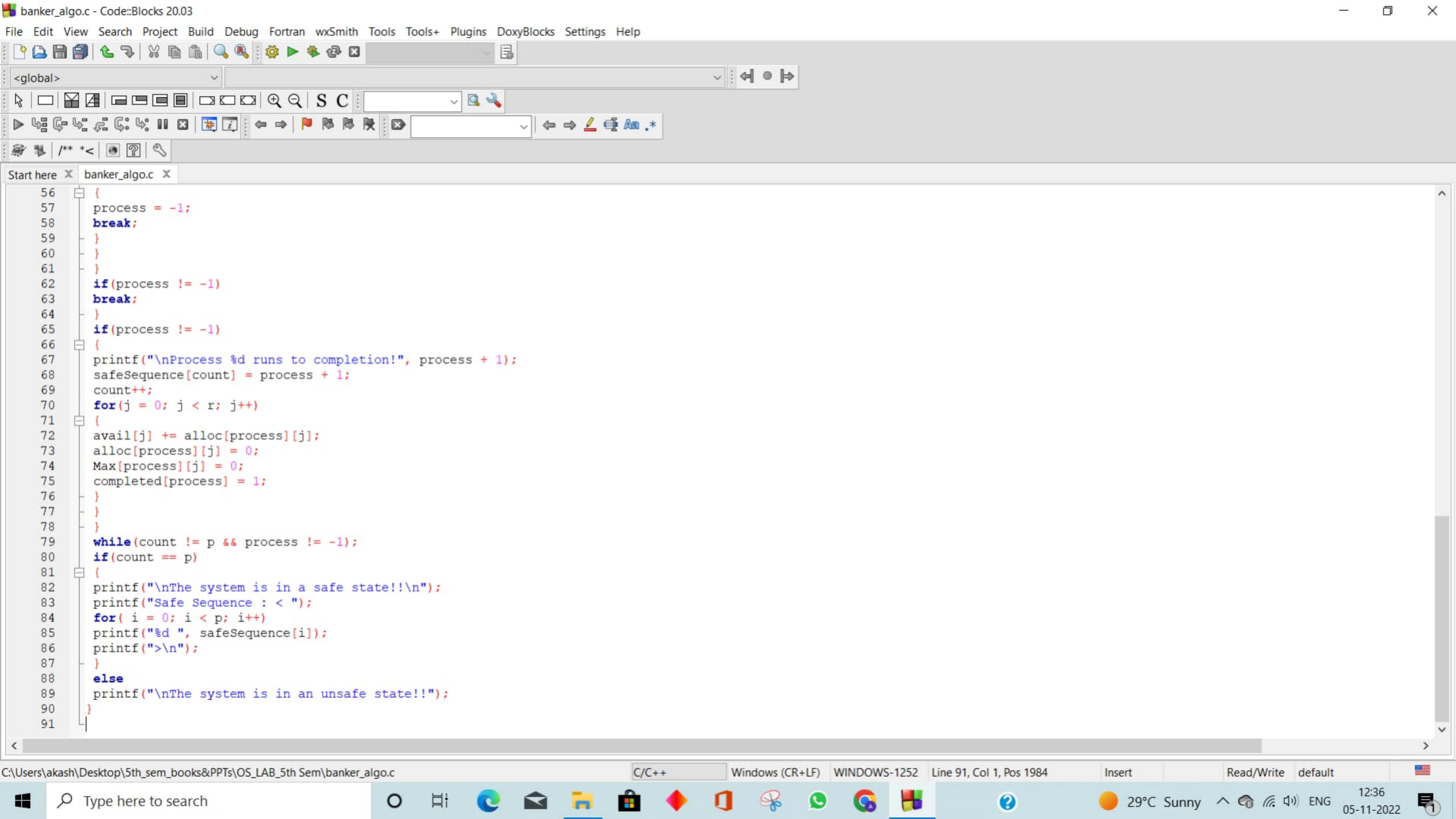
```c
#include <stdio.h>
#include <stdlib.h>
//BE20F05F062 Akash Shridharan
int main(){
 int Max[10][10], need[10][10], alloc[10][10], avail[10], completed[10],
safeSequence[10];
 int p, r, i, j, process, count;
 count = 0;
 printf("Enter the no of processes : ");
 scanf("%d", &p);
 for(i = 0; i< p; i++)
 completed[i] = 0;
 printf("\n\nEnter the no of resources : ");
 scanf("%d", &r);
 printf("\n\nEnter the Max Matrix for each process : ");
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n", i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &Max[i][j]);
 }
 printf("\n\nEnter the allocation for each process\n");
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n",i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &alloc[i][j]);
 }
 printf("\n\nEnter the Available Resources\n");
 for(i = 0; i < r; i++)
 scanf("%d", &avail[i]);
 for(i = 0; i < p; i++)
 for(j = 0; j < r; j++)
 need[i][j] = Max[i][j] - alloc[i][j];
 do
 {
```



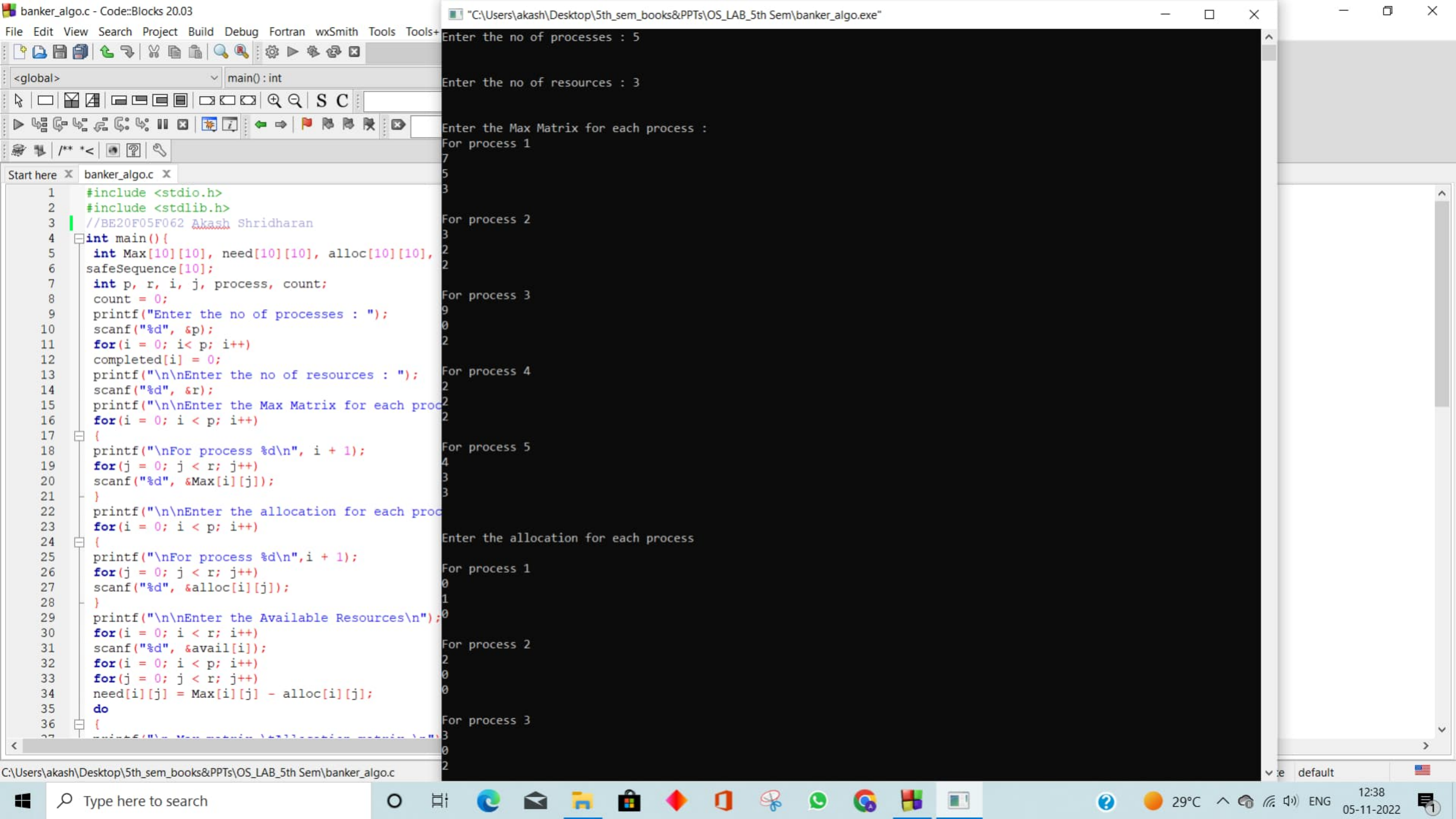
banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
Start here
banker_algo.c

```
need[i][j] = Max[i][j] - alloc[i][j];
do
{
printf("\n Max matrix \tAllocation matrix \n");
for(i = 0; i < p; i++)
{
for( j = 0; j < r; j++)
printf("%d ", Max[i][j]);
printf("\t\t");
for( j = 0; j < r; j++)
printf("%d ", alloc[i][j]);
printf("\n");
}
process = -1;
for(i = 0; i < p; i++)
{
if(completed[i] == 0)//if not completed
{
process = i ;
for(j = 0; j < r; j++)
{
if(avail[j] < need[i][j])
{
process = -1;
break;
}
}
}
if(process != -1)
break;
}
if(process != -1)
{
printf("\nProcess %d runs to completion!", process + 1);
safeSequence[count] = process + 1;
count++;
```

C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 67, Col 14, Pos 1461
Insert
Read/Write
default
Type here to search
29°C Sunny
ENG
12:36
05-11-2022

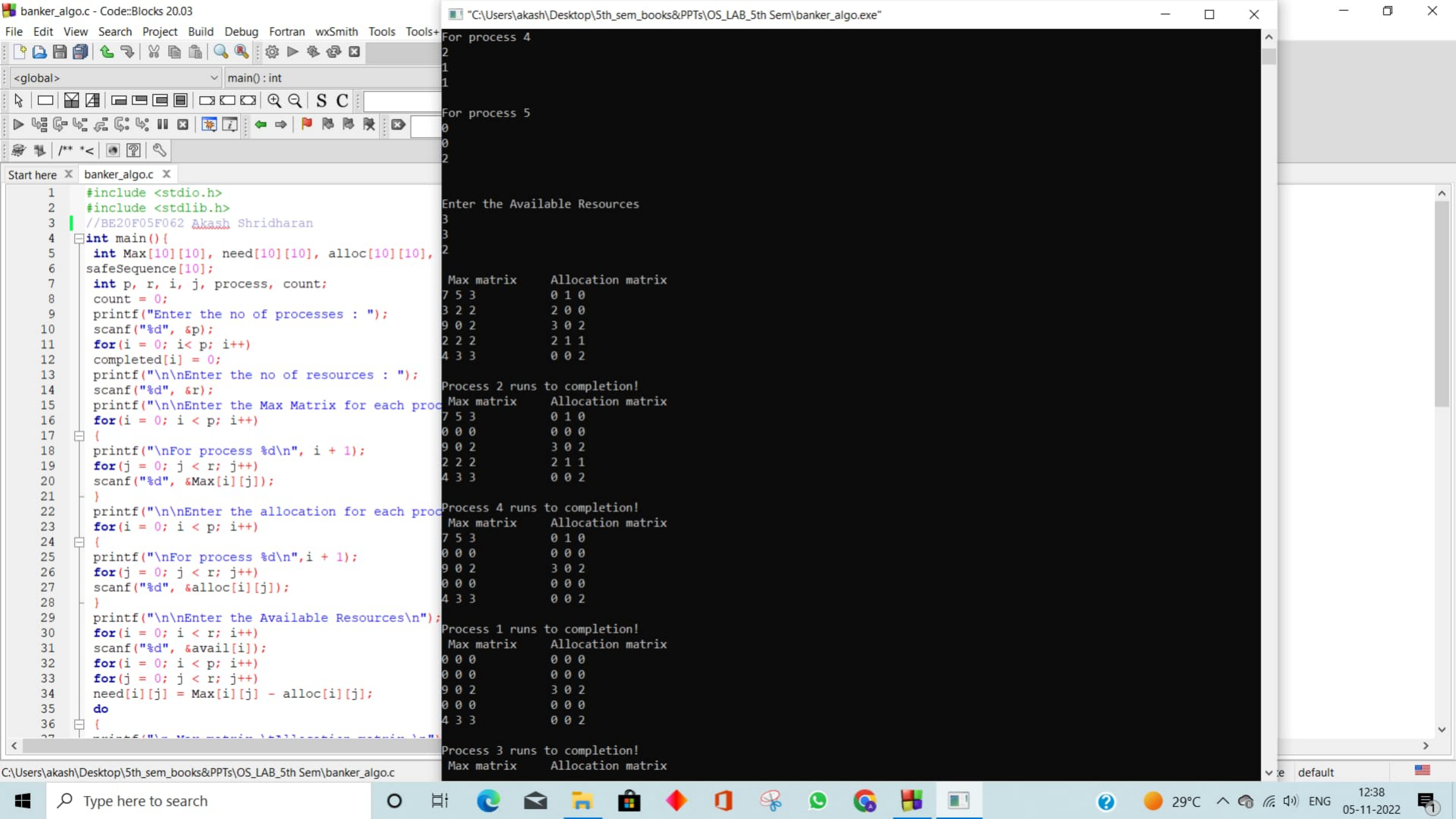

banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
banker_algo.c

```
{
process = -1;
break;
}
}
}
if(process != -1)
break;
}
if(process != -1)
{
printf("\nProcess %d runs to completion!", process + 1);
safeSequence[count] = process + 1;
count++;
for(j = 0; j < r; j++)
{
avail[j] += alloc[process][j];
alloc[process][j] = 0;
Max[process][j] = 0;
completed[process] = 1;
}
}
}
while(count != p && process != -1);
if(count == p)
{
printf("\nThe system is in a safe state!!\n");
printf("Safe Sequence : < ");
for( i = 0; i < p; i++)
printf("%d ", safeSequence[i]);
printf(">\n");
}
else
printf("\nThe system is in an unsafe state!!");
}

```


banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+
<global>
main() : int
Start here
banker_algo.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.c
default
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.exe"
Type here to search
29°C
ENG
12:38
05-11-2022

```
#include <stdio.h>
#include <stdlib.h>
//BE20F05F062 Akash Shridharan
int main(){
 int Max[10][10], need[10][10], alloc[10][10],
safeSequence[10];
 int p, r, i, j, process, count;
 count = 0;
 printf("Enter the no of processes : ");
 scanf("%d", &p);
 for(i = 0; i< p; i++)
 completed[i] = 0;
 printf("\n\nEnter the no of resources : ");
 scanf("%d", &r);
 printf("\n\nEnter the Max Matrix for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n", i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &Max[i][j]);
 }
 printf("\n\nEnter the allocation for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n",i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &alloc[i][j]);
 }
 printf("\n\nEnter the Available Resources\n");
 for(i = 0; i < r; i++)
 scanf("%d", &avail[i]);
 for(i = 0; i < p; i++)
 for(j = 0; j < r; j++)
 need[i][j] = Max[i][j] - alloc[i][j];
 do
 {
```

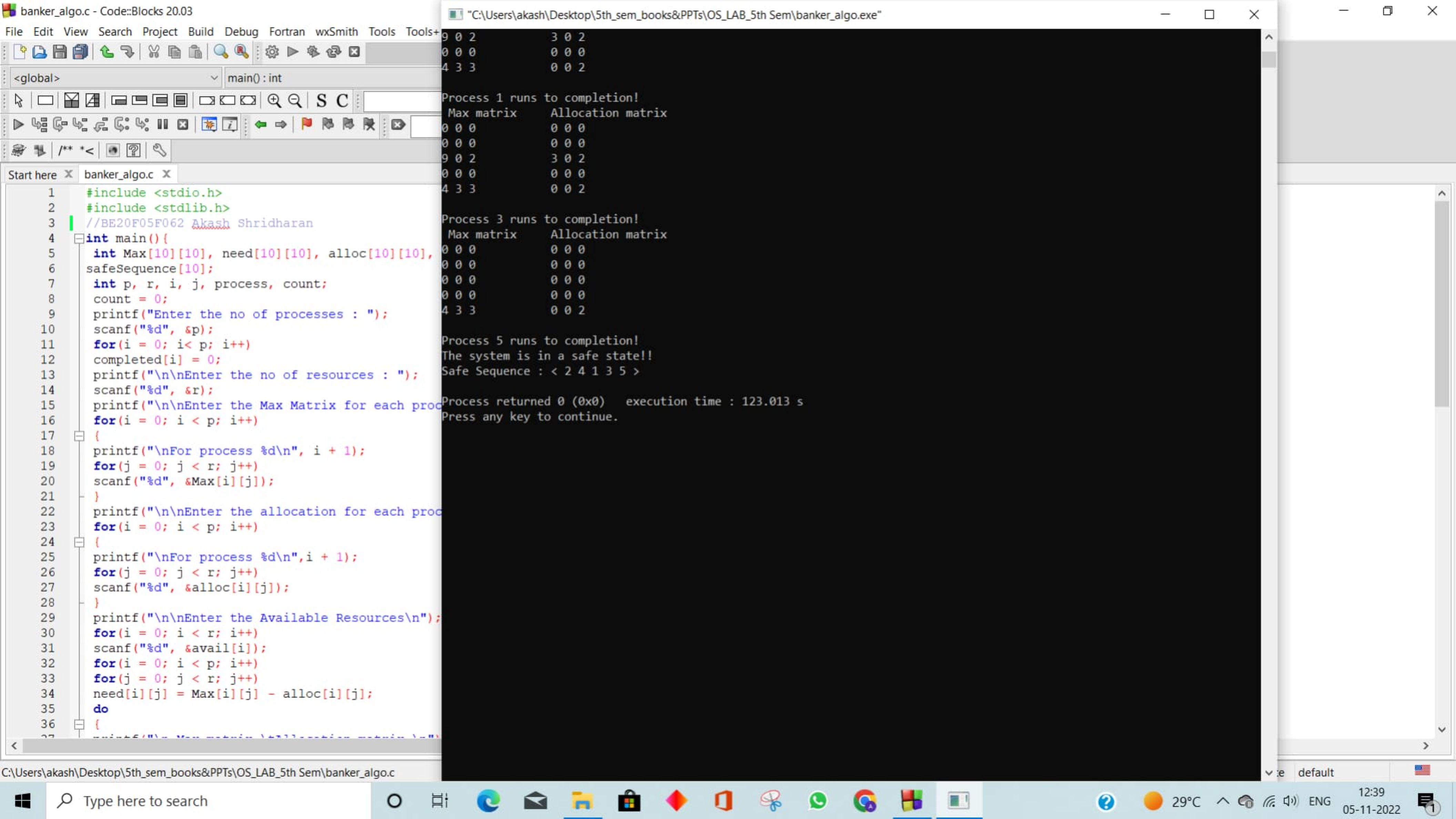
```
Enter the no of processes : 5


Enter the no of resources : 3


Enter the Max Matrix for each process :
For process 1
7
5
3


For process 2
3
2
2


For process 3
9
0
2


For process 4
2
2
2


For process 5
4
3
3


Enter the allocation for each process

For process 1
0
1
0


For process 2
2
0
0


For process 3
3
0
2
```


banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+
<global>
main() : int
Start here
banker_algo.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.c
default
Type here to search
29°C
ENG
12:38
05-11-2022

```
#include <stdio.h>
#include <stdlib.h>
//BE20F05F062 Akash Shridharan
int main(){
 int Max[10][10], need[10][10], alloc[10][10],
safeSequence[10];
 int p, r, i, j, process, count;
 count = 0;
 printf("Enter the no of processes : ");
 scanf("%d", &p);
 for(i = 0; i< p; i++)
 completed[i] = 0;
 printf("\n\nEnter the no of resources : ");
 scanf("%d", &r);
 printf("\n\nEnter the Max Matrix for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n", i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &Max[i][j]);
 }
 printf("\n\nEnter the allocation for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n",i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &alloc[i][j]);
 }
 printf("\n\nEnter the Available Resources\n");
 for(i = 0; i < r; i++)
 scanf("%d", &avail[i]);
 for(i = 0; i < p; i++)
 for(j = 0; j < r; j++)
 need[i][j] = Max[i][j] - alloc[i][j];
 do
 {
```

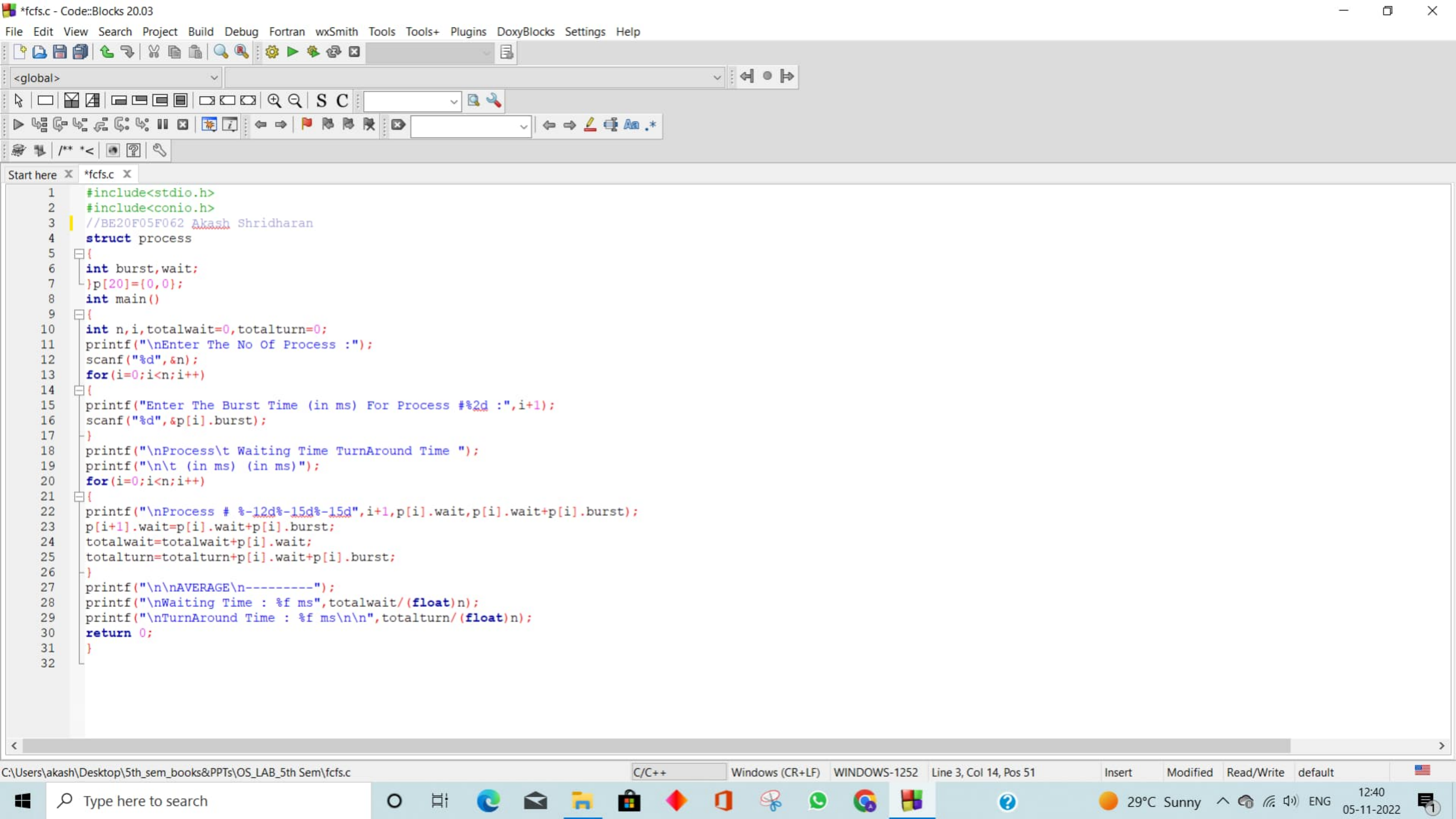
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.exe"

```
For process 4
2
1
1


For process 5
0
0
2



Enter the Available Resources
3
3
2


 Max matrix      Allocation matrix
7 5 3            0 1 0
3 2 2            2 0 0
9 0 2            3 0 2
2 2 2            2 1 1
4 3 3            0 0 2

Process 2 runs to completion!
 Max matrix      Allocation matrix
7 5 3            0 1 0
0 0 0            0 0 0
9 0 2            3 0 2
2 2 2            2 1 1
4 3 3            0 0 2

Process 4 runs to completion!
 Max matrix      Allocation matrix
7 5 3            0 1 0
0 0 0            0 0 0
9 0 2            3 0 2
0 0 0            0 0 0
4 3 3            0 0 2

Process 1 runs to completion!
 Max matrix      Allocation matrix
0 0 0            0 0 0
0 0 0            0 0 0
9 0 2            3 0 2
0 0 0            0 0 0
4 3 3            0 0 2

Process 3 runs to completion!
 Max matrix      Allocation matrix
```

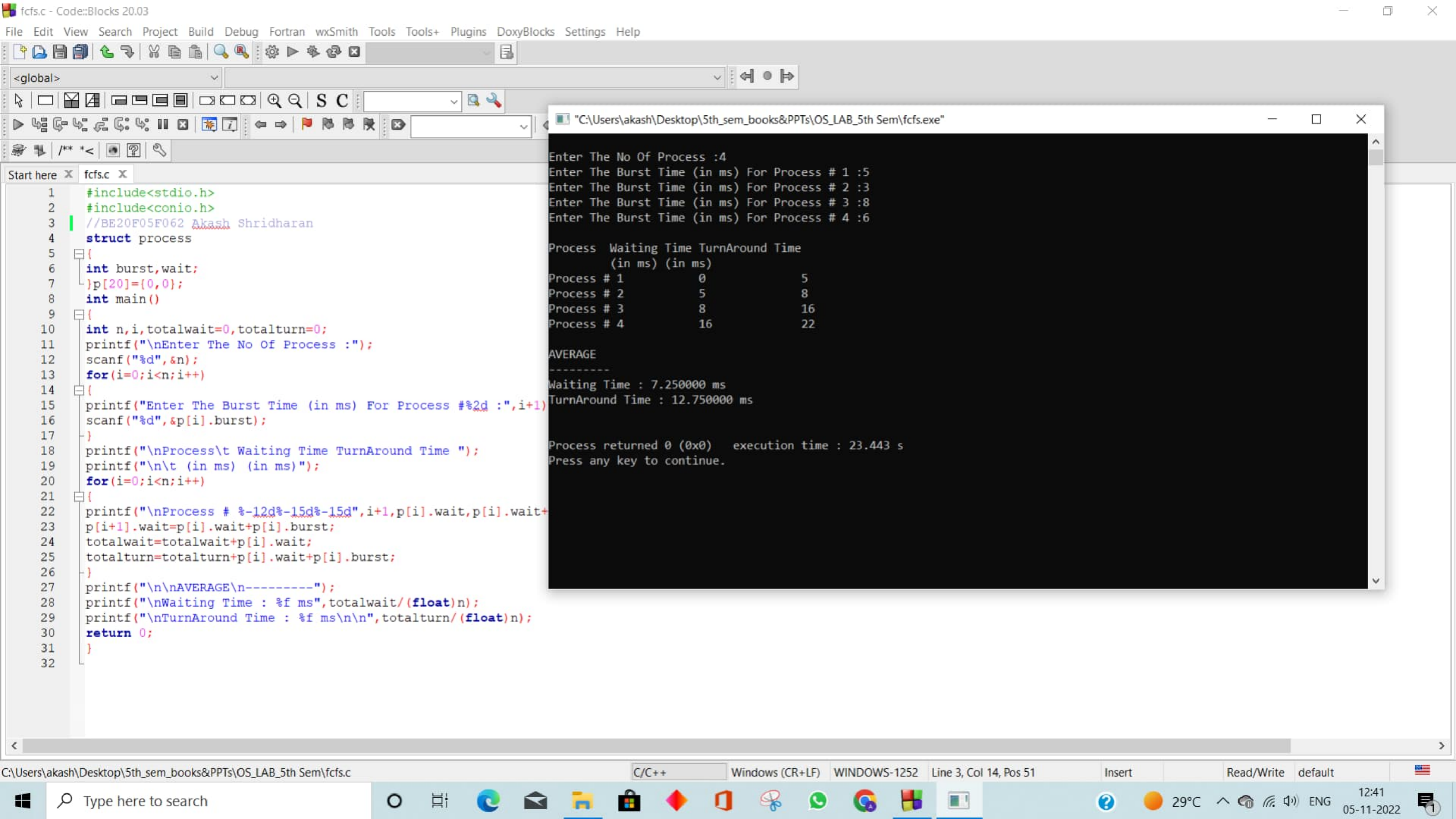

banker_algo.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+
<global>
main() : int
Start here
banker_algo.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.c
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\banker_algo.exe"
default
Type here to search
29°C
ENG
12:39
05-11-2022

```c
#include <stdio.h>
#include <stdlib.h>
//BE20F05F062 Akash Shridharan
int main(){
 int Max[10][10], need[10][10], alloc[10][10],
safeSequence[10];
 int p, r, i, j, process, count;
 count = 0;
 printf("Enter the no of processes : ");
 scanf("%d", &p);
 for(i = 0; i< p; i++)
 completed[i] = 0;
 printf("\n\nEnter the no of resources : ");
 scanf("%d", &r);
 printf("\n\nEnter the Max Matrix for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n", i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &Max[i][j]);
 }
 printf("\n\nEnter the allocation for each proc
 for(i = 0; i < p; i++)
 {
 printf("\nFor process %d\n",i + 1);
 for(j = 0; j < r; j++)
 scanf("%d", &alloc[i][j]);
 }
 printf("\n\nEnter the Available Resources\n");
 for(i = 0; i < r; i++)
 scanf("%d", &avail[i]);
 for(i = 0; i < p; i++)
 for(j = 0; j < r; j++)
 need[i][j] = Max[i][j] - alloc[i][j];
 do
 {
```

```
9 0 2           3 0 2
0 0 0           0 0 0
4 3 3           0 0 2

Process 1 runs to completion!
 Max matrix     Allocation matrix
0 0 0           0 0 0
0 0 0           0 0 0
9 0 2           3 0 2
0 0 0           0 0 0
4 3 3           0 0 2

Process 3 runs to completion!
 Max matrix     Allocation matrix
0 0 0           0 0 0
0 0 0           0 0 0
0 0 0           0 0 0
0 0 0           0 0 0
4 3 3           0 0 2

Process 5 runs to completion!
The system is in a safe state!!
Safe Sequence : < 2 4 1 3 5 >

Process returned 0 (0x0)   execution time : 123.013 s
Press any key to continue.
```


*fcfs.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
*fcfs.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\fcfs.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 3, Col 14, Pos 51
Insert
Modified
Read/Write
default
Type here to search
29°C Sunny
ENG
12:40
05-11-2022

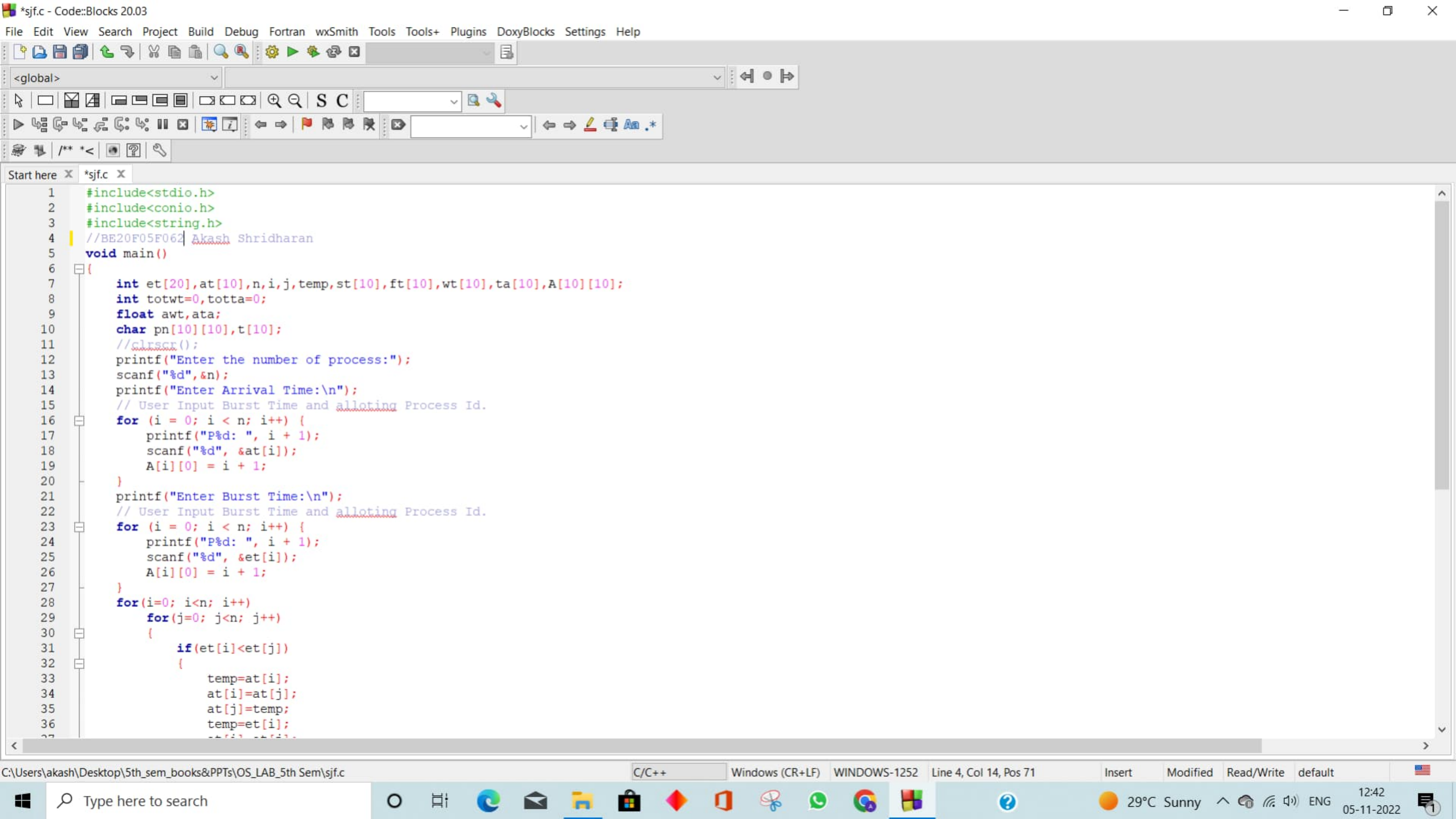
```c
#include<stdio.h>
#include<conio.h>
//BE20F05F062 Akash Shridharan
struct process
{
int burst,wait;
}p[20]={0,0};
int main()
{
int n,i,totalwait=0,totalturn=0;
printf("\nEnter The No Of Process :");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter The Burst Time (in ms) For Process #%2d :",i+1);
scanf("%d",&p[i].burst);
}
printf("\nProcess\t Waiting Time TurnAround Time ");
printf("\n\t (in ms) (in ms)");
for(i=0;i<n;i++)
{
printf("\nProcess # %-12d%-15d%-15d",i+1,p[i].wait,p[i].wait+p[i].burst);
p[i+1].wait=p[i].wait+p[i].burst;
totalwait=totalwait+p[i].wait;
totalturn=totalturn+p[i].wait+p[i].burst;
}
printf("\n\nAVERAGE\n---------");
printf("\nWaiting Time : %f ms",totalwait/(float)n);
printf("\nTurnAround Time : %f ms\n\n",totalturn/(float)n);
return 0;
}
```


fcfs.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
fcfs.c

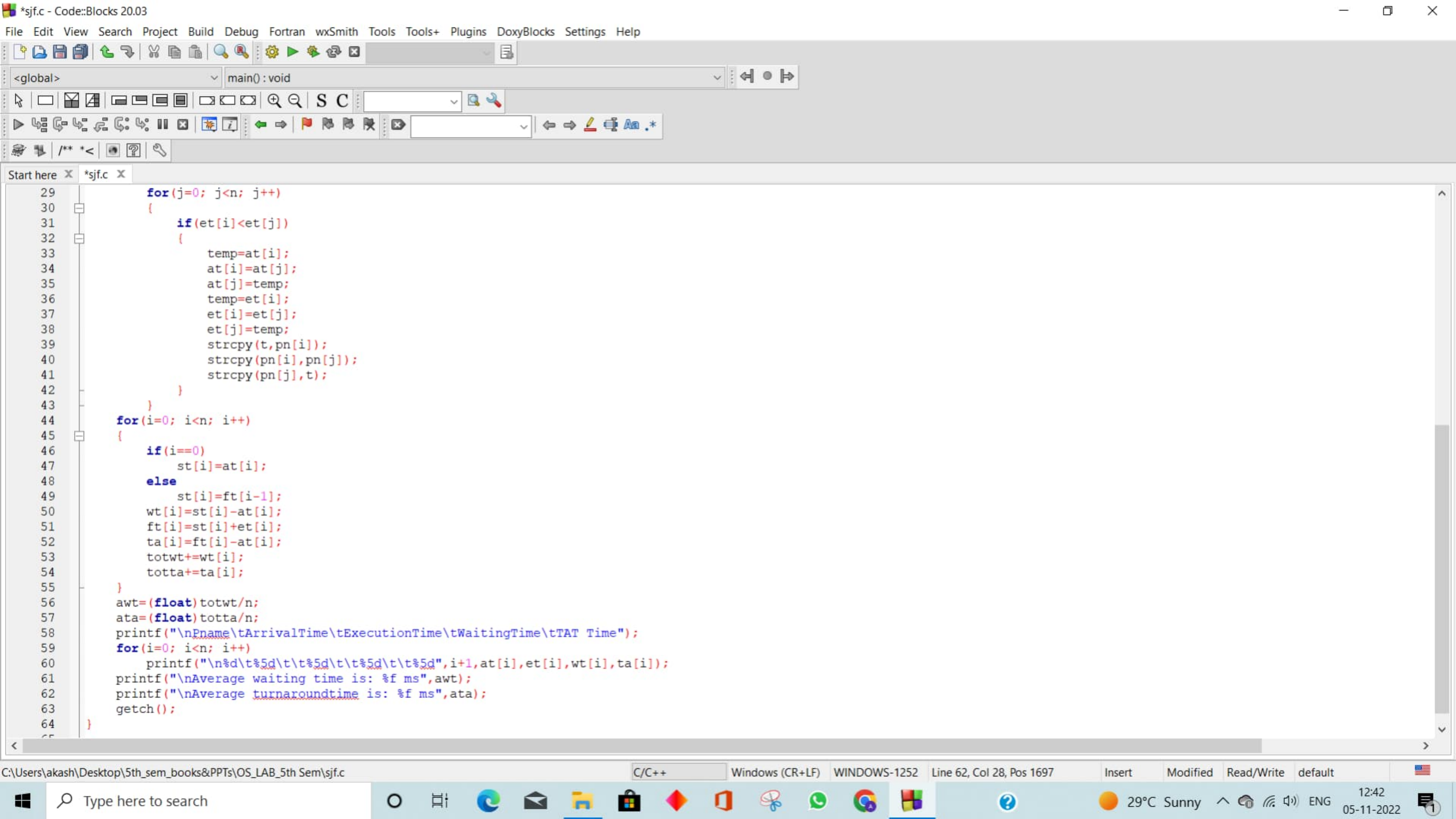
```c
#include<stdio.h>
#include<conio.h>
//BE20F05F062 Akash Shridharan
struct process
{
int burst,wait;
}p[20]={0,0};
int main()
{
int n,i,totalwait=0,totalturn=0;
printf("\nEnter The No Of Process :");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter The Burst Time (in ms) For Process #%2d :",i+1)
scanf("%d",&p[i].burst);
}
printf("\nProcess\t Waiting Time TurnAround Time ");
printf("\n\t (in ms) (in ms)");
for(i=0;i<n;i++)
{
printf("\nProcess # %-12d%-15d%-15d",i+1,p[i].wait,p[i].wait+
p[i+1].wait=p[i].wait+p[i].burst;
totalwait=totalwait+p[i].wait;
totalturn=totalturn+p[i].wait+p[i].burst;
}
printf("\n\nAVERAGE\n---------");
printf("\nWaiting Time : %f ms",totalwait/(float)n);
printf("\nTurnAround Time : %f ms\n\n",totalturn/(float)n);
return 0;
}
```

"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\fcfs.exe"

```
Enter The No Of Process :4
Enter The Burst Time (in ms) For Process # 1 :5
Enter The Burst Time (in ms) For Process # 2 :3
Enter The Burst Time (in ms) For Process # 3 :8
Enter The Burst Time (in ms) For Process # 4 :6

Process  Waiting Time TurnAround Time
         (in ms) (in ms)
Process # 1          0              5
Process # 2          5              8
Process # 3          8              16
Process # 4          16             22

AVERAGE
---------
Waiting Time : 7.250000 ms
TurnAround Time : 12.750000 ms


Process returned 0 (0x0)   execution time : 23.443 s
Press any key to continue.
```

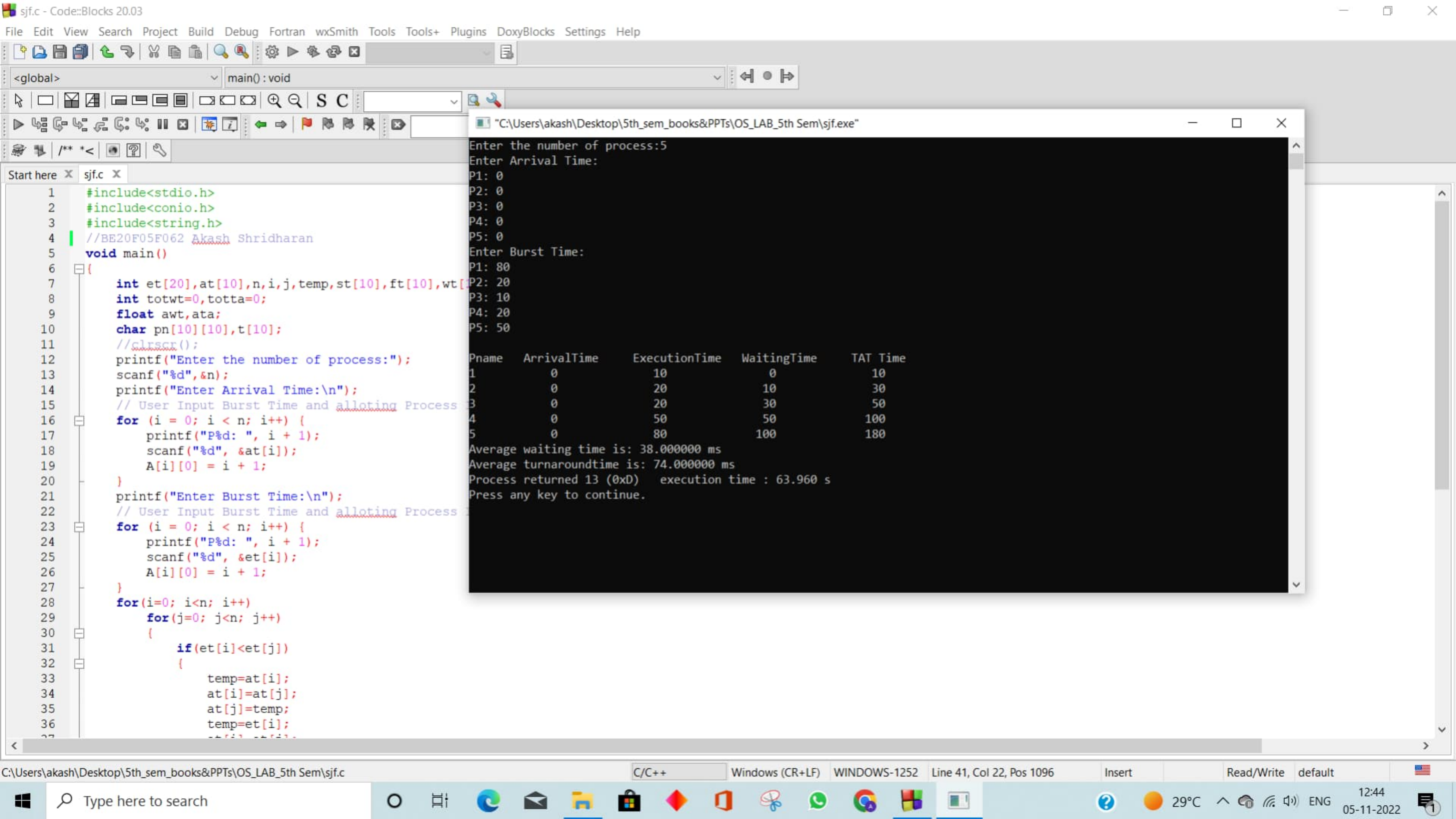

*sjf.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
*sjf.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\sjf.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 4, Col 14, Pos 71
Insert
Modified
Read/Write
default
Type here to search
29°C Sunny
ENG
12:42
05-11-2022

```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
//BE20F05F062 Akash Shridharan
void main()
{
    int et[20],at[10],n,i,j,temp,st[10],ft[10],wt[10],ta[10],A[10][10];
    int totwt=0,totta=0;
    float awt,ata;
    char pn[10][10],t[10];
    //clrscr();
    printf("Enter the number of process:");
    scanf("%d",&n);
    printf("Enter Arrival Time:\n");
    // User Input Burst Time and alloting Process Id.
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &at[i]);
        A[i][0] = i + 1;
    }
    printf("Enter Burst Time:\n");
    // User Input Burst Time and alloting Process Id.
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &et[i]);
        A[i][0] = i + 1;
    }
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
        {
            if(et[i]<et[j])
            {
                temp=at[i];
                at[i]=at[j];
                at[j]=temp;
                temp=et[i];
```



*sjf.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : void
Start here
*sjf.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\sjf.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 62, Col 28, Pos 1697
Insert
Modified
Read/Write
default
Type here to search
29°C Sunny
ENG
12:42
05-11-2022

```c
        for(j=0; j<n; j++)
        {
            if(et[i]<et[j])
            {
                temp=at[i];
                at[i]=at[j];
                at[j]=temp;
                temp=et[i];
                et[i]=et[j];
                et[j]=temp;
                strcpy(t,pn[i]);
                strcpy(pn[i],pn[j]);
                strcpy(pn[j],t);
            }
        }
    for(i=0; i<n; i++)
    {
        if(i==0)
            st[i]=at[i];
        else
            st[i]=ft[i-1];
        wt[i]=st[i]-at[i];
        ft[i]=st[i]+et[i];
        ta[i]=ft[i]-at[i];
        totwt+=wt[i];
        totta+=ta[i];
    }
    awt=(float)totwt/n;
    ata=(float)totta/n;
    printf("\nPname\tArrivalTime\tExecutionTime\tWaitingTime\tTAT Time");
    for(i=0; i<n; i++)
        printf("\n%d\t%5d\t\t%5d\t\t%5d\t\t%5d",i+1,at[i],et[i],wt[i],ta[i]);
    printf("\nAverage waiting time is: %f ms",awt);
    printf("\nAverage turnaroundtime is: %f ms",ata);
    getch();
}
```

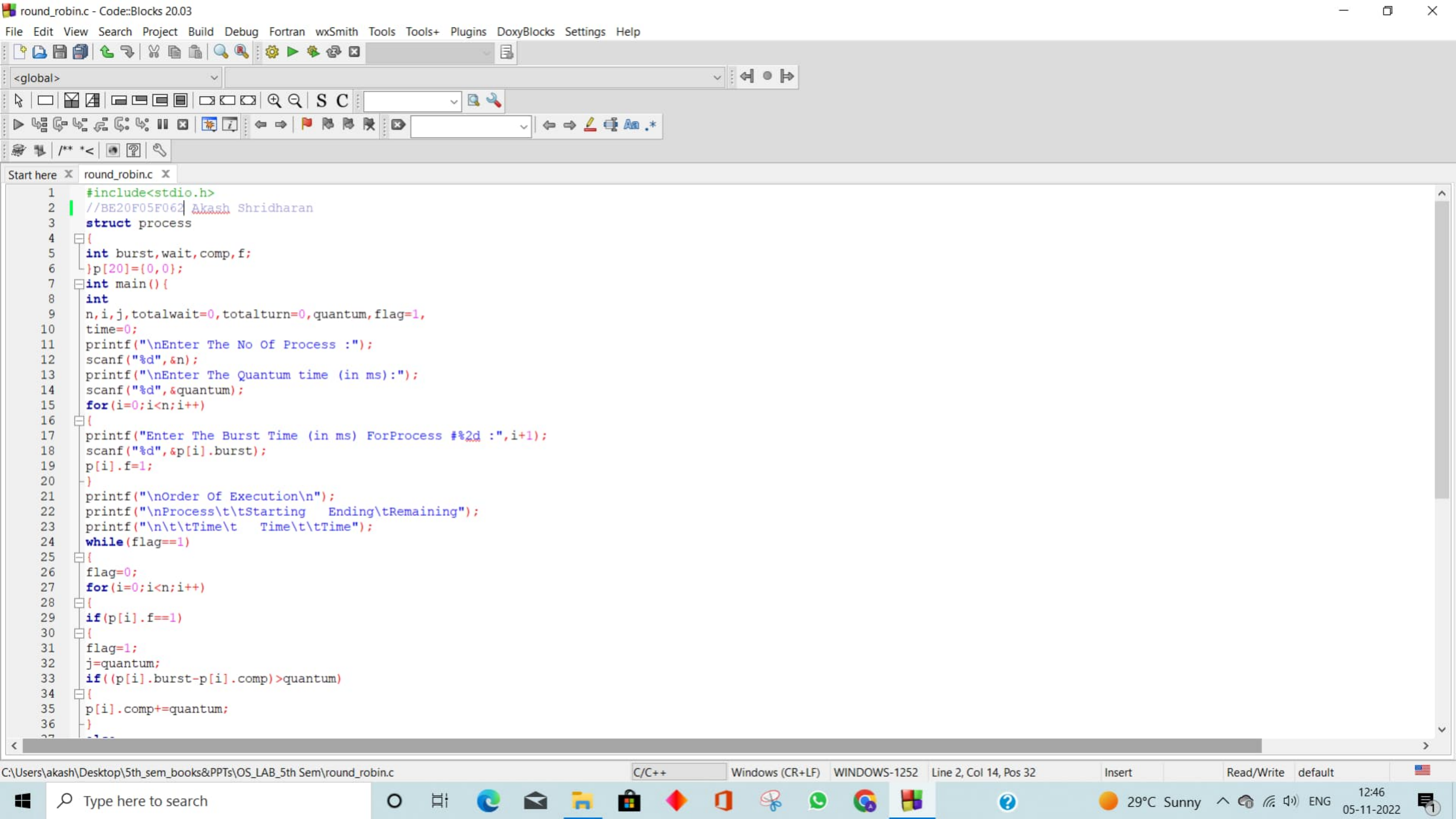


sjf.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : void
Start here
sjf.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\sjf.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 41, Col 22, Pos 1096
Insert
Read/Write
default
Type here to search
29°C
ENG
12:44
05-11-2022

```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
//BE20F05F062 Akash Shridharan
void main()
{
    int et[20],at[10],n,i,j,temp,st[10],ft[10],wt[
    int totwt=0,totta=0;
    float awt,ata;
    char pn[10][10],t[10];
    //clrscr();
    printf("Enter the number of process:");
    scanf("%d",&n);
    printf("Enter Arrival Time:\n");
    // User Input Burst Time and alloting Process
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &at[i]);
        A[i][0] = i + 1;
    }
    printf("Enter Burst Time:\n");
    // User Input Burst Time and alloting Process
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &et[i]);
        A[i][0] = i + 1;
    }
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
        {
            if(et[i]<et[j])
            {
                temp=at[i];
                at[i]=at[j];
                at[j]=temp;
                temp=et[i];
```

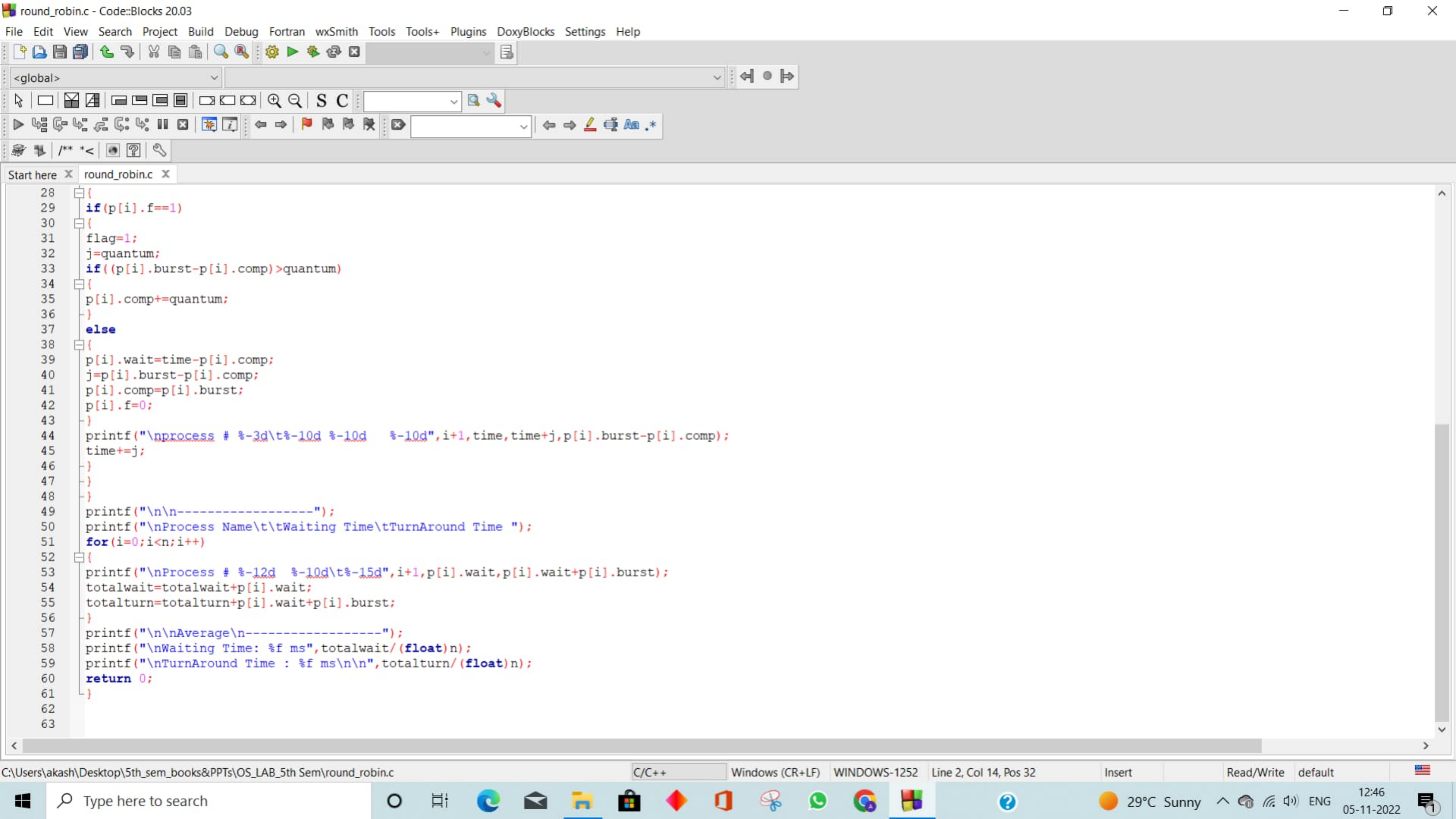
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\sjf.exe"

```
Enter the number of process:5
Enter Arrival Time:
P1: 0
P2: 0
P3: 0
P4: 0
P5: 0
Enter Burst Time:
P1: 80
P2: 20
P3: 10
P4: 20
P5: 50

Pname   ArrivalTime     ExecutionTime   WaitingTime     TAT Time
1           0               10              0              10
2           0               20              10             30
3           0               20              30             50
4           0               50              50             100
5           0               80              100            180
Average waiting time is: 38.000000 ms
Average turnaroundtime is: 74.000000 ms
Process returned 13 (0xD)   execution time : 63.960 s
Press any key to continue.
```

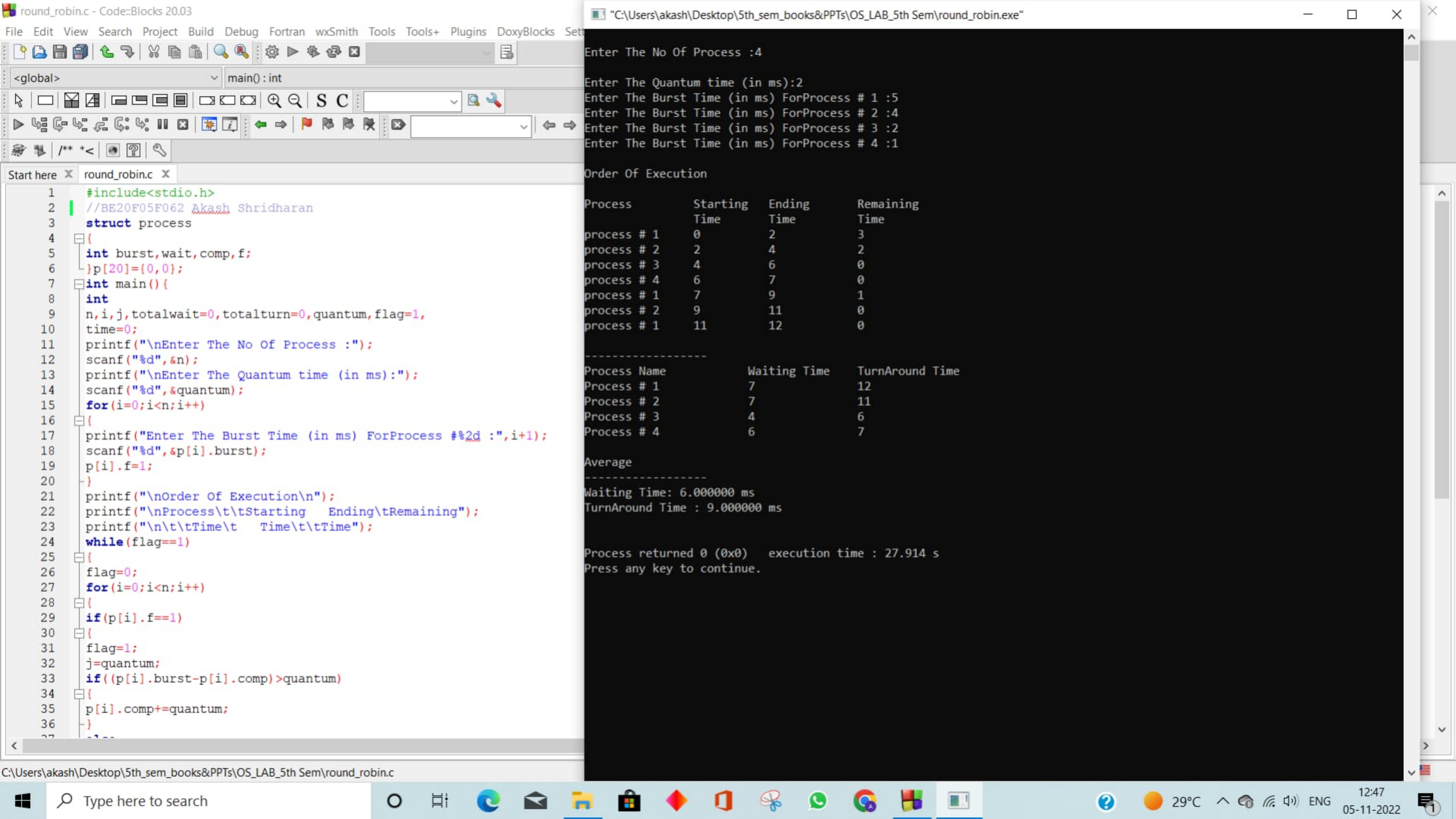

round_robin.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
round_robin.c
Type here to search
29°C Sunny
ENG
12:46
05-11-2022

```
#include<stdio.h>
//BE20F05F062 Akash Shridharan
struct process
{
int burst,wait,comp,f;
}p[20]={0,0};
int main(){
int
n,i,j,totalwait=0,totalturn=0,quantum,flag=1,
time=0;
printf("\nEnter The No Of Process :");
scanf("%d",&n);
printf("\nEnter The Quantum time (in ms):");
scanf("%d",&quantum);
for(i=0;i<n;i++)
{
printf("Enter The Burst Time (in ms) ForProcess #%2d :",i+1);
scanf("%d",&p[i].burst);
p[i].f=1;
}
printf("\nOrder Of Execution\n");
printf("\nProcess\t\tStarting   Ending\tRemaining");
printf("\n\t\tTime\t   Time\t\tTime");
while(flag==1)
{
flag=0;
for(i=0;i<n;i++)
{
if(p[i].f==1)
{
flag=1;
j=quantum;
if((p[i].burst-p[i].comp)>quantum)
{
p[i].comp+=quantum;
}
```

C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\round_robin.c   C/C++   Windows (CR+LF)   WINDOWS-1252   Line 2, Col 14, Pos 32   Insert   Read/Write   default


round_robin.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
round_robin.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\round_robin.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 2, Col 14, Pos 32
Insert
Read/Write
default
Type here to search
29°C Sunny
ENG
12:46
05-11-2022

```
{
if(p[i].f==1)
{
flag=1;
j=quantum;
if((p[i].burst-p[i].comp)>quantum)
{
p[i].comp+=quantum;
}
else
{
p[i].wait=time-p[i].comp;
j=p[i].burst-p[i].comp;
p[i].comp=p[i].burst;
p[i].f=0;
}
printf("\nprocess # %-3d\t%-10d %-10d   %-10d",i+1,time,time+j,p[i].burst-p[i].comp);
time+=j;
}
}
}
printf("\n\n-----------------");
printf("\nProcess Name\t\tWaiting Time\tTurnAround Time ");
for(i=0;i<n;i++)
{
printf("\nProcess # %-12d  %-10d\t%-15d",i+1,p[i].wait,p[i].wait+p[i].burst);
totalwait=totalwait+p[i].wait;
totalturn=totalturn+p[i].wait+p[i].burst;
}
printf("\n\nAverage\n-----------------");
printf("\nWaiting Time: %f ms",totalwait/(float)n);
printf("\nTurnAround Time : %f ms\n\n",totalturn/(float)n);
return 0;
}
```


round_robin.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Sett
<global>
main() : int
Start here
round_robin.c
C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\round_robin.c
"C:\Users\akash\Desktop\5th_sem_books&PPTs\OS_LAB_5th Sem\round_robin.exe"
Type here to search
29°C
ENG
12:47
05-11-2022

```c
#include<stdio.h>
//BE20F05F062 Akash Shridharan
struct process
{
int burst,wait,comp,f;
}p[20]={0,0};
int main(){
int
n,i,j,totalwait=0,totalturn=0,quantum,flag=1,
time=0;
printf("\nEnter The No Of Process :");
scanf("%d",&n);
printf("\nEnter The Quantum time (in ms):");
scanf("%d",&quantum);
for(i=0;i<n;i++)
{
printf("Enter The Burst Time (in ms) ForProcess #%2d :",i+1);
scanf("%d",&p[i].burst);
p[i].f=1;
}
printf("\nOrder Of Execution\n");
printf("\nProcess\t\tStarting   Ending\tRemaining");
printf("\n\t\tTime\t   Time\t\tTime");
while(flag==1)
{
flag=0;
for(i=0;i<n;i++)
{
if(p[i].f==1)
{
flag=1;
j=quantum;
if((p[i].burst-p[i].comp)>quantum)
{
p[i].comp+=quantum;
}
```

```
Enter The No Of Process :4

Enter The Quantum time (in ms):2
Enter The Burst Time (in ms) ForProcess # 1 :5
Enter The Burst Time (in ms) ForProcess # 2 :4
Enter The Burst Time (in ms) ForProcess # 3 :2
Enter The Burst Time (in ms) ForProcess # 4 :1

Order Of Execution

Process         Starting   Ending      Remaining
                Time       Time        Time
process # 1     0          2           3
process # 2     2          4           2
process # 3     4          6           0
process # 4     6          7           0
process # 1     7          9           1
process # 2     9          11          0
process # 1     11         12          0

----------------
Process Name           Waiting Time    TurnAround Time
Process # 1            7               12
Process # 2            7               11
Process # 3            4               6
Process # 4            6               7

Average
----------------
Waiting Time: 6.000000 ms
TurnAround Time : 9.000000 ms


Process returned 0 (0x0)   execution time : 27.914 s
Press any key to continue.
```