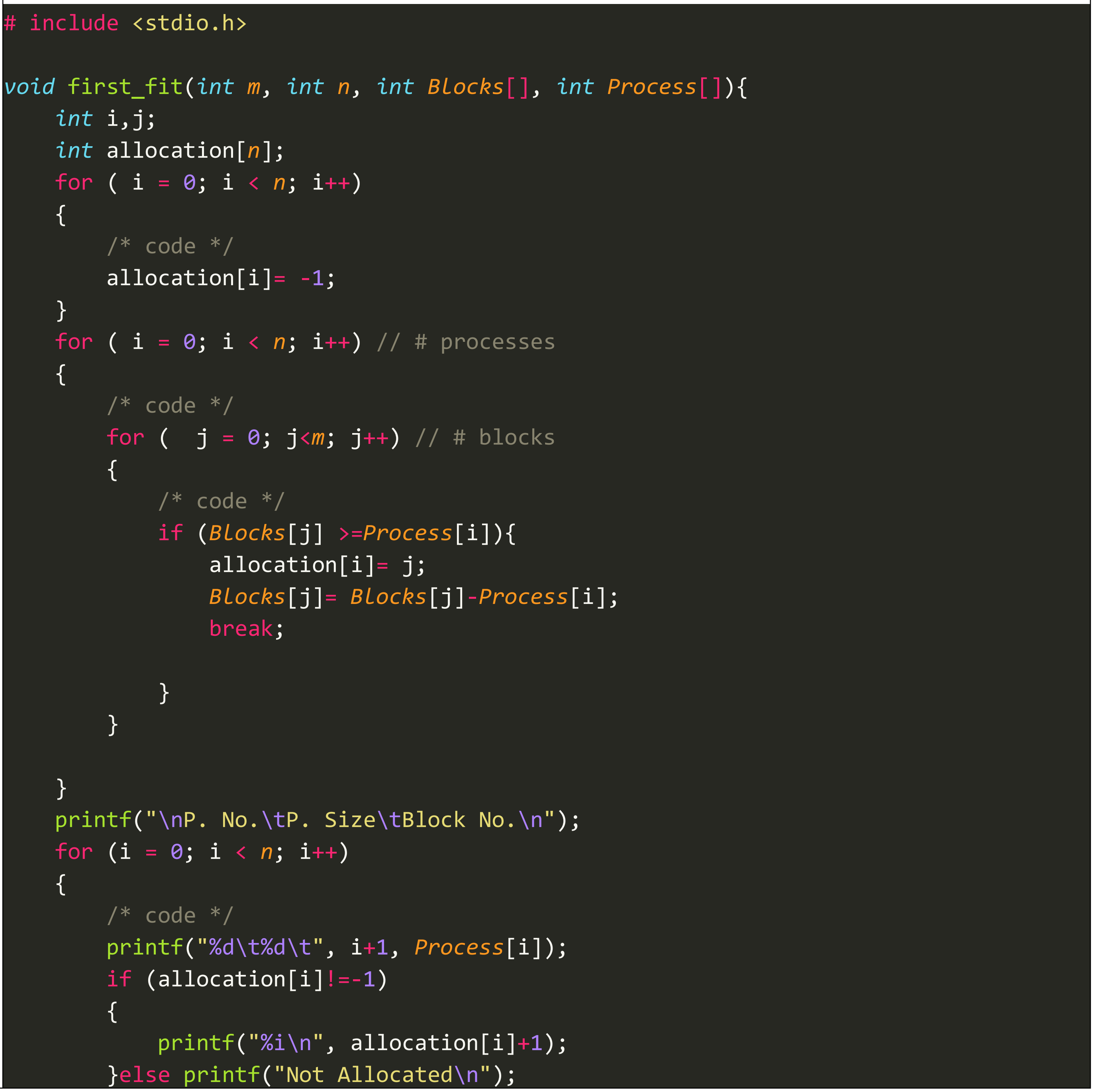
INTERNAL ASSIGNMENT

Name: Deepankar Sharma

Student ID: 233512013

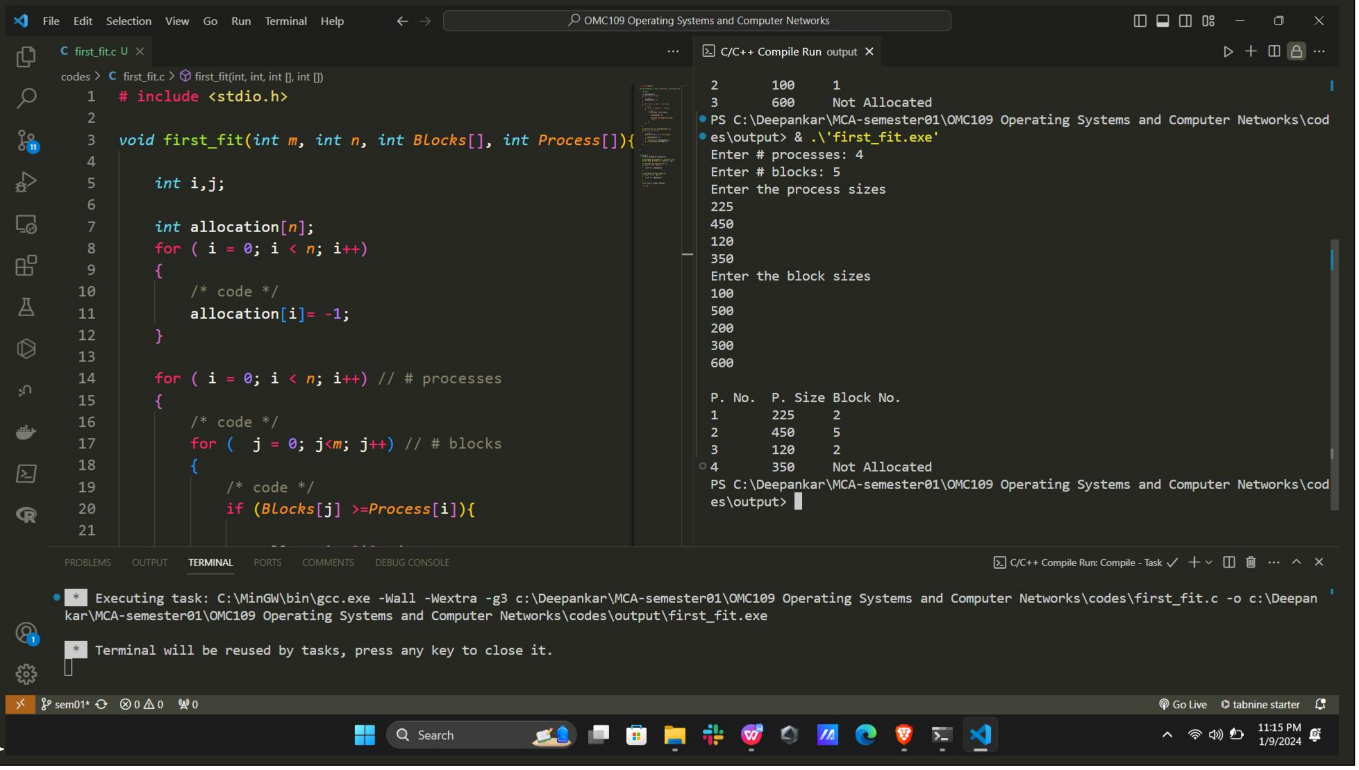
Course Code: OMC-109

Course Title: Operating Systems & Computer Networks lab

Q.No.	**Question**
1	Write a C program to Simulate the following Memory management algorithm-First fit 

```
# include <stdio.h>

void first_fit(int m, int n, int Blocks[], int Process[]){
    int i,j;
    int allocation[n];
    for ( i = 0; i < n; i++)
    {
        /* code */
        allocation[i]= -1;
    }
    for ( i = 0; i < n; i++) // # processes
    {
        /* code */
        for (  j = 0; j<m; j++) // # blocks
        {
            /* code */
            if (Blocks[j] >=Process[i]){
                allocation[i]= j;
                Blocks[j]= Blocks[j]-Process[i];
                break;

            }
        }

    }
    printf("\nP. No.\tP. Size\tBlock No.\n");
    for (i = 0; i < n; i++)
    {
        /* code */
        printf("%d\t%d\t", i+1, Process[i]);
        if (allocation[i]!=-1)
        {
            printf("%i\n", allocation[i]+1);
        }else printf("Not Allocated\n");
```

```c
    }
}
int main(){
    int m, n, Blocks[10], Process[10];
    printf("Enter # processes: "); scanf("%d", &n);
    printf("Enter # blocks: "); scanf("%d", &m);
    printf("Enter the process sizes\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &Process[i]);
    }
    printf("Enter the block sizes\n");
    for (int i = 0; i < m; i++)
    {
        scanf("%d", &Blocks[i]);
    }
    first_fit(m, n, Blocks, Process);
    return 0;
```

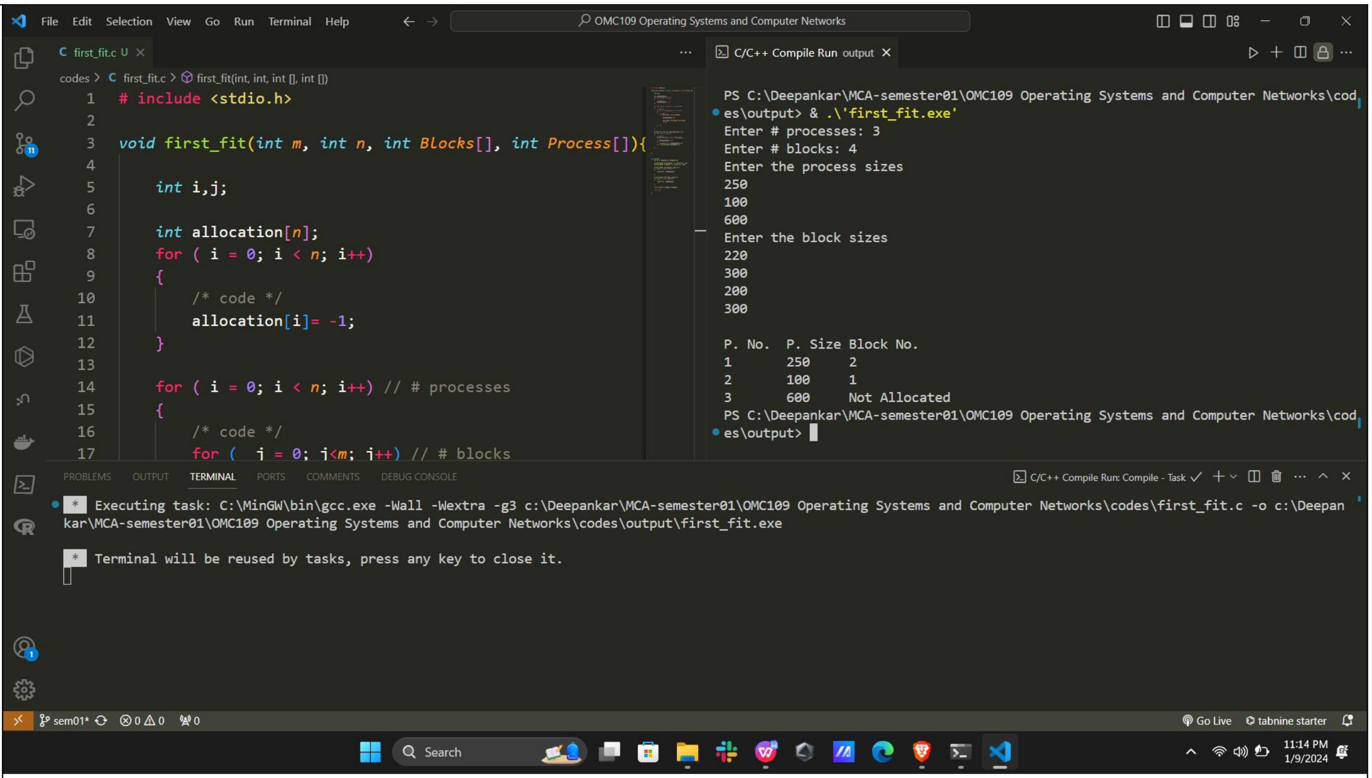


```c
# include <stdio.h>

void first_fit(int m, int n, int Blocks[], int Process[]){

    int i,j;

    int allocation[n];
    for ( i = 0; i < n; i++)
    {
        /* code */
        allocation[i]= -1;
    }

    for ( i = 0; i < n; i++) // # processes
    {
        /* code */
        for (  j = 0; j<m; j++) // # blocks
        {
            /* code */
            if (Blocks[j] >=Process[i]){
```

```
2       100     1
3       600     Not Allocated
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> & .\'first_fit.exe'
Enter # processes: 4
Enter # blocks: 5
Enter the process sizes
225
450
120
350
Enter the block sizes
100
500
200
300
600

P. No.  P. Size Block No.
1       225     2
2       450     5
3       120     2
4       350     Not Allocated
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output>
```

```
 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\first_fit.c -o c:\Deepan
kar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\first_fit.exe

 *  Terminal will be reused by tasks, press any key to close it.
```

```c
}
```



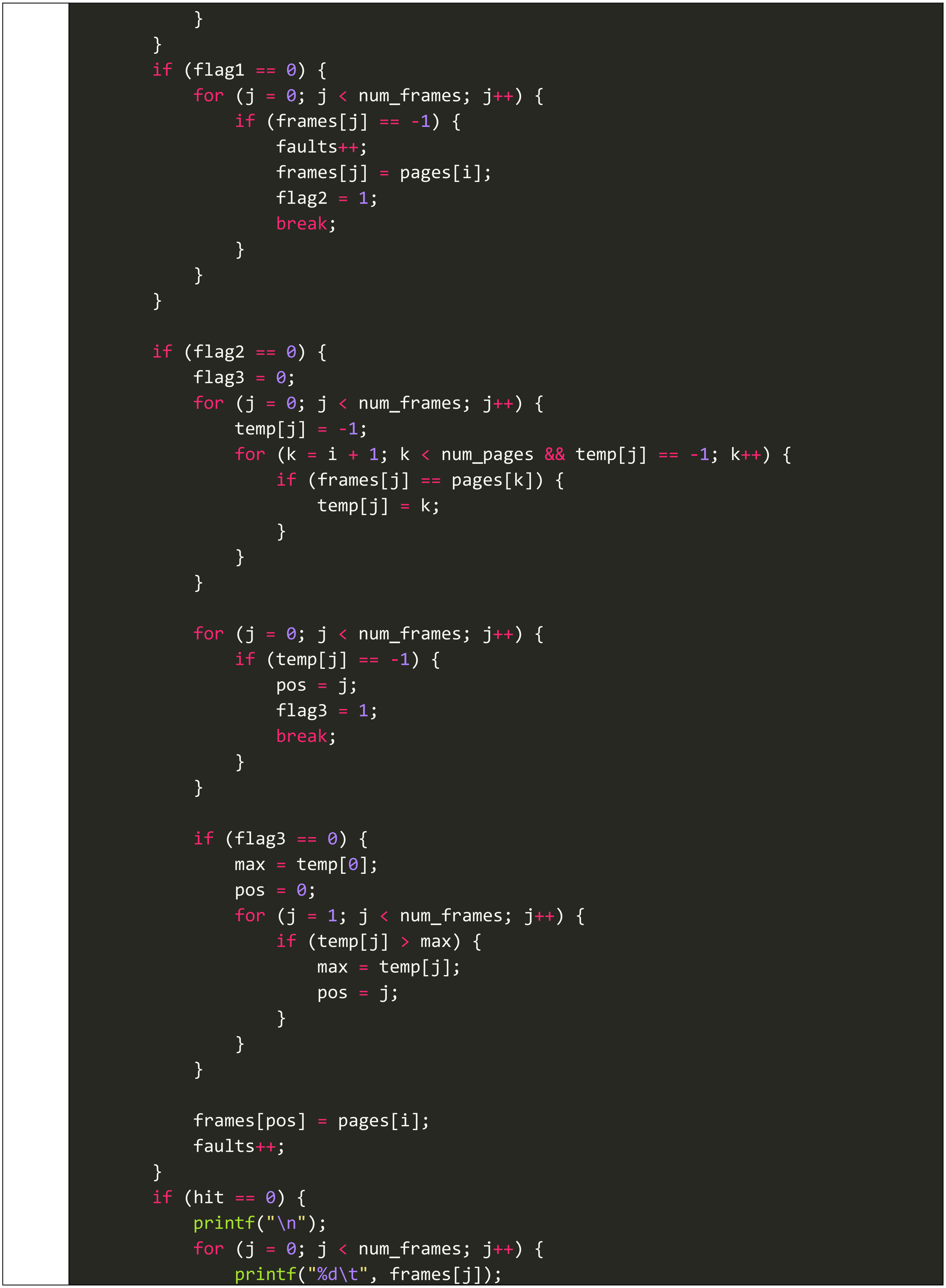
```c
# include <stdio.h>

void first_fit(int m, int n, int Blocks[], int Process[]){

    int i,j;

    int allocation[n];
    for ( i = 0; i < n; i++)
    {
        /* code */
        allocation[i]= -1;
    }

    for ( i = 0; i < n; i++) // # processes
    {
        /* code */
        for (  j = 0; j<m; j++) // # blocks
```

```
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> & .\'first_fit.exe'
Enter # processes: 3
Enter # blocks: 4
Enter the process sizes
250
100
600
Enter the block sizes
220
300
200
300

P. No.  P. Size Block No.
1       250     2
2       100     1
3       600     Not Allocated
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> 
```

```
 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\first_fit.c -o c:\Deepan
kar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\first_fit.exe

 *  Terminal will be reused by tasks, press any key to close it.
```

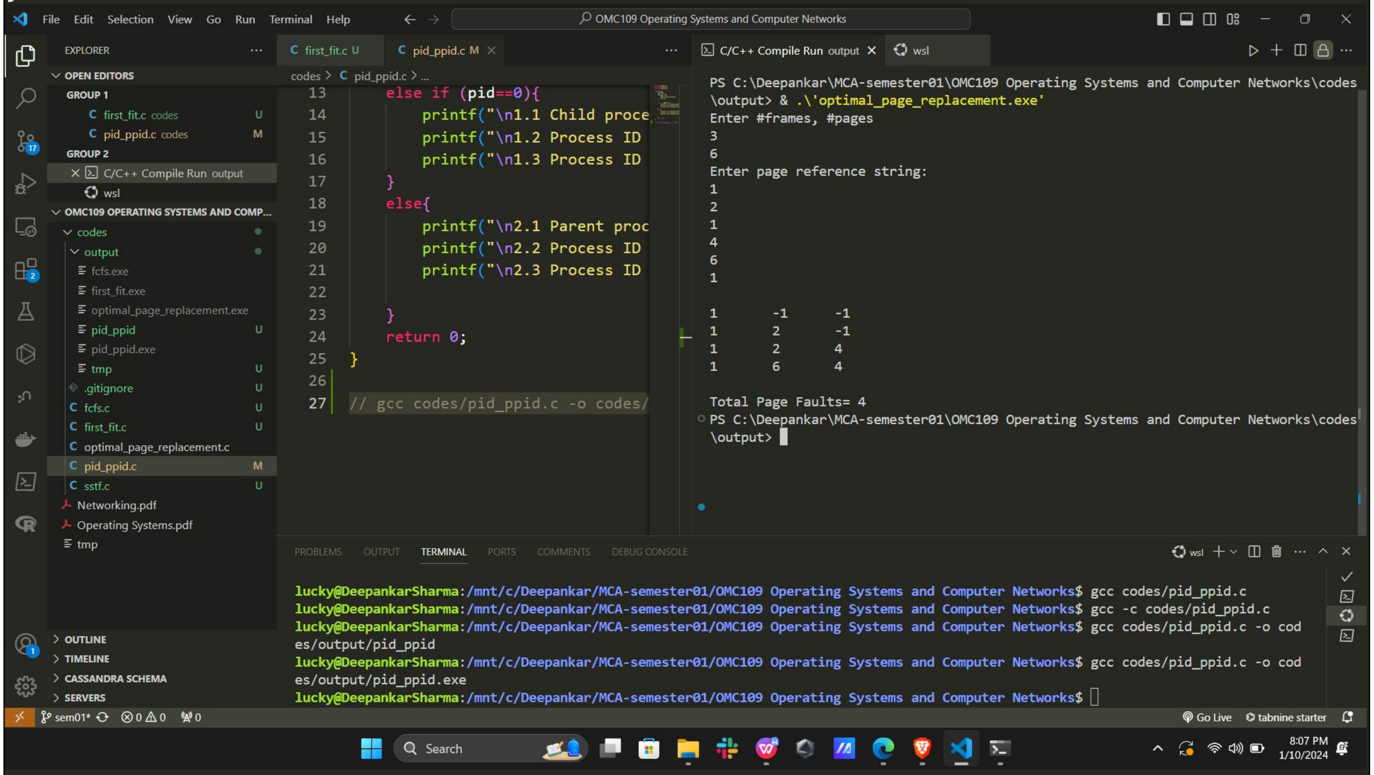
2 Write a C program to Implement the optimal page replacement algorithm

```c
#include<stdio.h>

int main() {
    int num_frames, num_pages, frames[10], pages[30], temp[10];
    int flag1, flag2, flag3, i, j, k, pos, max, faults=0, hit;

    printf("Enter #frames, #pages\n");
    scanf("%d%d", &num_frames, &num_pages);

    printf("Enter page reference string: \n");
    for (i = 0; i < num_pages; i++) {
        scanf("%d", &pages[i]);
    }

    for (i = 0; i < num_frames; i++) {
        frames[i] = -1;
    }

    for (i = 0; i < num_pages; i++) {
        flag1 = flag2 = 0;
        hit = 0;
        for (j = 0; j < num_frames; j++) {
            if (frames[j] == pages[i]) {
                flag1 = flag2 = 1;
                hit = 1;
                break;
```



```
            }
        }
        if (flag1 == 0) {
            for (j = 0; j < num_frames; j++) {
                if (frames[j] == -1) {
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }

        if (flag2 == 0) {
            flag3 = 0;
            for (j = 0; j < num_frames; j++) {
                temp[j] = -1;
                for (k = i + 1; k < num_pages && temp[j] == -1; k++) {
                    if (frames[j] == pages[k]) {
                        temp[j] = k;
                    }
                }
            }

            for (j = 0; j < num_frames; j++) {
                if (temp[j] == -1) {
                    pos = j;
                    flag3 = 1;
                    break;
                }
            }

            if (flag3 == 0) {
                max = temp[0];
                pos = 0;
                for (j = 1; j < num_frames; j++) {
                    if (temp[j] > max) {
                        max = temp[j];
                        pos = j;
                    }
                }
            }

            frames[pos] = pages[i];
            faults++;
        }
        if (hit == 0) {
            printf("\n");
            for (j = 0; j < num_frames; j++) {
                printf("%d\t", frames[j]);
```

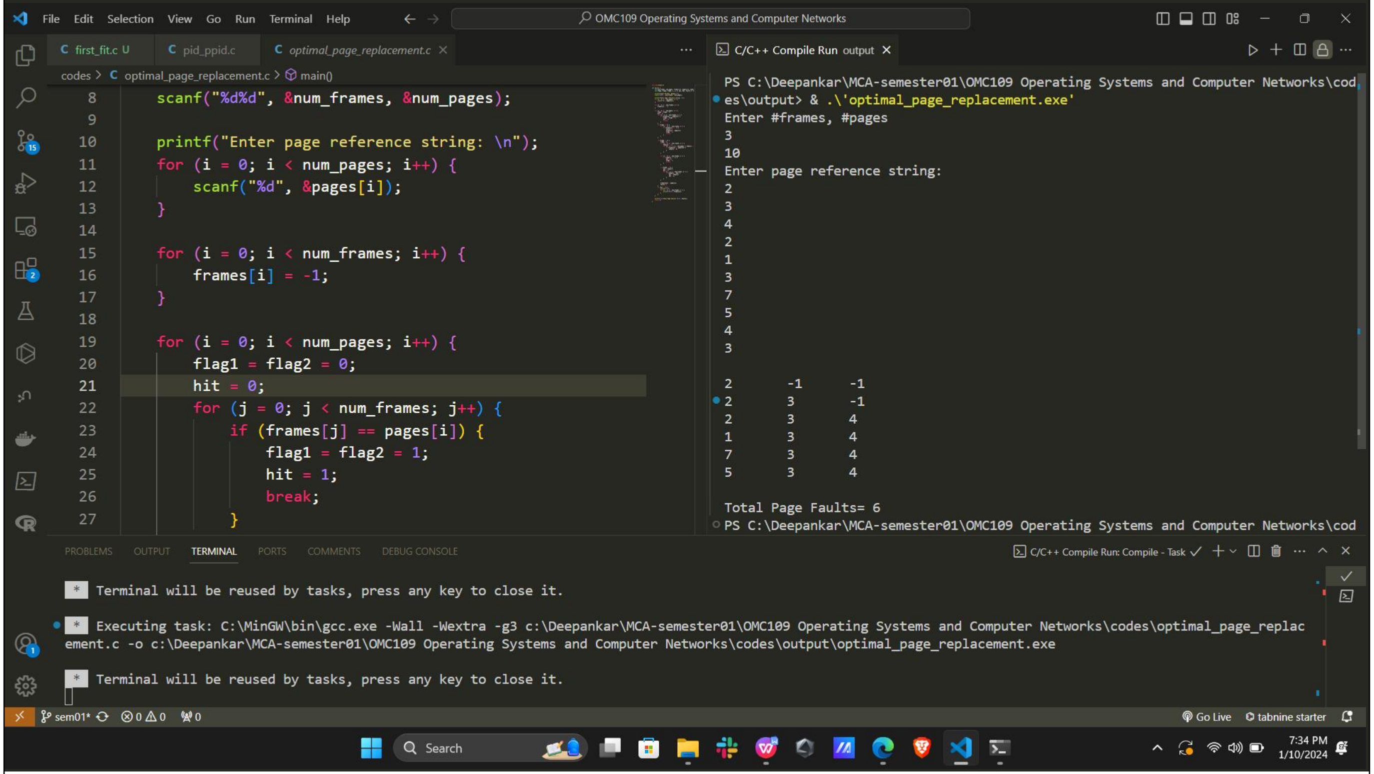
```c
            }
        }
    }

    printf("\n\nTotal Page Faults= %d\n", faults);
    return 0;
}
```



```c
    else if (pid==0){
        printf("\n1.1 Child proce
        printf("\n1.2 Process ID
        printf("\n1.3 Process ID
    }
    else{
        printf("\n2.1 Parent proc
        printf("\n2.2 Process ID
        printf("\n2.3 Process ID

    }
    return 0;
}

// gcc codes/pid_ppid.c -o codes/
```

```
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes
\output> & .\'optimal_page_replacement.exe'
Enter #frames, #pages
3
6
Enter page reference string:
1
2
1
4
6
1

1       -1      -1
1       2       -1
1       2       4
1       6       4

Total Page Faults= 4
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes
\output> 
```

```
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc -c codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid.exe
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ 
```

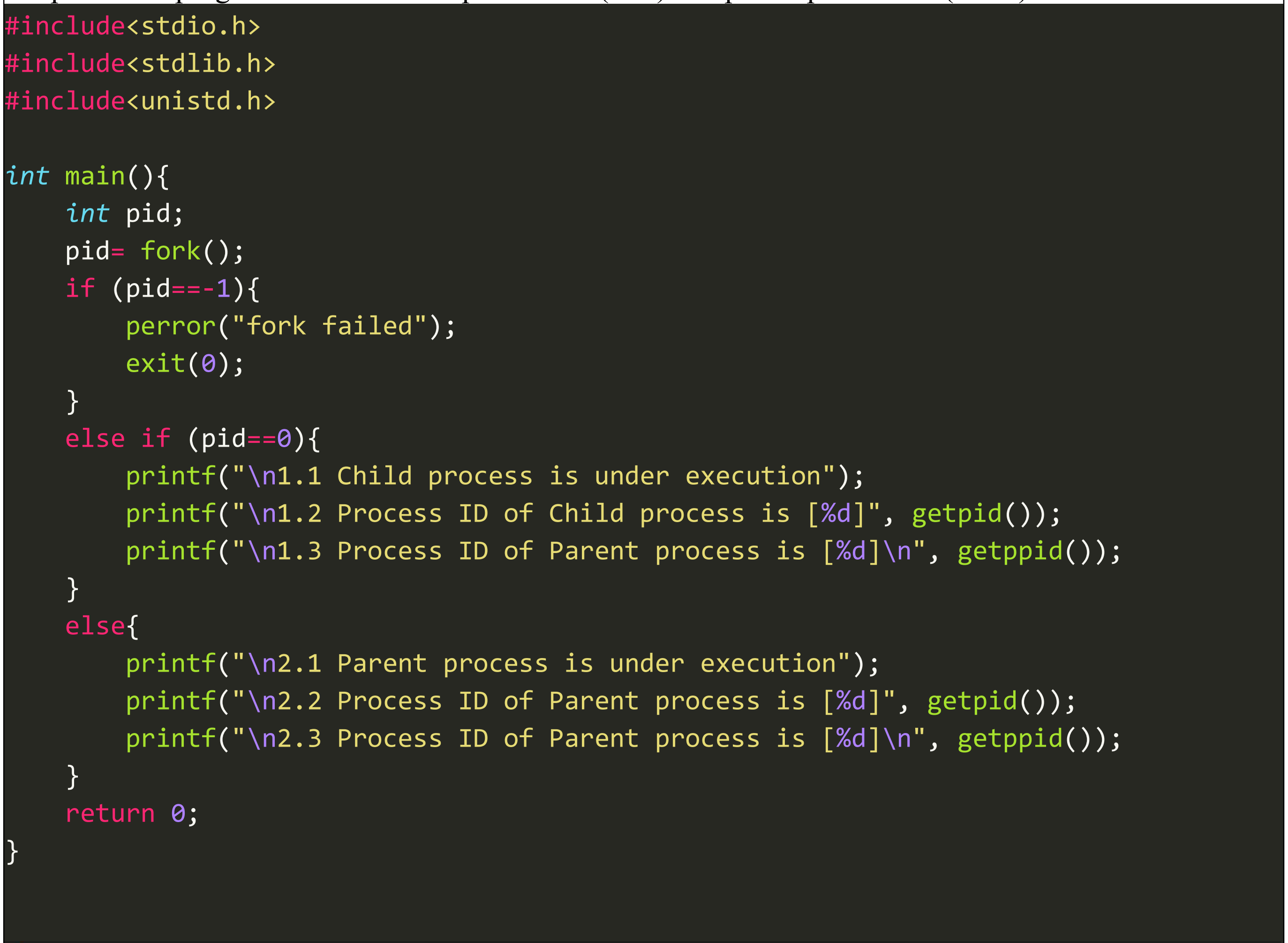

File Edit Selection View Go Run Terminal Help
OMC109 Operating Systems and Computer Networks
first_fit.c U
pid_ppid.c
optimal_page_replacement.c
codes > optimal_page_replacement.c > main()
C/C++ Compile Run output
PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
C/C++ Compile Run: Compile - Task
sem01* 0 0 0
Go Live
tabnine starter
Search
7:34 PM
1/10/2024

```c
    scanf("%d%d", &num_frames, &num_pages);

    printf("Enter page reference string: \n");
    for (i = 0; i < num_pages; i++) {
        scanf("%d", &pages[i]);
    }

    for (i = 0; i < num_frames; i++) {
        frames[i] = -1;
    }

    for (i = 0; i < num_pages; i++) {
        flag1 = flag2 = 0;
        hit = 0;
        for (j = 0; j < num_frames; j++) {
            if (frames[j] == pages[i]) {
                flag1 = flag2 = 1;
                hit = 1;
                break;
            }
```

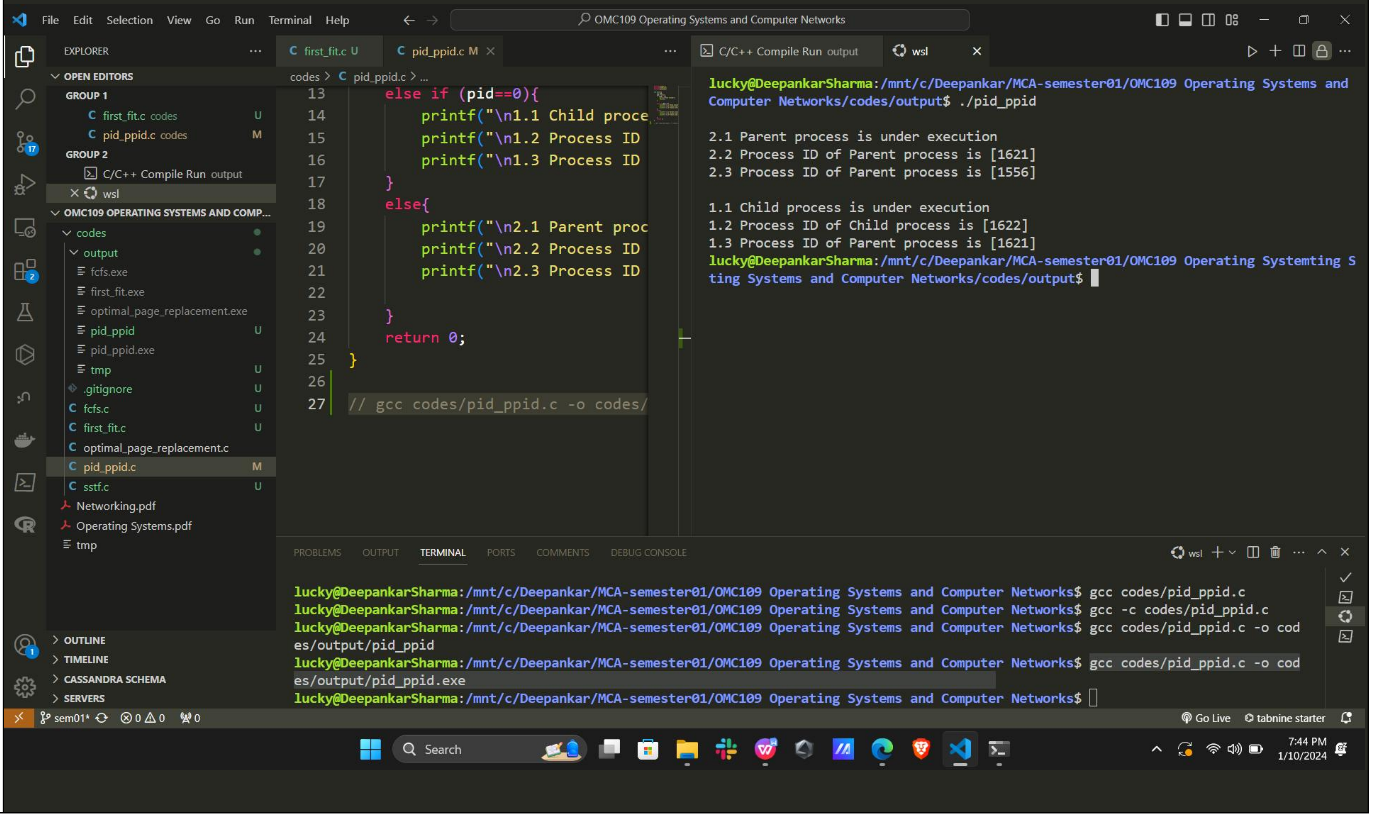
```
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> & .\'optimal_page_replacement.exe'
Enter #frames, #pages
3
10
Enter page reference string:
2
3
4
2
1
3
7
5
4
3

2       -1      -1
2       3       -1
2       3       4
1       3       4
7       3       4
5       3       4

Total Page Faults= 6
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
```

```
 *  Terminal will be reused by tasks, press any key to close it.

 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\optimal_page_replac
ement.c -o c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\optimal_page_replacement.exe

 *  Terminal will be reused by tasks, press any key to close it.
```

3 Implement a program in C to extract process ID (PID) and parent process ID (PPID)



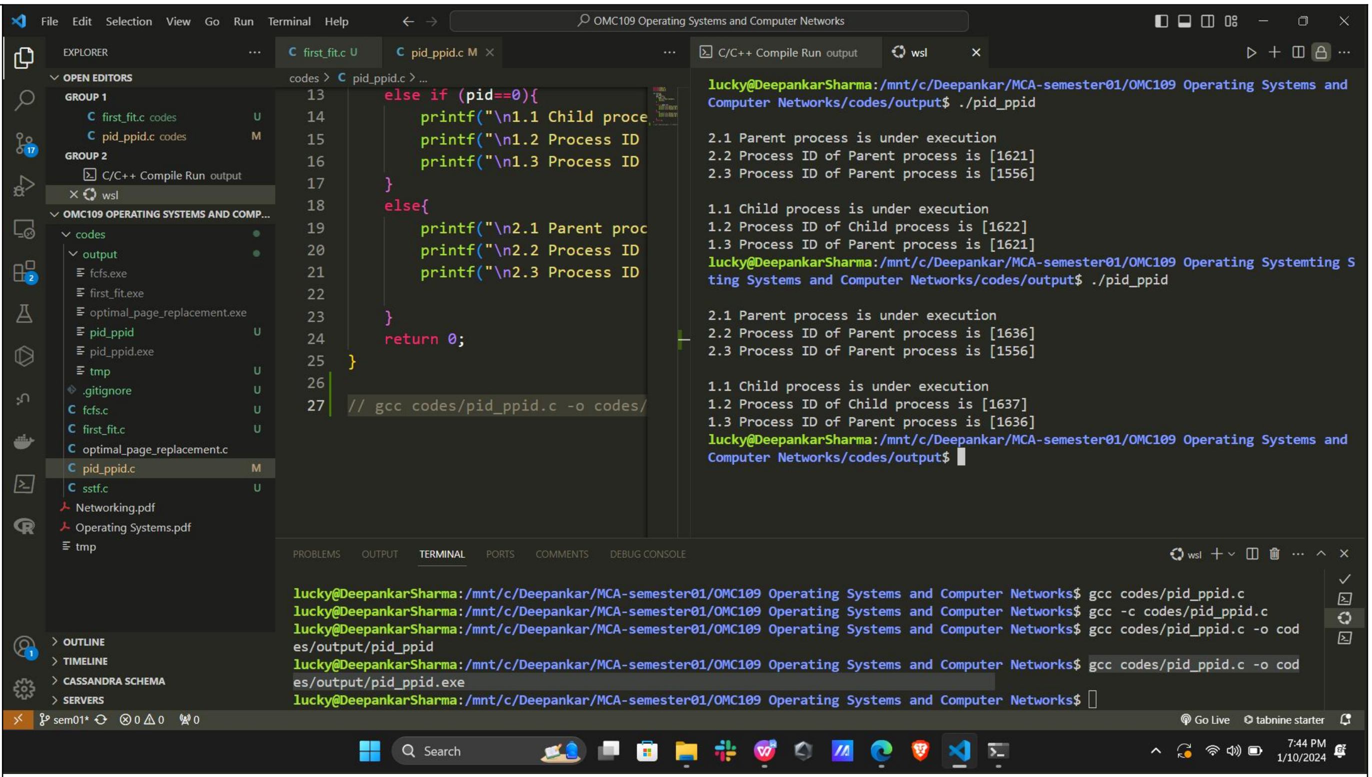
```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main(){
    int pid;
    pid= fork();
    if (pid==-1){
        perror("fork failed");
        exit(0);
    }
    else if (pid==0){
        printf("\n1.1 Child process is under execution");
        printf("\n1.2 Process ID of Child process is [%d]", getpid());
        printf("\n1.3 Process ID of Parent process is [%d]\n", getppid());
    }
    else{
        printf("\n2.1 Parent process is under execution");
        printf("\n2.2 Process ID of Parent process is [%d]", getpid());
        printf("\n2.3 Process ID of Parent process is [%d]\n", getppid());
    }
    return 0;
}
```



```c
    else if (pid==0){
        printf("\n1.1 Child proce
        printf("\n1.2 Process ID
        printf("\n1.3 Process ID
    }
    else{
        printf("\n2.1 Parent proc
        printf("\n2.2 Process ID
        printf("\n2.3 Process ID

    }
    return 0;
}

// gcc codes/pid_ppid.c -o codes/
```

```
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and
Computer Networks/codes/output$ ./pid_ppid

2.1 Parent process is under execution
2.2 Process ID of Parent process is [1621]
2.3 Process ID of Parent process is [1556]

1.1 Child process is under execution
1.2 Process ID of Child process is [1622]
1.3 Process ID of Parent process is [1621]
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systemting S
ting Systems and Computer Networks/codes/output$ 
```

```
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc -c codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid.exe
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ 
```



```c
    else if (pid==0){
        printf("\n1.1 Child proce
        printf("\n1.2 Process ID
        printf("\n1.3 Process ID
    }
    else{
        printf("\n2.1 Parent proc
        printf("\n2.2 Process ID
        printf("\n2.3 Process ID

    }
    return 0;
}

// gcc codes/pid_ppid.c -o codes/
```

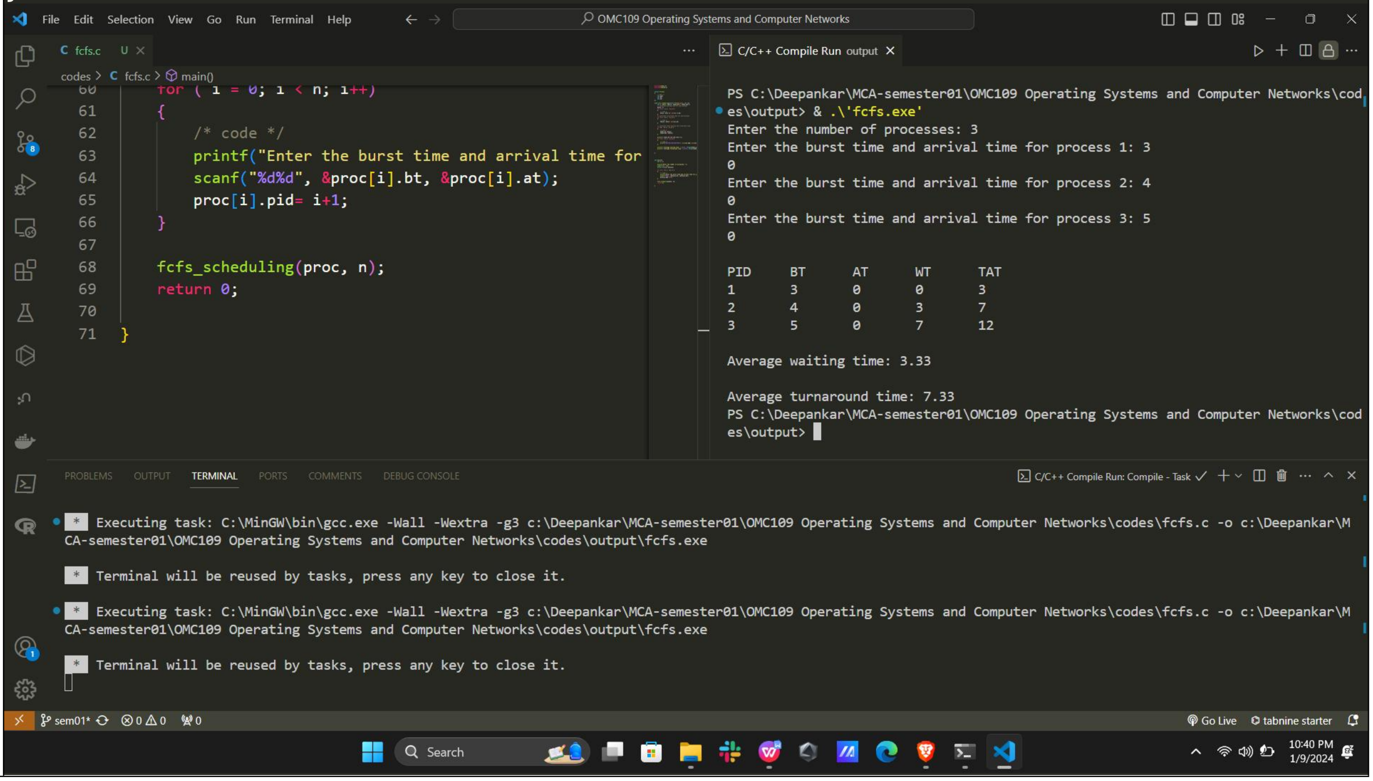
```
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and
Computer Networks/codes/output$ ./pid_ppid

2.1 Parent process is under execution
2.2 Process ID of Parent process is [1621]
2.3 Process ID of Parent process is [1556]

1.1 Child process is under execution
1.2 Process ID of Child process is [1622]
1.3 Process ID of Parent process is [1621]
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systemting S
ting Systems and Computer Networks/codes/output$ ./pid_ppid

2.1 Parent process is under execution
2.2 Process ID of Parent process is [1636]
2.3 Process ID of Parent process is [1556]

1.1 Child process is under execution
1.2 Process ID of Child process is [1637]
1.3 Process ID of Parent process is [1636]
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and
Computer Networks/codes/output$
```

```
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc -c codes/pid_ppid.c
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$ gcc codes/pid_ppid.c -o cod
es/output/pid_ppid.exe
lucky@DeepankarSharma:/mnt/c/Deepankar/MCA-semester01/OMC109 Operating Systems and Computer Networks$
```

4

Simulate the following CPU scheduling algorithms-FCFS

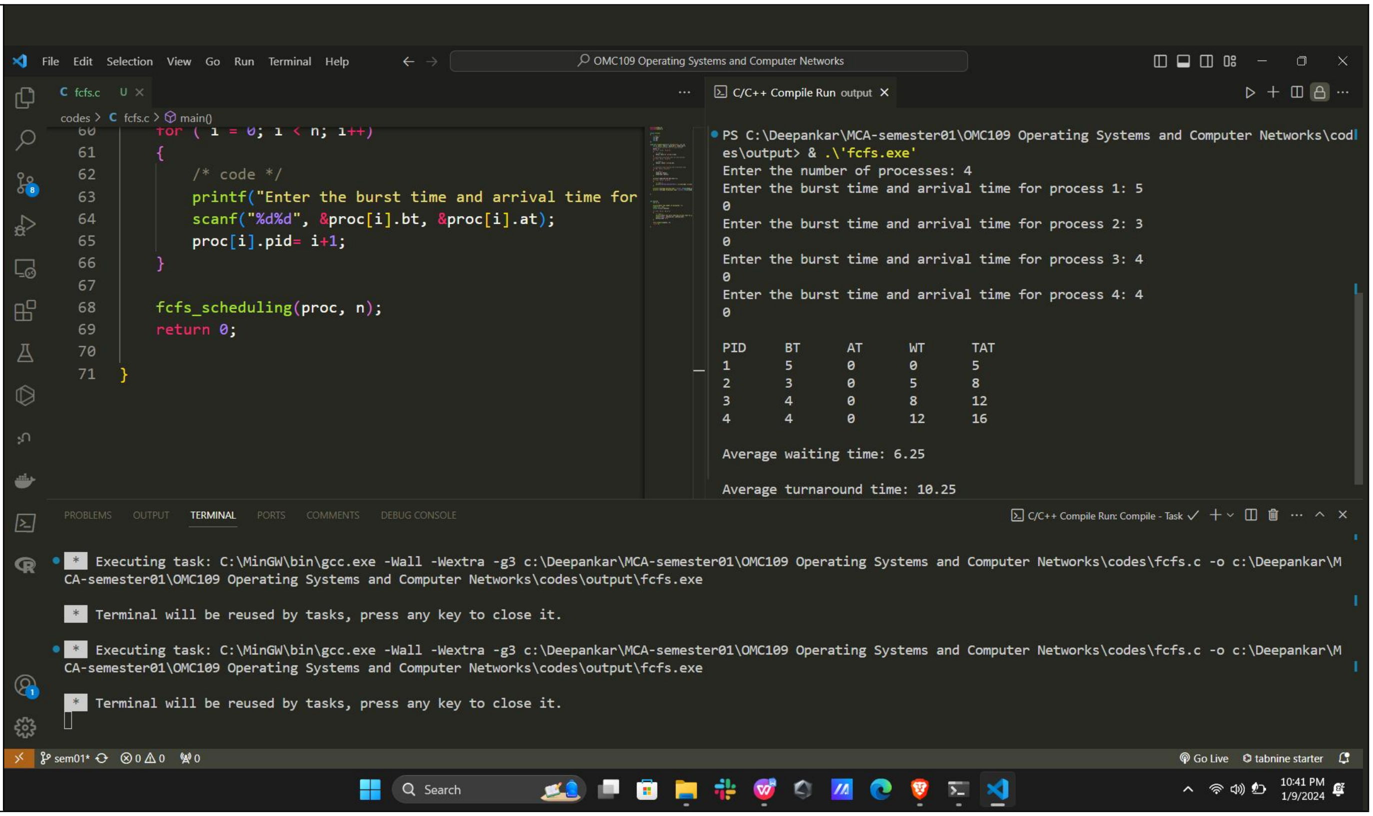
```c
#include<stdio.h>
#include<stdlib.h>


struct Process
{
    /* data */
    int pid;
    int bt;
    int at;
};

void fcfs_scheduling(struct Process*proc, int n){
    int i, wt[n], tat[n], total_wt=0, total_tat=0;
    // calculate waiting time for each process
    wt[0]= 0;
    for ( i = 1; i < n; i++)
    {
        /* code */
        wt[i]= wt[i-1]+ proc[i-1].bt;
    }
    // calculate turnaround time for each process
    for ( i = 0; i < n; i++)
    {
        /* code */
        tat[i]= wt[i]+ proc[i].bt;
    }
    // calculate total waiting and turnaround time
    for (i = 0; i < n; i++)
    {
        /* code */
        total_wt+=wt[i];
        total_tat+=tat[i];
    }
    printf("\nPID\tBT\tAT\tWT\tTAT\n");
    for ( i = 0; i < n; i++)
    {
        /* code */
        printf("%d\t%d\t%d\t%d\t%d\n", proc[i].pid, proc[i].bt, proc[i].at, wt[i],
tat[i]);
    }
    printf("\nAverage waiting time: %.2f\n", (float)total_wt/n);
    printf("\nAverage turnaround time: %.2f\n", (float)total_tat/n);
}
int main(){
    int n, i;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    struct Process proc[n];
    for ( i = 0; i < n; i++)
    {
        /* code */
        printf("Enter the burst time and arrival time for process %d: ", i+1);
        scanf("%d%d", &proc[i].bt, &proc[i].at);
        proc[i].pid= i+1;
    }
    fcfs_scheduling(proc, n);
    return 0;
}
```



```
    for ( i = 0; i < n; i++)
    {
        /* code */
        printf("Enter the burst time and arrival time for
        scanf("%d%d", &proc[i].bt, &proc[i].at);
        proc[i].pid= i+1;
    }

    fcfs_scheduling(proc, n);
    return 0;

}
```

```
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> & .\'fcfs.exe'
Enter the number of processes: 3
Enter the burst time and arrival time for process 1: 3
0
Enter the burst time and arrival time for process 2: 4
0
Enter the burst time and arrival time for process 3: 5
0

PID     BT      AT      WT      TAT
1       3       0       0       3
2       4       0       3       7
3       5       0       7       12

Average waiting time: 3.33

Average turnaround time: 7.33
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output>
```

```
 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\fcfs.c -o c:\Deepankar\M
CA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\fcfs.exe

 *  Terminal will be reused by tasks, press any key to close it.

 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\fcfs.c -o c:\Deepankar\M
CA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\fcfs.exe

 *  Terminal will be reused by tasks, press any key to close it.
```


File Edit Selection View Go Run Terminal Help
OMC109 Operating Systems and Computer Networks
fcfs.c U
codes > fcfs.c > main()
C/C++ Compile Run output
PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
C/C++ Compile Run: Compile - Task
sem01* 0 0 0
Go Live tabnine starter
Search
10:41 PM
1/9/2024

```c
    for ( i = 0; i < n; i++)
    {
        /* code */
        printf("Enter the burst time and arrival time for
        scanf("%d%d", &proc[i].bt, &proc[i].at);
        proc[i].pid= i+1;
    }

    fcfs_scheduling(proc, n);
    return 0;

}
```

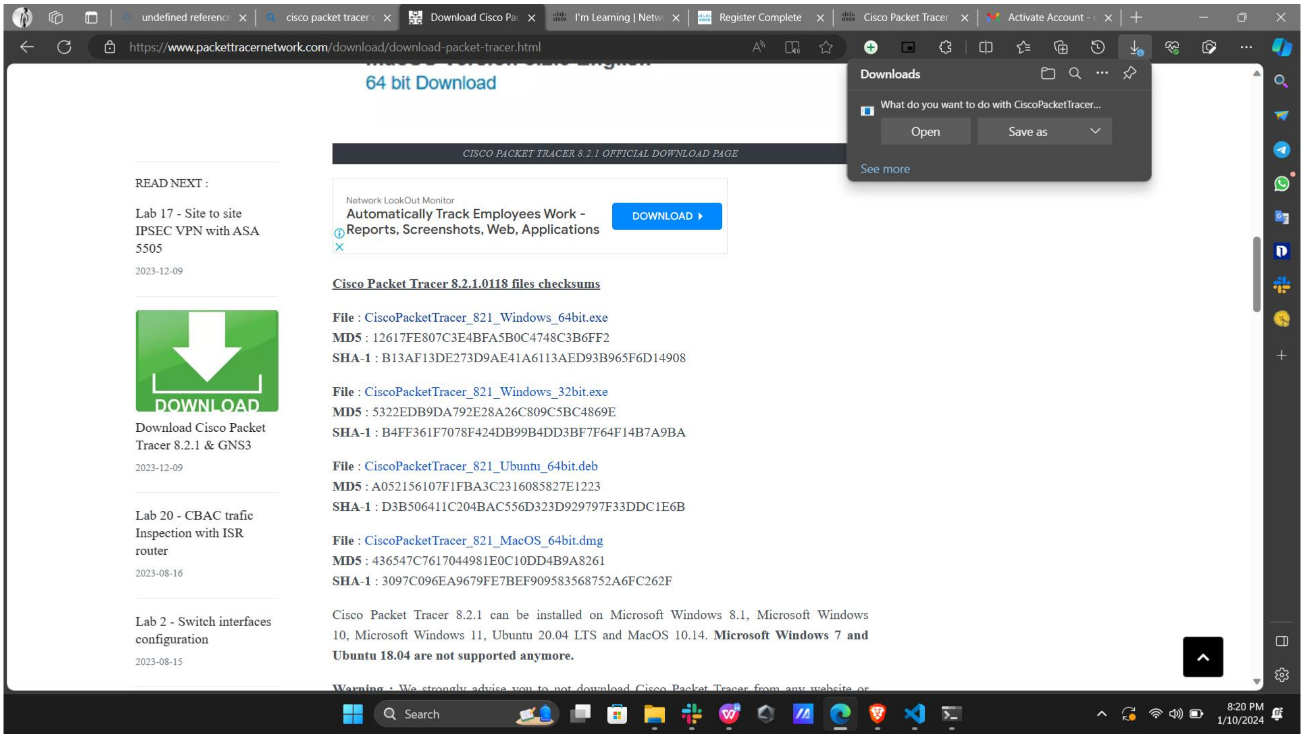
```
PS C:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\cod
es\output> & .\'fcfs.exe'
Enter the number of processes: 4
Enter the burst time and arrival time for process 1: 5
0
Enter the burst time and arrival time for process 2: 3
0
Enter the burst time and arrival time for process 3: 4
0
Enter the burst time and arrival time for process 4: 4
0

PID     BT      AT      WT      TAT
1       5       0       0       5
2       3       0       5       8
3       4       0       8       12
4       4       0       12      16

Average waiting time: 6.25

Average turnaround time: 10.25
```

```
 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\fcfs.c -o c:\Deepankar\M
CA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\fcfs.exe

 *  Terminal will be reused by tasks, press any key to close it.

 *  Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 c:\Deepankar\MCA-semester01\OMC109 Operating Systems and Computer Networks\codes\fcfs.c -o c:\Deepankar\M
CA-semester01\OMC109 Operating Systems and Computer Networks\codes\output\fcfs.exe

 *  Terminal will be reused by tasks, press any key to close it.
```

Compulsory question:

Explain the installation steps for Cisco Packet Tracer, and include snapshotsfor clarification.

1. Download the latest version of Cisco Packet Tracer from the official website or from your instructor.



64 bit Download

CISCO PACKET TRACER 8.2.1 OFFICIAL DOWNLOAD PAGE

READ NEXT :

Lab 17 - Site to site IPSEC VPN with ASA 5505
2023-12-09

Download Cisco Packet Tracer 8.2.1 & GNS3
2023-12-09

Lab 20 - CBAC trafic Inspection with ISR router
2023-08-16

Lab 2 - Switch interfaces configuration
2023-08-15

Cisco Packet Tracer 8.2.1.0118 files checksums

File : CiscoPacketTracer_821_Windows_64bit.exe
MD5 : 12617FE807C3E4BFA5B0C4748C3B6FF2
SHA-1 : B13AF13DE273D9AE41A6113AED93B965F6D14908

File : CiscoPacketTracer_821_Windows_32bit.exe
MD5 : 5322EDB9DA792E28A26C809C5BC4869E
SHA-1 : B4FF361F7078F424DB99B4DD3BF7F64F14B7A9BA

File : CiscoPacketTracer_821_Ubuntu_64bit.deb
MD5 : A052156107F1FBA3C2316085827E1223
SHA-1 : D3B506411C204BAC556D323D929797F33DDC1E6B

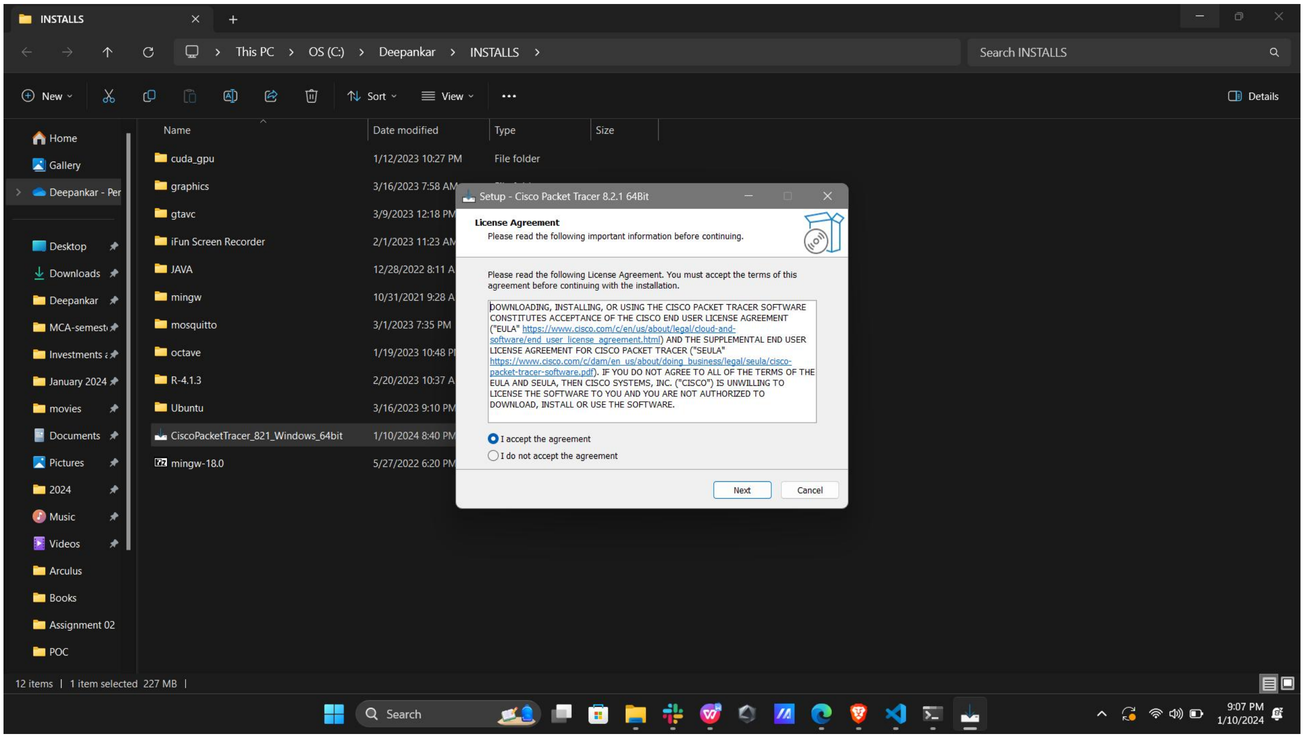
File : CiscoPacketTracer_821_MacOS_64bit.dmg
MD5 : 436547C7617044981E0C10DD4B9A8261
SHA-1 : 3097C096EA9679FE7BEF909583568752A6FC262F

Cisco Packet Tracer 8.2.1 can be installed on Microsoft Windows 8.1, Microsoft Windows 10, Microsoft Windows 11, Ubuntu 20.04 LTS and MacOS 10.14. **Microsoft Windows 7 and Ubuntu 18.04 are not supported anymore.**

Warning : We strongly advise you to not download Cisco Packet Tracer from any website or

2. Run the installer file and accept the license

agreement.



License Agreement

Please read the following important information before continuing.

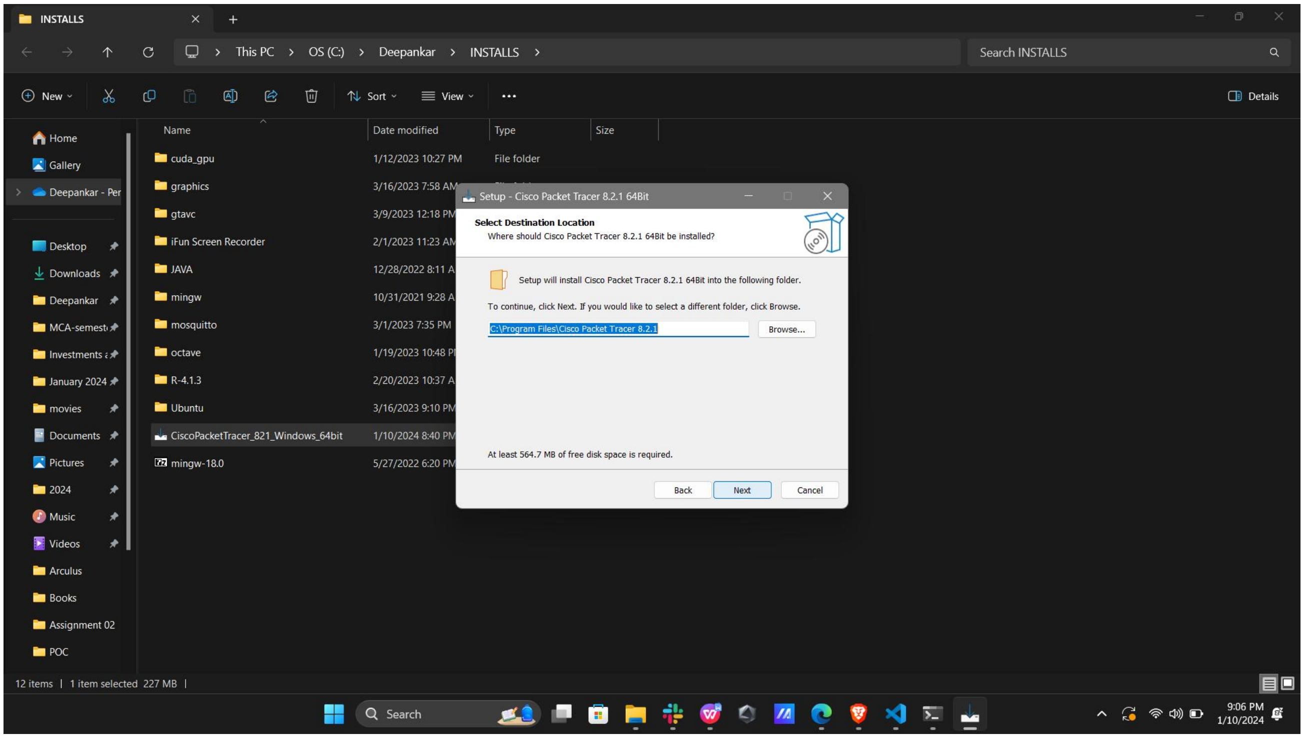
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

DOWNLOADING, INSTALLING, OR USING THE CISCO PACKET TRACER SOFTWARE CONSTITUTES ACCEPTANCE OF THE CISCO END USER LICENSE AGREEMENT ("EULA" https://www.cisco.com/c/en/us/about/legal/cloud-and-software/end_user_license_agreement.html) AND THE SUPPLEMENTAL END USER LICENSE AGREEMENT FOR CISCO PACKET TRACER ("SEULA" https://www.cisco.com/c/dam/en_us/about/doing_business/legal/seula/cisco-packet-tracer-software.pdf). IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THE EULA AND SEULA, THEN CISCO SYSTEMS, INC. ("CISCO") IS UNWILLING TO LICENSE THE SOFTWARE TO YOU AND YOU ARE NOT AUTHORIZED TO DOWNLOAD, INSTALL OR USE THE SOFTWARE.

● I accept the agreement

○ I do not accept the agreement

3. Choose the installation location and the components you want to install. You can also customize the shortcuts and associations.



Select Destination Location

Where should Cisco Packet Tracer 8.2.1 64Bit be installed?

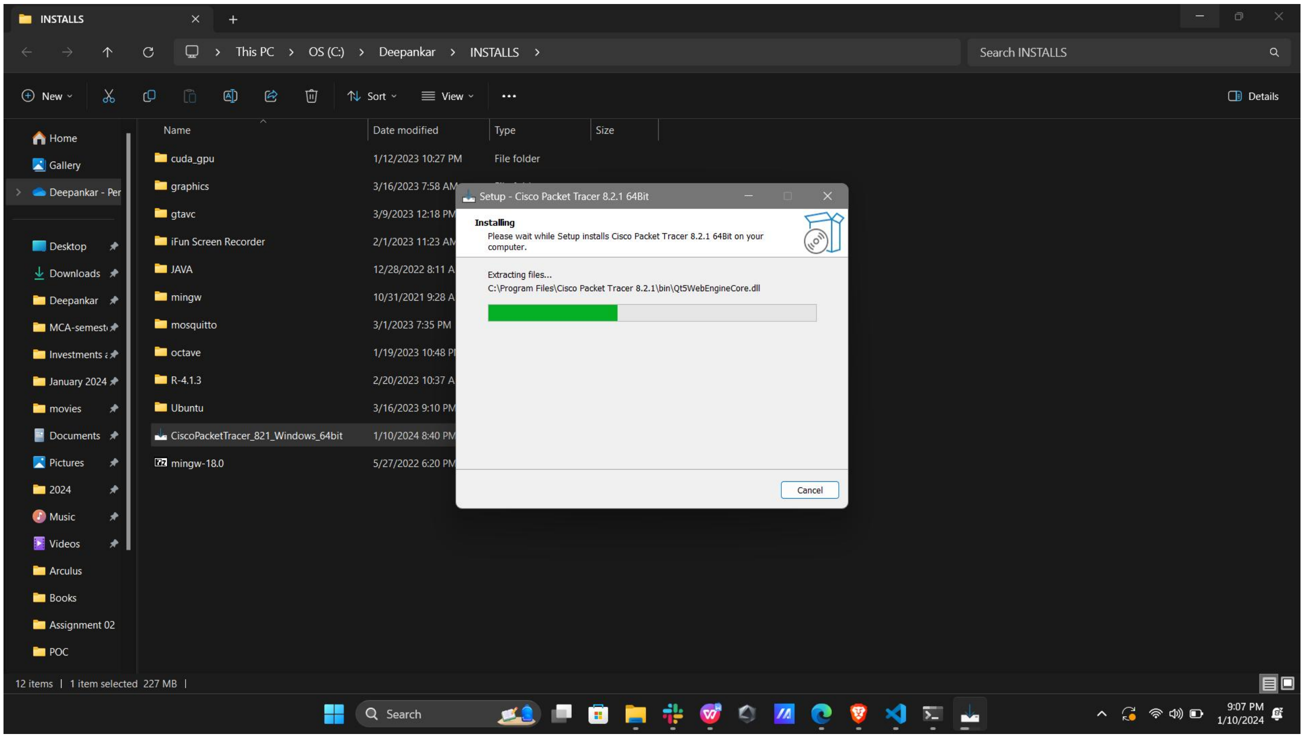
Setup will install Cisco Packet Tracer 8.2.1 64Bit into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Cisco Packet Tracer 8.2.1

At least 564.7 MB of free disk space is required.

4. Click on Install and wait for the installation to complete.



Name	Date modified	Type	Size
cuda_gpu	1/12/2023 10:27 PM	File folder	
graphics	3/16/2023 7:58 AM		
gtavc	3/9/2023 12:18 PM		
iFun Screen Recorder	2/1/2023 11:23 AM		
JAVA	12/28/2022 8:11 A		
mingw	10/31/2021 9:28 A		
mosquitto	3/1/2023 7:35 PM		
octave	1/19/2023 10:48 P		
R-4.1.3	2/20/2023 10:37 A		
Ubuntu	3/16/2023 9:10 PM		
CiscoPacketTracer_821_Windows_64bit	1/10/2024 8:40 PM		
mingw-18.0	5/27/2022 6:20 PM		

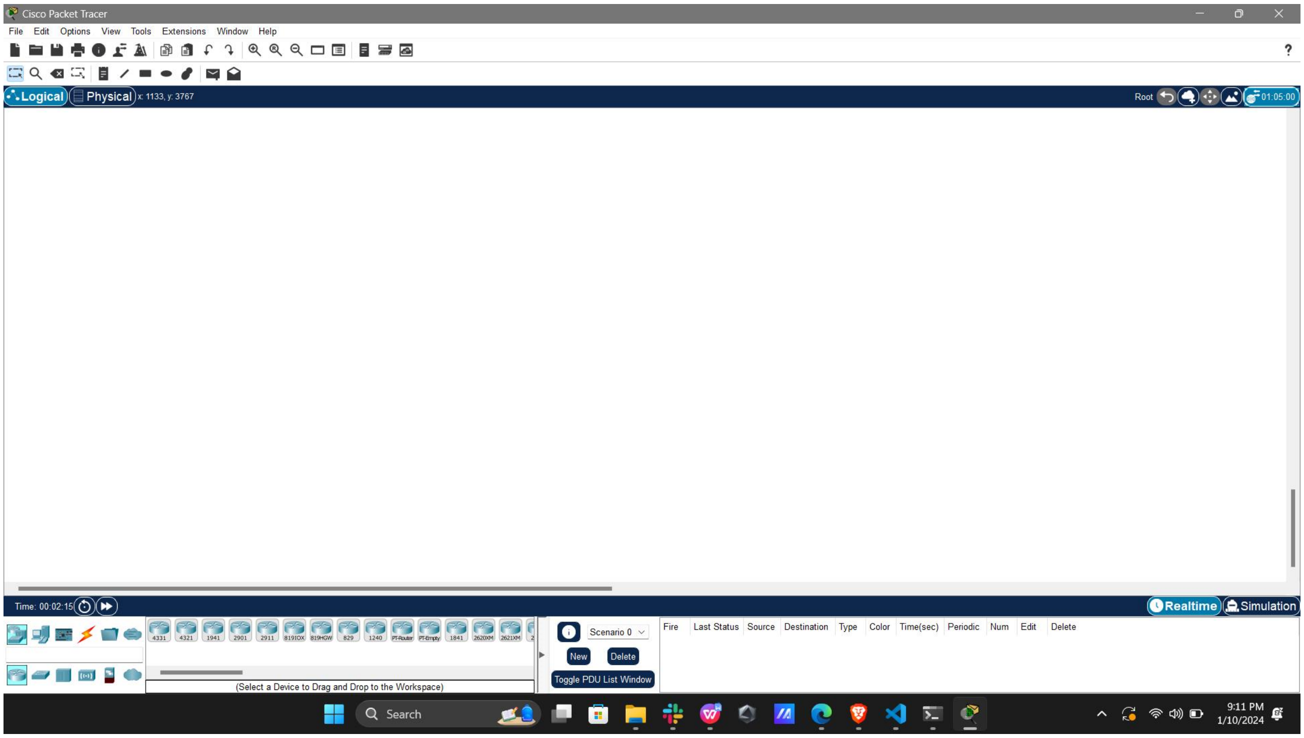
Setup - Cisco Packet Tracer 8.2.1 64Bit

Installing

Please wait while Setup installs Cisco Packet Tracer 8.2.1 64Bit on your computer.

Extracting files...
C:\Program Files\Cisco Packet Tracer 8.2.1\bin\Qt5WebEngineCore.dll

Cancel
12 items | 1 item selected 227 MB |
Search
9:07 PM
1/10/2024


Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical
Physical
x: 1133, y: 3767
Root
01:05:00
Time: 00:02:15
Realtime
Simulation
4331 4321 1941 2901 2911 819IOX 819HGW 829 1240 PT-Router PT-Empty 1841 2620XM 2621XM
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New
Delete
Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete