Question – 10

Write a program to solve the weighted interval scheduling problem .

Code :

```
#include<iostream>

#include<algorithm>

using namespace std;


int computepj(int i,int S[],int FT[])

{

   int c=0;


        for(int j=1;j<i;j++)

        {


                if(FT[j]<=S[i])

                {

                  c= j ;

                }


        }
```

```
        return c;
}


int Computeopt(int j,int S[],int FT[],int Wt[])
{
        if(j==0)
        return 0;

        else
         return
                max(Wt[j]+Computeopt(computepj(j,S,FT),S,FT,Wt),
                Computeopt(j-1,S,FT,Wt));
}


int main()
{

int num;
cout<<"Enter the number of requests : ";
```

```
cin>>num;

int J[num];    // job no.
int S[num];   // start time
int FT[num];    // finish time
int Wt[num];      // weight
int t;
cout<<"\nEnter the job names : \n";

for(int i=1;i<=num;i++)
{
cin>>J[i];
}


cout<<"\nEnter the start time  ,finish time and weight for each request : \n";
for(int i=1;i<=num;i++)
{
cout<<"\nFor "<<i<<"\n ST\t";
cin>>S[i];

cout<<"\n FT\t";
```

```
cin>>FT[i];
cout<<"\n WT\t";
cin>>Wt[i];
}
cout<<"\nEntered requests\n";
cout<<"Job\t Start_time\t finish_time\t weight\n";
for(int i=1;i<=num;i++)
{
cout<<J[i]<<"\t\t"<<S[i]<<"\t\t"<<FT[i]<<"\t\t"<<Wt[i]<<"\t\t\n";
}

bool flag = true ;
//to sort the jobs in the order of increasing finish time
for(int i=1;i<=num;i++)
{
for(int j=1;j<=num-i;j++)
{

 if(FT[j]>FT[j+1])
 {
        flag = false ;
 t=FT[j];
```

```
FT[j]=FT[j+1];
FT[j+1]=t;

t=S[j];
S[j]=S[j+1];
S[j+1]=t;

t=J[j];
J[j]=J[j+1];
J[j+1]=t;

t=Wt[j];
Wt[j]=Wt[j+1];
Wt[j+1]=t;
}


}
}

int pj;
```

```
if(flag == false)
  // will execute only if the we have sorted the data according to finish time
{

        cout<<"After sorting\n";
        cout<<"Job\t Start_time\t finish_time\t weight\n";
        for(int i=1;i<=num;i++)
        {
        cout<<J[i]<<"\t\t"<<S[i]<<"\t\t"<<FT[i]<<"\t\t"<<Wt[i]<<"\t\t\n";
        }
}

cout<<"\n\n";
for(int i=1;i<=num;i++)
{
        cout<<"p("<<i<<")\t";
        pj=computepj(i,S,FT);
        cout<<pj<<"\n";
}

cout<<"\nOptimal value "<<Computeopt(num,S,FT,Wt);
```

return 0 ;

}

Output :

```
Enter the number of requests : 4

Enter the job names :
1 2 3 4

Enter the start time  ,finish time and weight for each request :

For 1
 ST     1

 FT     2

 WT     50

For 2
 ST     3

 FT     5

 WT     20

For 3
 ST     6

 FT     19

 WT     100

For 4
 ST     2

 FT     100

 WT     200
```

```
Entered requests
Job      Start_time      finish_time     weight
1                1                2                50
2                3                5                20
3                6                19               100
4                2                100              200


p(1)    0
p(2)    1
p(3)    2
p(4)    1

Optimal value 250
```