

Job Sequencing with deadlines(Greedy approach)

```
#include<stdio.h>
#include<conio.h>
int jobsequence(int d[10],int j[10],int p[10],int n)
{
int q,i,r,k,profit=0;
d[0]=0;
j[0]=0;
j[1]=1;
k=1;
profit=p[1];
for(i=2;i<=n;i++)
{
r=k;
while((d[j[r]]>d[i]) &&(d[j[r]]!=r))
r=r-1;
if((d[j[r]]<=d[i]) && (d[i]>r))
{
for(q=k;q>=r+1;q--)
{
j[q+1]=j[q];
}
j[r+1]=i;
profit=profit+p[i];
k=k+1;
}
}
printf("\n\n INDEX PROFIT DEADLINE SLOT ALLOTTED ");
printf("Solution vector = \n {");
for(i=1;i<=k;i++)
{
printf("%d, ",j[i]);
}
printf("} \n");
return profit;
}
int main()
{
int n,i,t;
printf("enter the no of jobs : ");
scanf("%d",&n);
int p[10],d[10],k,j[10],profit;
for(i=1;i<=n;i++)
{printf("\n enter the profit of job # %d : ",i+1);
scanf("%d",&p[i]);
printf("\n enter the deadline of job # %d : ",i+1);
scanf("%d",&d[i]);
```

```

    }
//sorting the jobs in the order of profits
for(i=1;i<=n;i++)
for(k=i+1;k<=n;k++)
if(p[i]<p[k])
{ t=p[i];
p[i]=p[k];
p[k]=t;
t=d[i];
d[i]=d[k];
d[k]=t;
}

profit=jobsequence(d,j,p,n);
printf("solution completed with %d profit",profit);
}

```

Sample Output

```

enter the no of jobs : 5
enter the profit of job #2 :20
enter the deadline of job #2 :2
enter the profit of job #3 :15
enter the deadline of job #3 :2
enter the profit of job #4 :10
enter the deadline of job #4 :1
enter the profit of job #5 :5
enter the deadline of job #5 :3
enter the profit of job #6 :1
enter the deadline of job #6 :3

INDEX PROFIT DEADLINE SLOT ALLOTTED Solution vector =
{1, 2, 4, }
solution completed with 40 profit
-----
Process exited after 13.01 seconds with return value 0
Press any key to continue ...

```