LAB-4

DOUBLE ENDED QUEUE

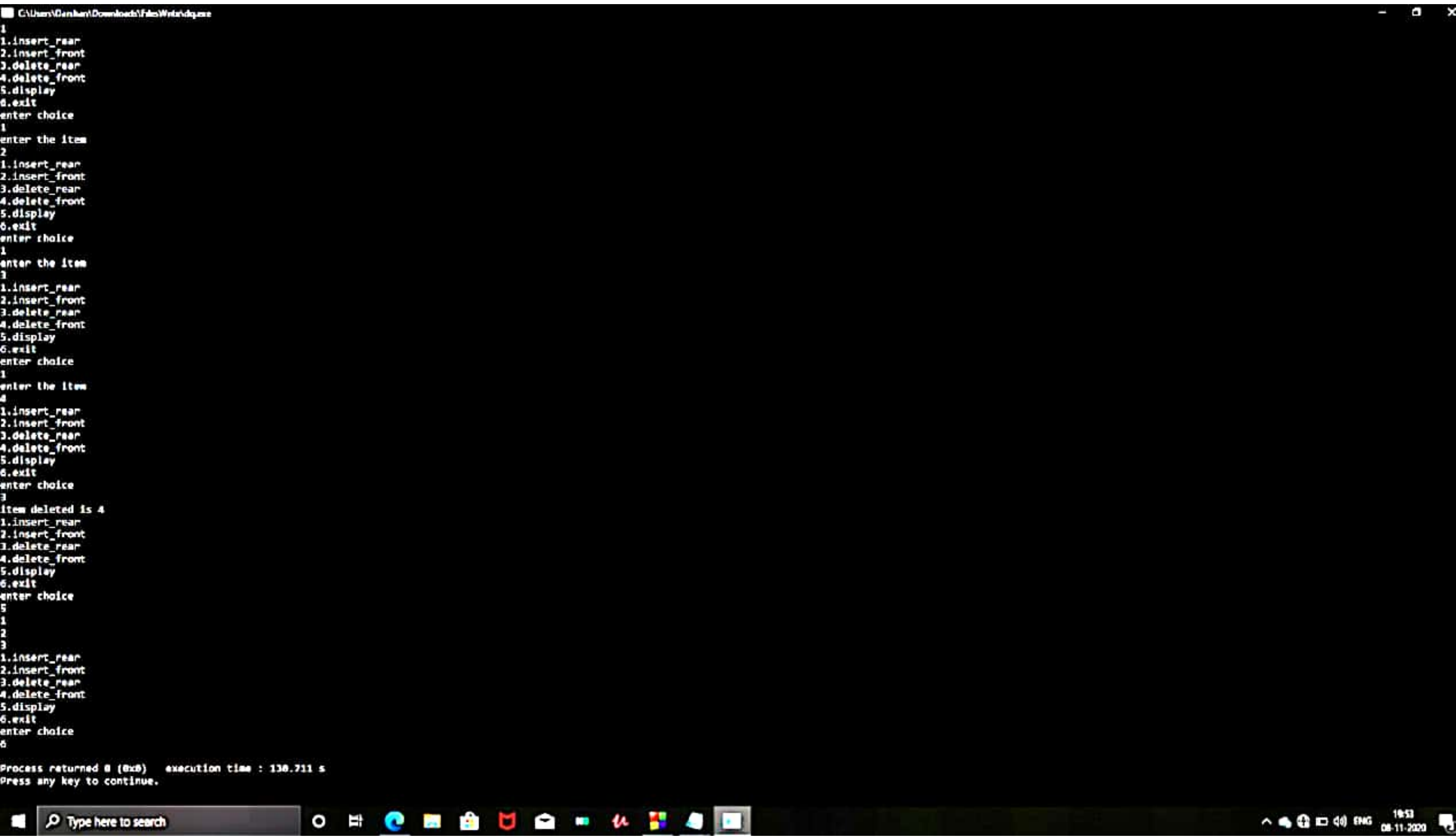
```
#include < stdio.h >
~~#define q size 5~~
#include < stdlib.h >
#define q size 5
int f=0, r=-1, ch;
int item, q[10];
int is full()
{
 return (r==q size-1)?1:0;
}
 int is empty()
 {
  return(f>r)?1:0;
 }
void insert_rear()
{
 if(is full())
  {
   printf(" queue overflow\n");
        retun;
      }
```

```
r = r+1,
q[r] = item,
}
voide delete_front()
{
 if (is empty())
 {
  printf (" queue empty \n");
   return;
  }
printf ("item deleted is %d \n", q[(f)++]);
if (f > r)
  {
   f=0;
    r = -1;
  }
}
void insert_front()
 {
   if (f != 0)
   {
    f = f-1
    q[f] = item;
     return,
    }
```

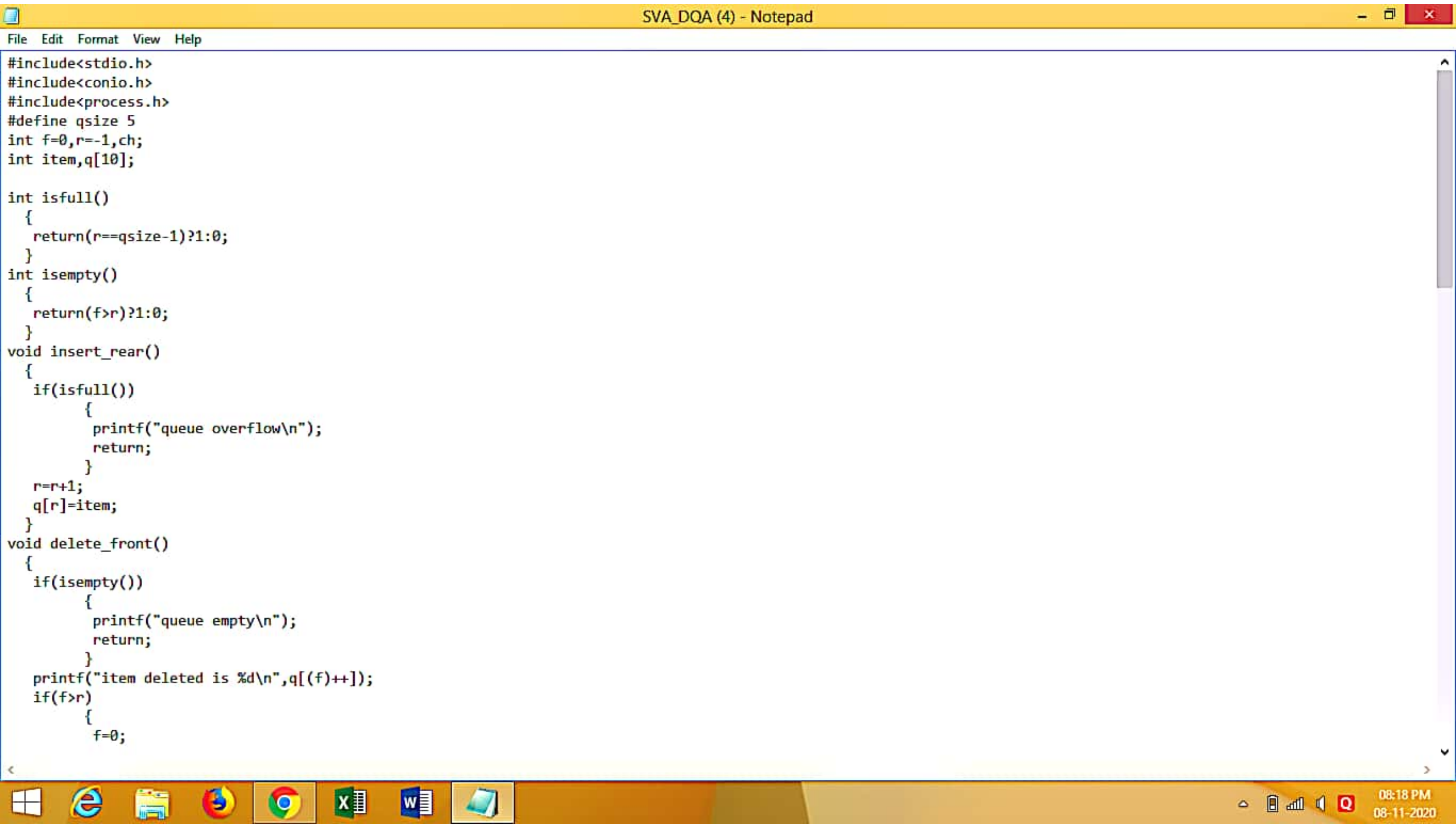
```
else if ((f ==0) && (r ==-1))
{
   q [++(r) ] = item;
   return;
 }
else
   printf ("insertion not possible\n");
}
void delete_rear()
{
  if (is empty())
    {
     printf ("queue is empty\n");
      return;
    }
printf ("item deleted is %d \n", q [(r)--]);
if (f>r)
    {
     f =0;
     r=-1;
    }
 }
  void display()
 {
   int i;
```

```
if (is empty())
 {
   printf(" queue empty \n");
      return;
 }
for(i=f; i<=r; i++)
      printf("%d", q[i]);
  }
  void  main()
  for(;;)

printf("1.insert_rear \n 2. insertfront \n 3.
delete-rear \n 4. delete_front \n. 5.display
6. exit \n");
 printf(" enter choice \n");
   scanf("%d", &ch);

 switch (ch)
   {
    case1: printf(" enter the item \n");
           scanf("%d", & item);
            insert-rear();
             break;
```

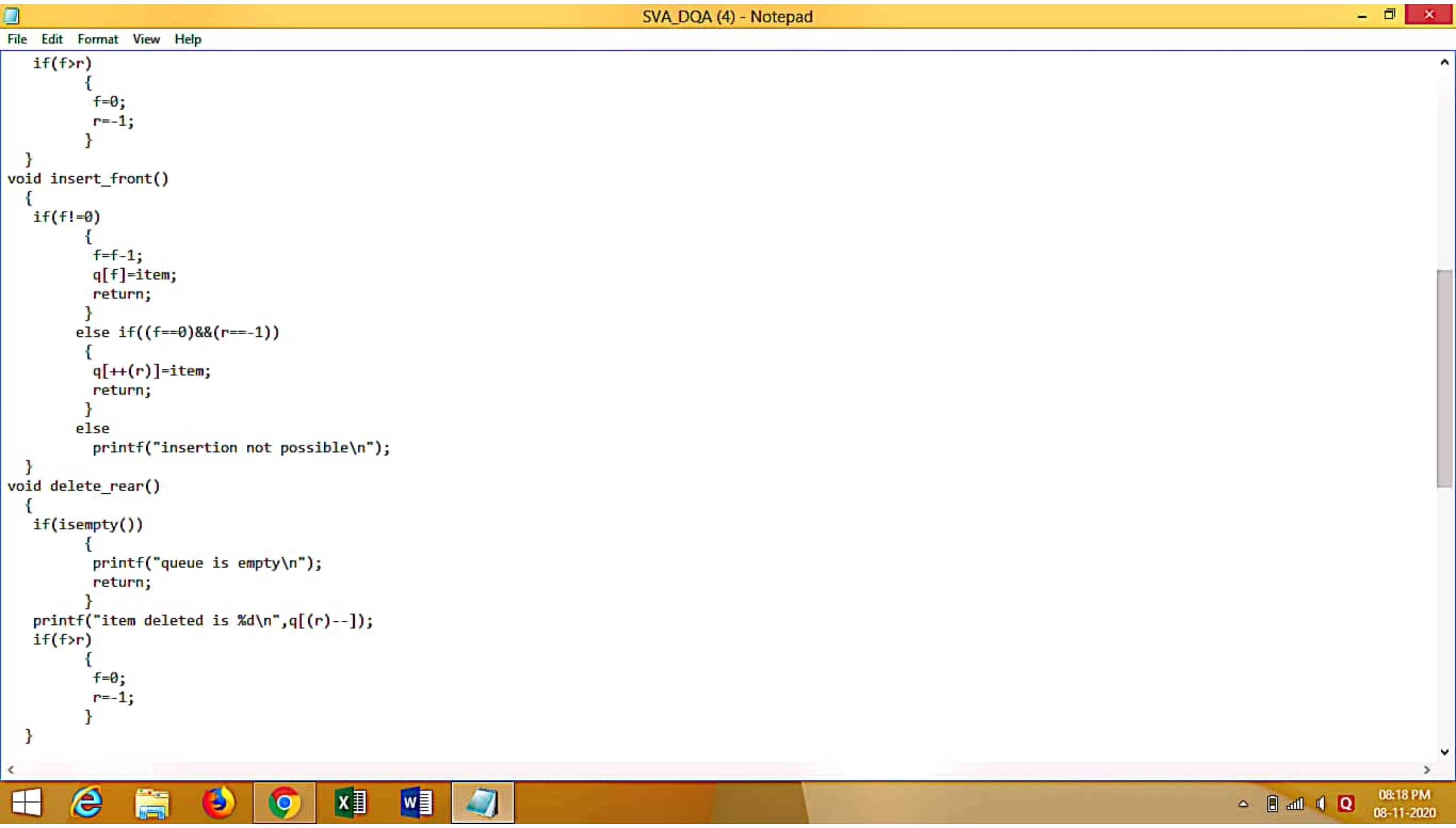
```
case 2 : printf ("enter the item\n"),
         scanf ("%d", & item),
         insert_front();
          break;
 case 3 : delete _ rear( )
          break;
 case 4 : delete_ front();
            break;
 case 5 :  display ();
            break;
   deafut: exit (0);
   }
  }
 }
```



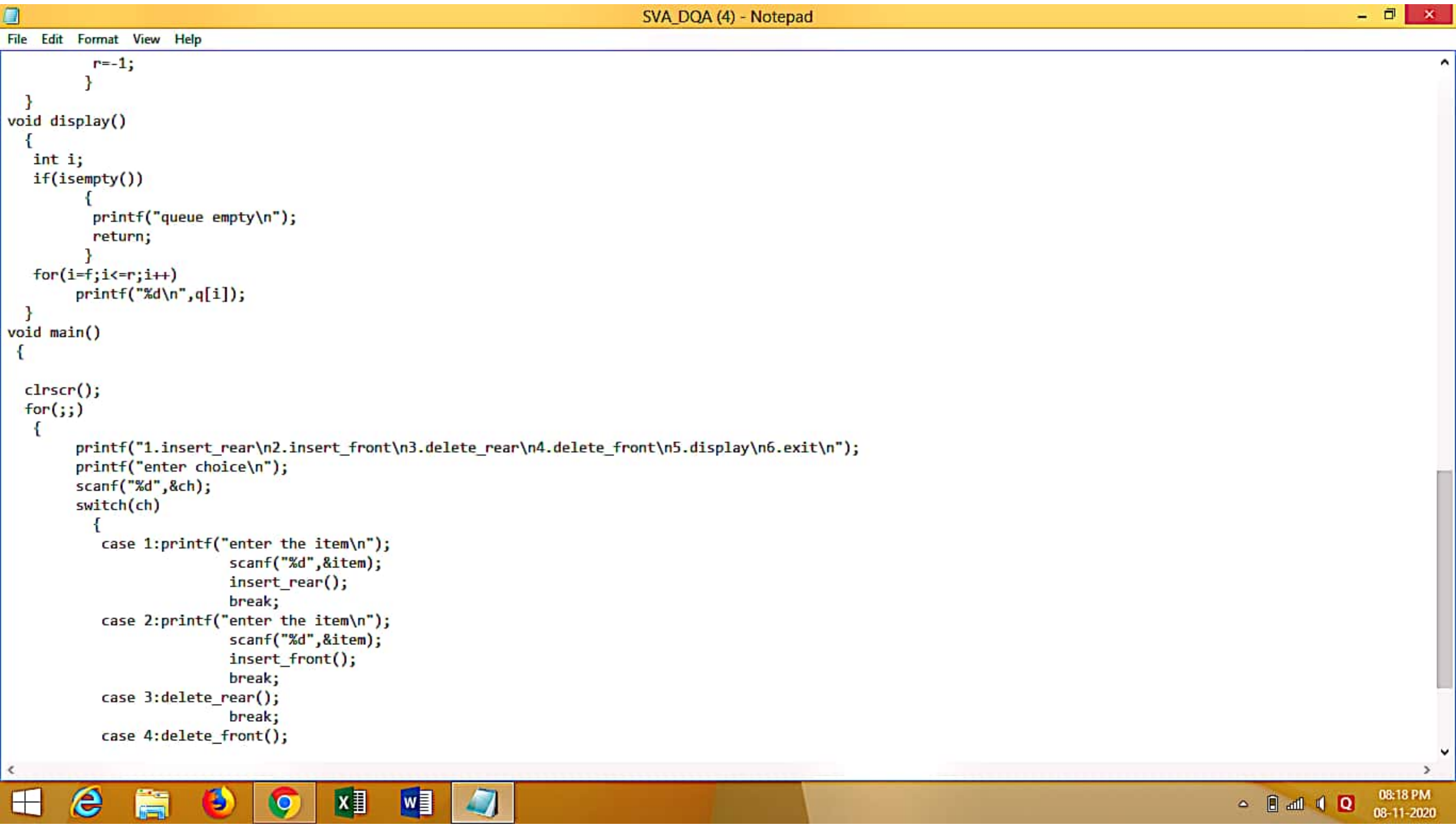
```
1
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
3
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
1
2
3
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
6

Process returned 0 (0x0)   execution time : 130.711 s
Press any key to continue.
```



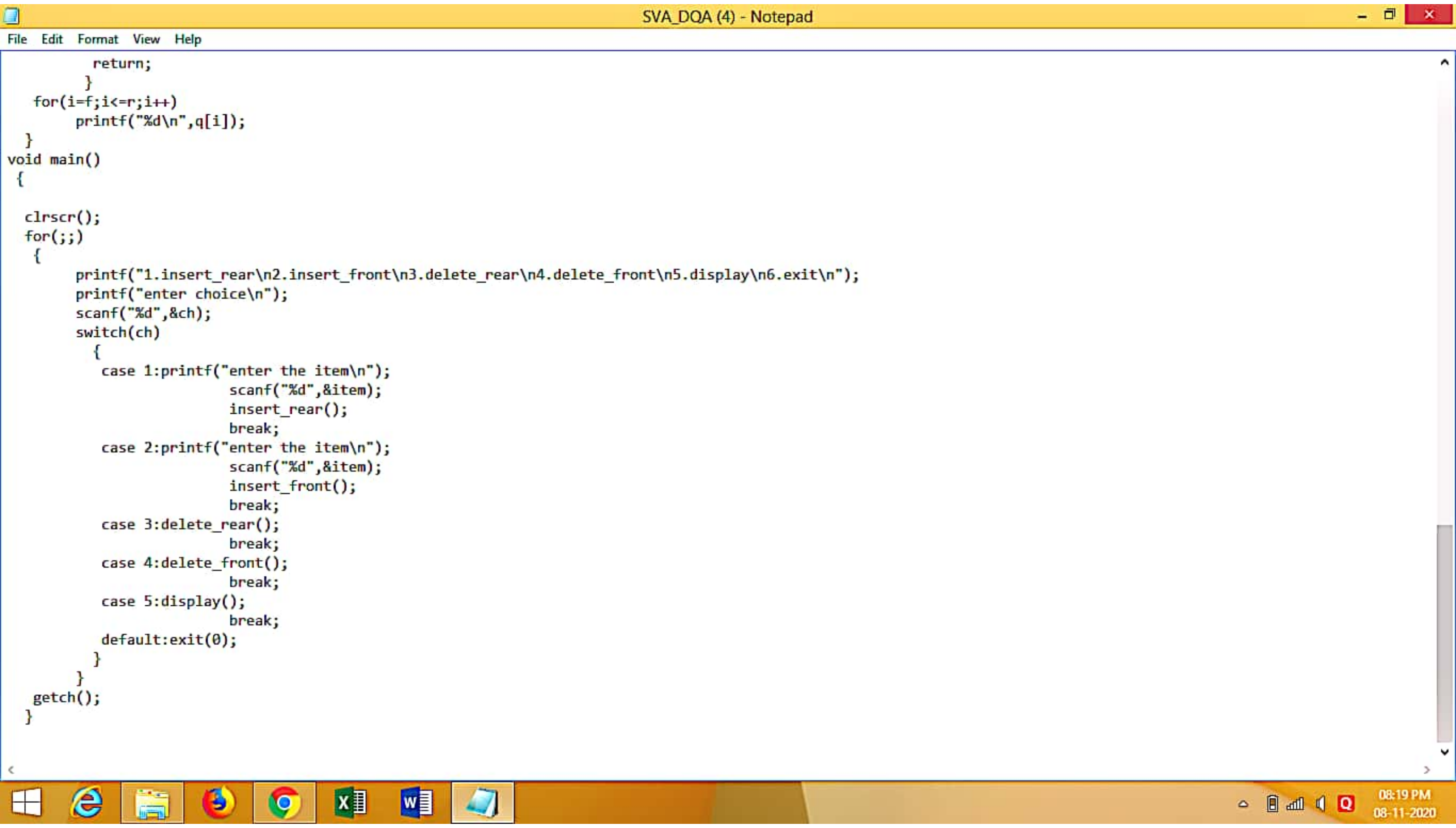
```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#define qsize 5
int f=0,r=-1,ch;
int item,q[10];

int isfull()
  {
   return(r==qsize-1)?1:0;
  }
int isempty()
  {
   return(f>r)?1:0;
  }
void insert_rear()
  {
   if(isfull())
         {
          printf("queue overflow\n");
          return;
         }
   r=r+1;
   q[r]=item;
  }
void delete_front()
  {
   if(isempty())
         {
          printf("queue empty\n");
          return;
         }
   printf("item deleted is %d\n",q[(f)++]);
   if(f>r)
         {
          f=0;
```



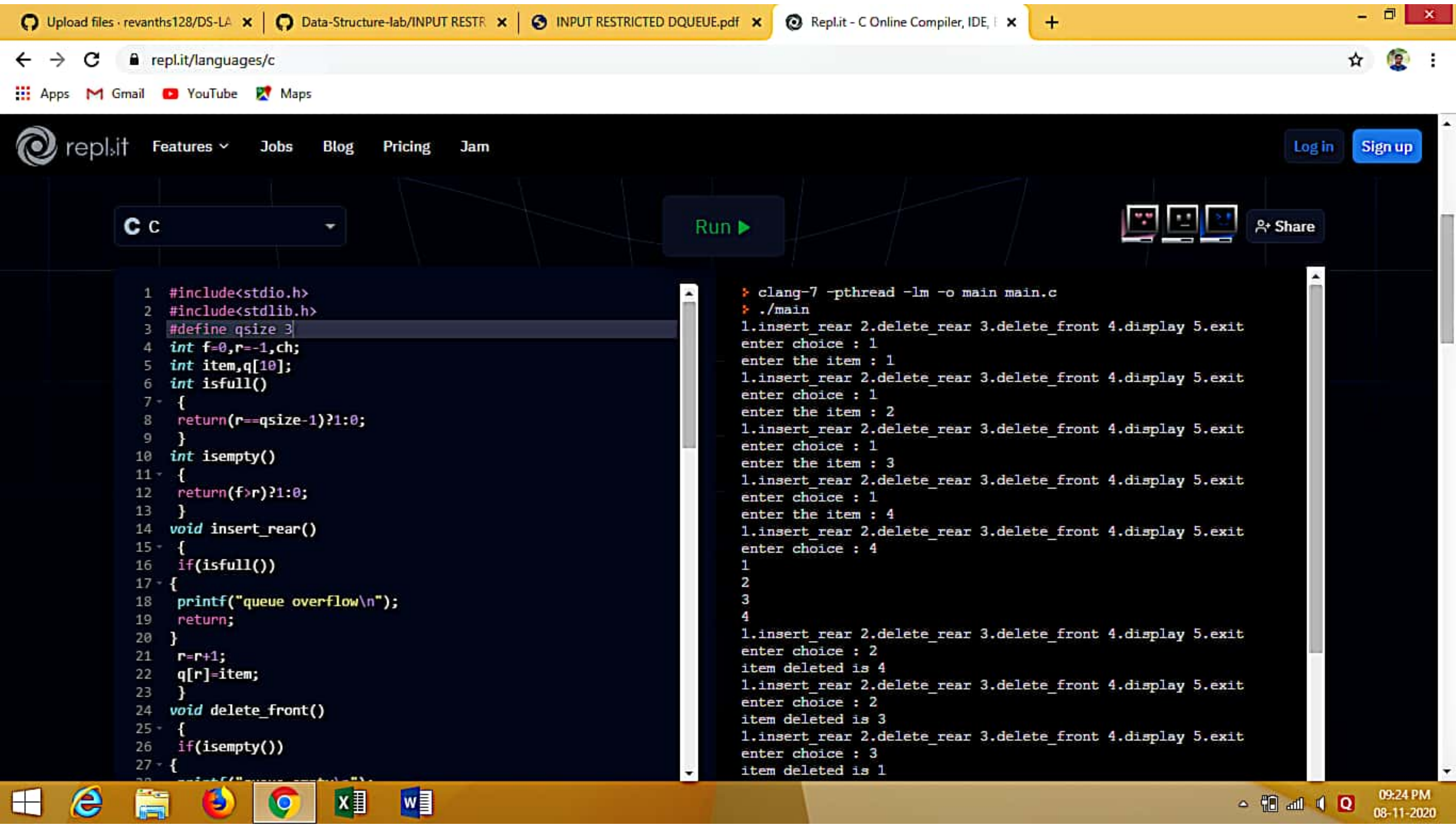
```
   if(f>r)
         {
          f=0;
          r=-1;
         }
  }
void insert_front()
  {
   if(f!=0)
         {
          f=f-1;
          q[f]=item;
          return;
         }
        else if((f==0)&&(r==-1))
         {
          q[++(r)]=item;
          return;
         }
        else
          printf("insertion not possible\n");
  }
void delete_rear()
  {
   if(isempty())
         {
          printf("queue is empty\n");
          return;
         }
   printf("item deleted is %d\n",q[(r)--]);
   if(f>r)
         {
          f=0;
          r=-1;
         }
  }
```



```
          r=-1;
         }
  }
void display()
  {
   int i;
   if(isempty())
         {
          printf("queue empty\n");
          return;
         }
   for(i=f;i<=r;i++)
        printf("%d\n",q[i]);
  }
void main()
 {

  clrscr();
  for(;;)
   {
        printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
        printf("enter choice\n");
        scanf("%d",&ch);
        switch(ch)
          {
           case 1:printf("enter the item\n");
                          scanf("%d",&item);
                          insert_rear();
                          break;
           case 2:printf("enter the item\n");
                          scanf("%d",&item);
                          insert_front();
                          break;
           case 3:delete_rear();
                          break;
           case 4:delete_front();
```



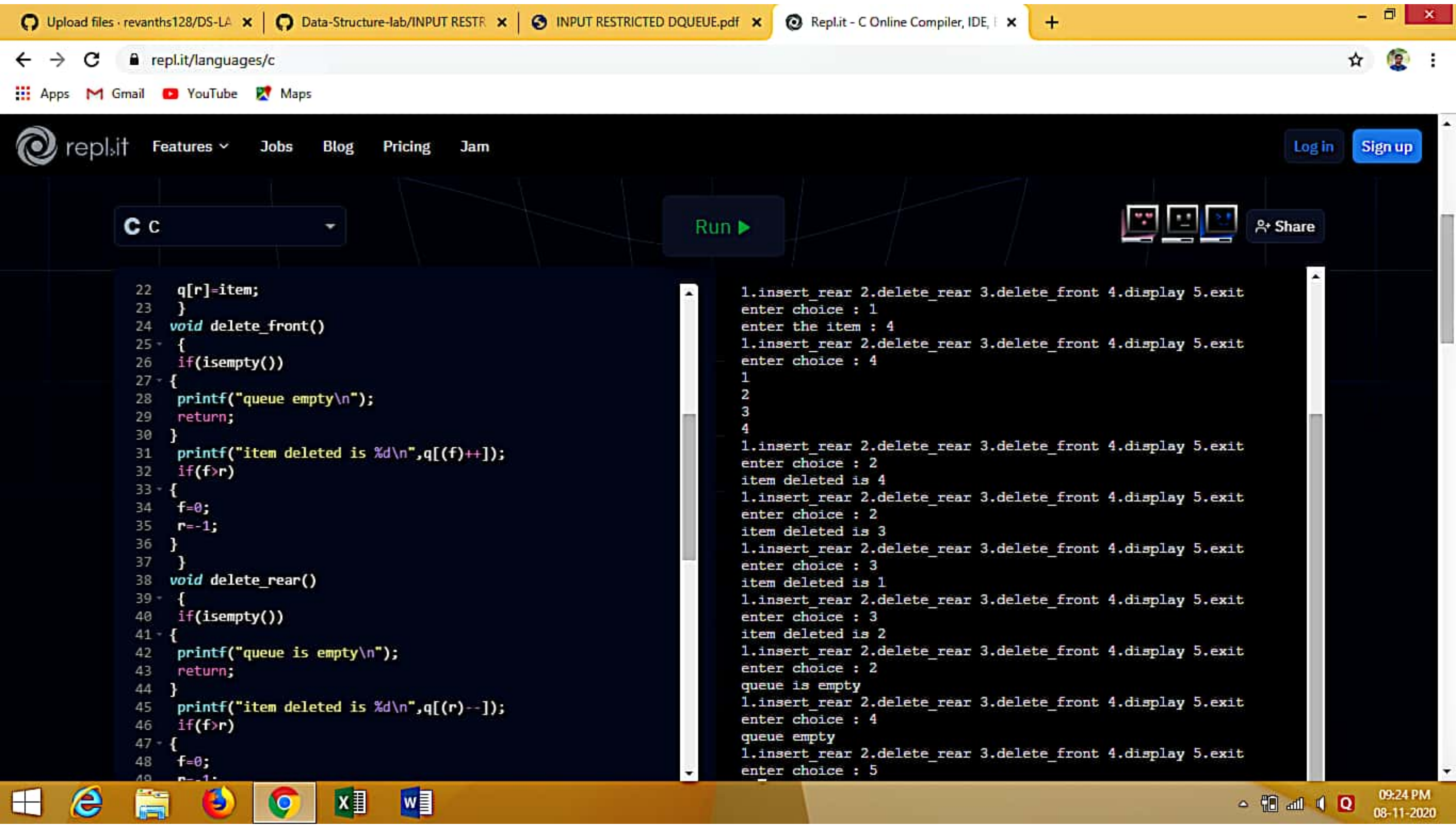
```
          return;
         }
   for(i=f;i<=r;i++)
        printf("%d\n",q[i]);
  }
void main()
 {

  clrscr();
  for(;;)
   {
        printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
        printf("enter choice\n");
        scanf("%d",&ch);
        switch(ch)
          {
           case 1:printf("enter the item\n");
                          scanf("%d",&item);
                          insert_rear();
                          break;
           case 2:printf("enter the item\n");
                          scanf("%d",&item);
                          insert_front();
                          break;
           case 3:delete_rear();
                          break;
           case 4:delete_front();
                          break;
           case 5:display();
                          break;
           default:exit(0);
          }
        }
   getch();
  }
```



Upload files · revanths128/DS-LA
Data-Structure-lab/INPUT RESTR
INPUT RESTRICTED DQUEUE.pdf
Repl.it - C Online Compiler, IDE, I
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:24 PM
08-11-2020

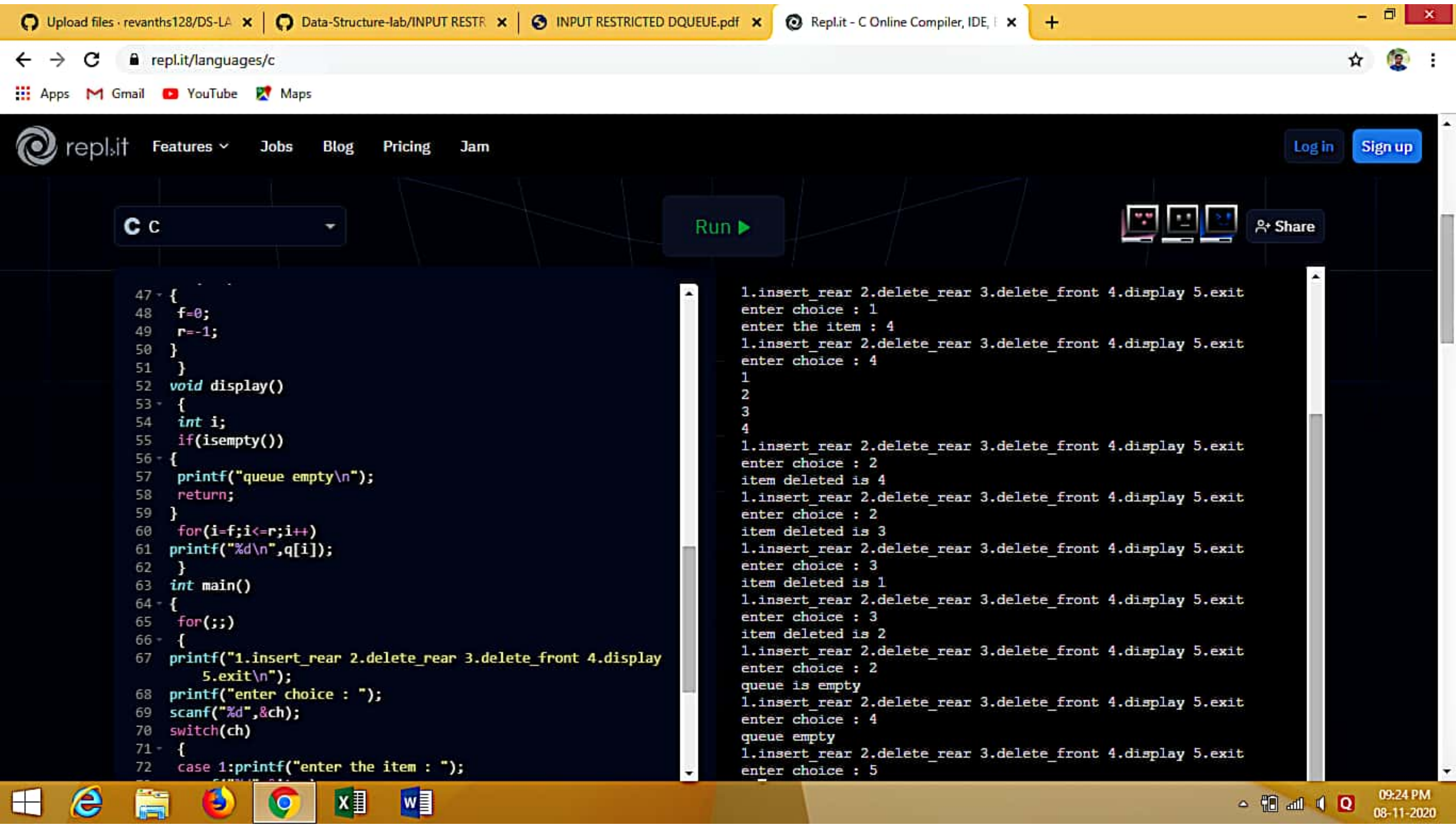
```c
#include<stdio.h>
#include<stdlib.h>
#define qsize 3
int f=0,r=-1,ch;
int item,q[10];
int isfull()
 {
 return(r==qsize-1)?1:0;
 }
int isempty()
 {
 return(f>r)?1:0;
 }
void insert_rear()
 {
 if(isfull())
{
 printf("queue overflow\n");
 return;
}
 r=r+1;
 q[r]=item;
 }
void delete_front()
 {
 if(isempty())
{
 printf("queue empty\n");
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
```


Upload files · revanths128/DS-LA
Data-Structure-lab/INPUT RESTR
INPUT RESTRICTED DQUEUE.pdf
Repl.it - C Online Compiler, IDE, I
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:24 PM
08-11-2020

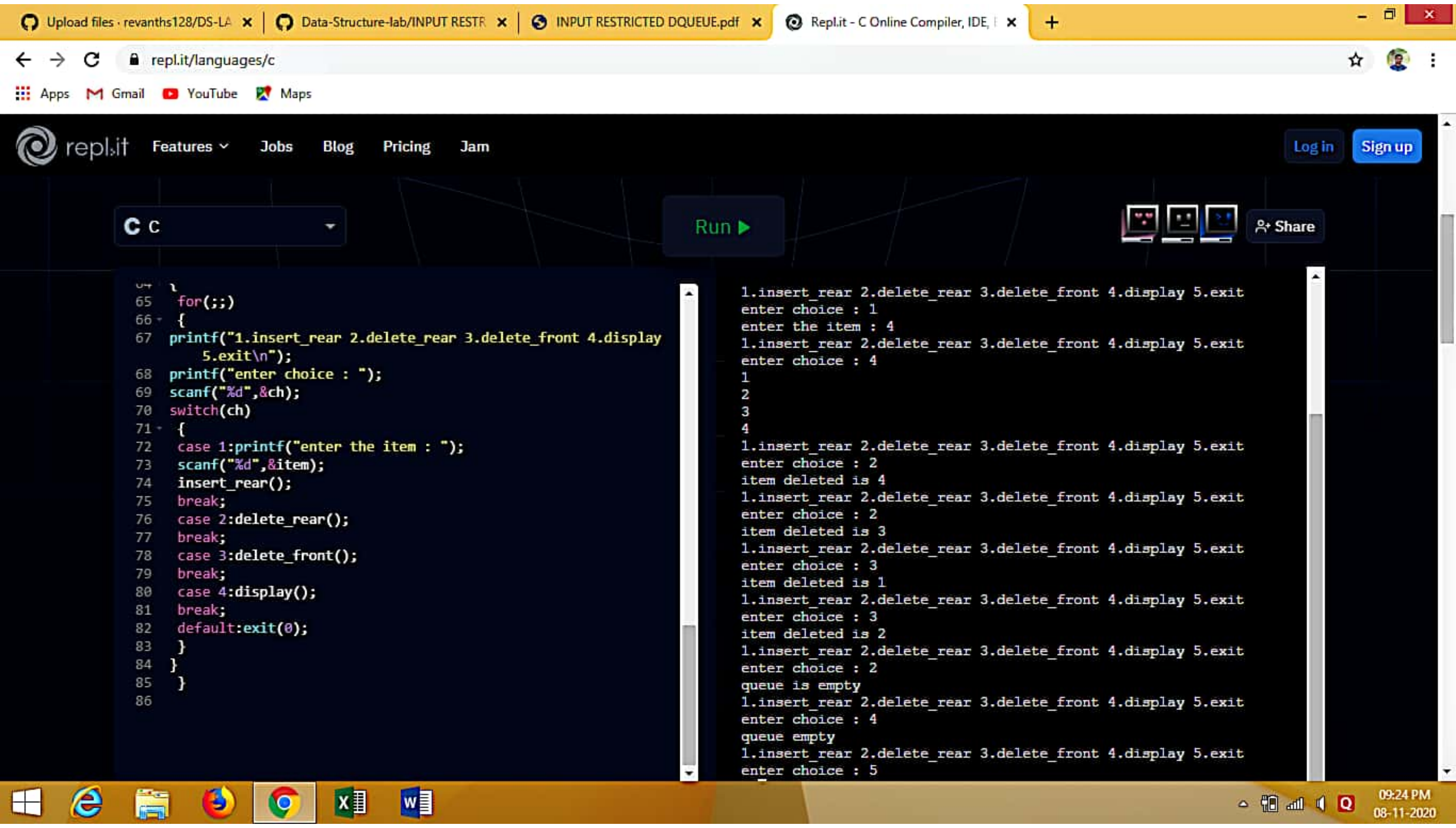
```c
 q[r]=item;
 }
void delete_front()
 {
 if(isempty())
{
 printf("queue empty\n");
 return;
}
 printf("item deleted is %d\n",q[(f)++]);
 if(f>r)
{
 f=0;
 r=-1;
}
 }
void delete_rear()
 {
 if(isempty())
{
 printf("queue is empty\n");
 return;
}
 printf("item deleted is %d\n",q[(r)--]);
 if(f>r)
{
 f=0;
```

```
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
queue is empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
queue empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 5
```


Upload files · revanths128/DS-LA
Data-Structure-lab/INPUT RESTR
INPUT RESTRICTED DQUEUE.pdf
Repl.it - C Online Compiler, IDE,
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:24 PM
08-11-2020

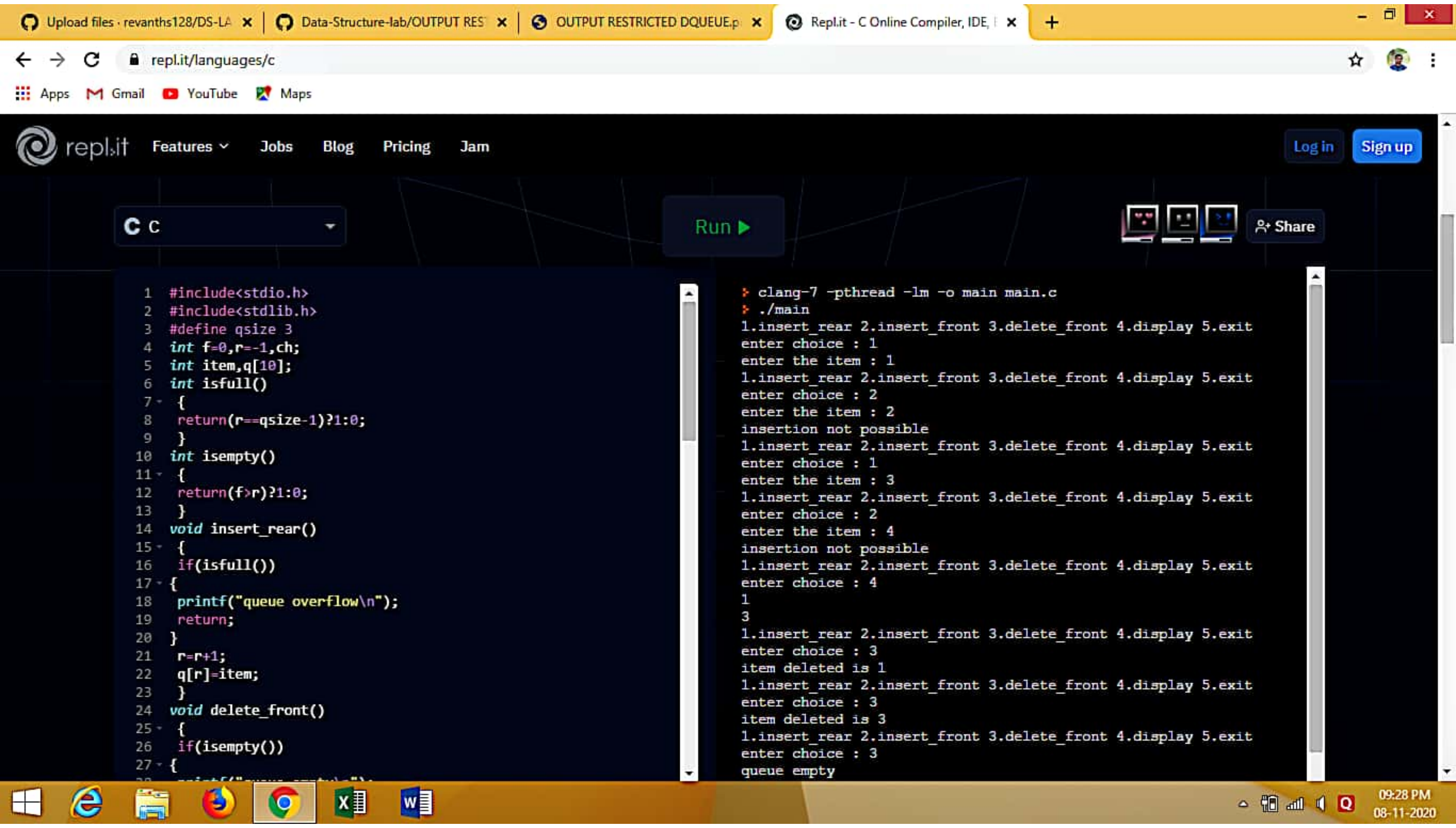
```
{
 f=0;
 r=-1;
}
 }
void display()
 {
 int i;
 if(isempty())
{
 printf("queue empty\n");
 return;
}
 for(i=f;i<=r;i++)
printf("%d\n",q[i]);
 }
int main()
{
 for(;;)
 {
printf("1.insert_rear 2.delete_rear 3.delete_front 4.display
    5.exit\n");
printf("enter choice : ");
scanf("%d",&ch);
switch(ch)
 {
 case 1:printf("enter the item : ");
```

```
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
queue is empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
queue empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 5
```


Upload files · revanths128/DS-LA
Data-Structure-lab/INPUT RESTR
INPUT RESTRICTED DQUEUE.pdf
Repl.it - C Online Compiler, IDE,
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:24 PM
08-11-2020

```c
{
 for(;;)
 {
printf("1.insert_rear 2.delete_rear 3.delete_front 4.display
    5.exit\n");
printf("enter choice : ");
scanf("%d",&ch);
switch(ch)
 {
 case 1:printf("enter the item : ");
 scanf("%d",&item);
 insert_rear();
 break;
 case 2:delete_rear();
 break;
 case 3:delete_front();
 break;
 case 4:display();
 break;
 default:exit(0);
 }
}
 }
```

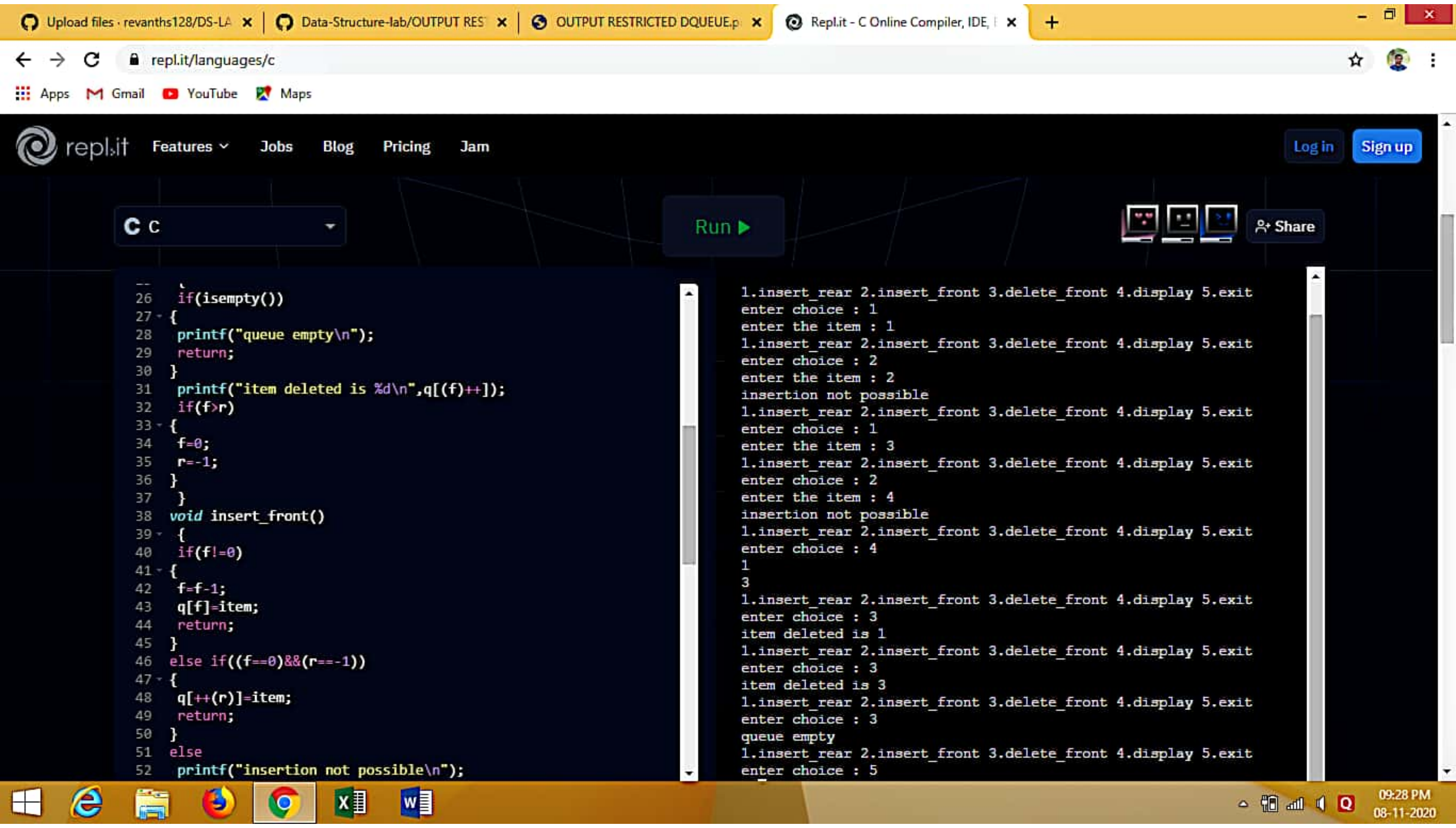
```
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
queue is empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
queue empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 5
```



Upload files · revanths128/DS-LA
Data-Structure-lab/OUTPUT RES
OUTPUT RESTRICTED DQUEUE.p
Repl.it - C Online Compiler, IDE, I
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:28 PM
08-11-2020

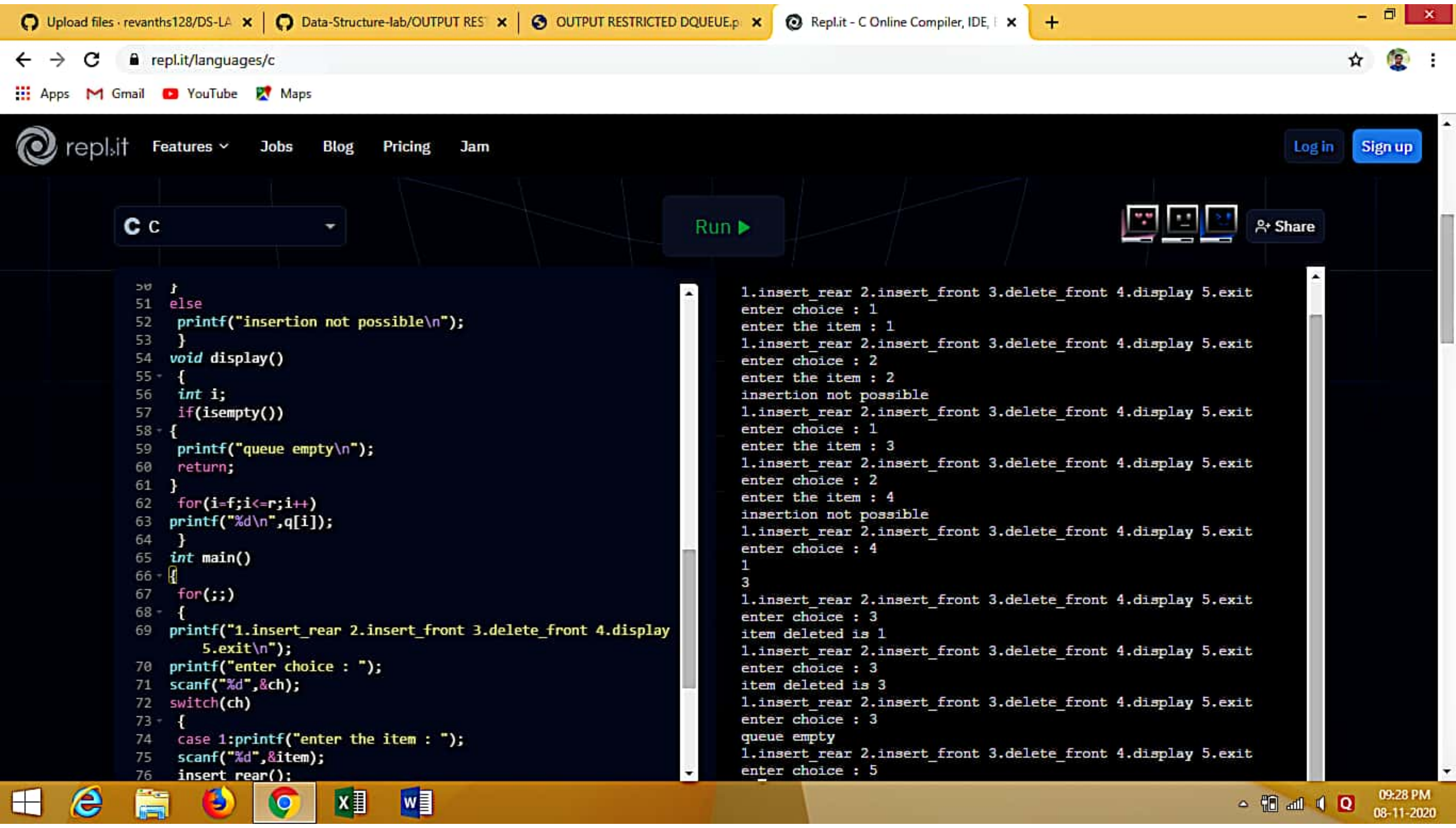
```c
#include<stdio.h>
#include<stdlib.h>
#define qsize 3
int f=0,r=-1,ch;
int item,q[10];
int isfull()
 {
 return(r==qsize-1)?1:0;
 }
int isempty()
 {
 return(f>r)?1:0;
 }
void insert_rear()
 {
 if(isfull())
{
 printf("queue overflow\n");
 return;
}
 r=r+1;
 q[r]=item;
 }
void delete_front()
 {
 if(isempty())
{
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
```


Upload files · revanths128/DS-LA
Data-Structure-lab/OUTPUT RES
OUTPUT RESTRICTED DQUEUE.p
Repl.it - C Online Compiler, IDE, I
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:28 PM
08-11-2020

```
 if(isempty())
{
 printf("queue empty\n");
 return;
}
 printf("item deleted is %d\n",q[(f)++]);
 if(f>r)
{
 f=0;
 r=-1;
}
 }
void insert_front()
 {
 if(f!=0)
{
 f=f-1;
 q[f]=item;
 return;
}
else if((f==0)&&(r==-1))
{
 q[++(r)]=item;
 return;
}
else
 printf("insertion not possible\n");
```

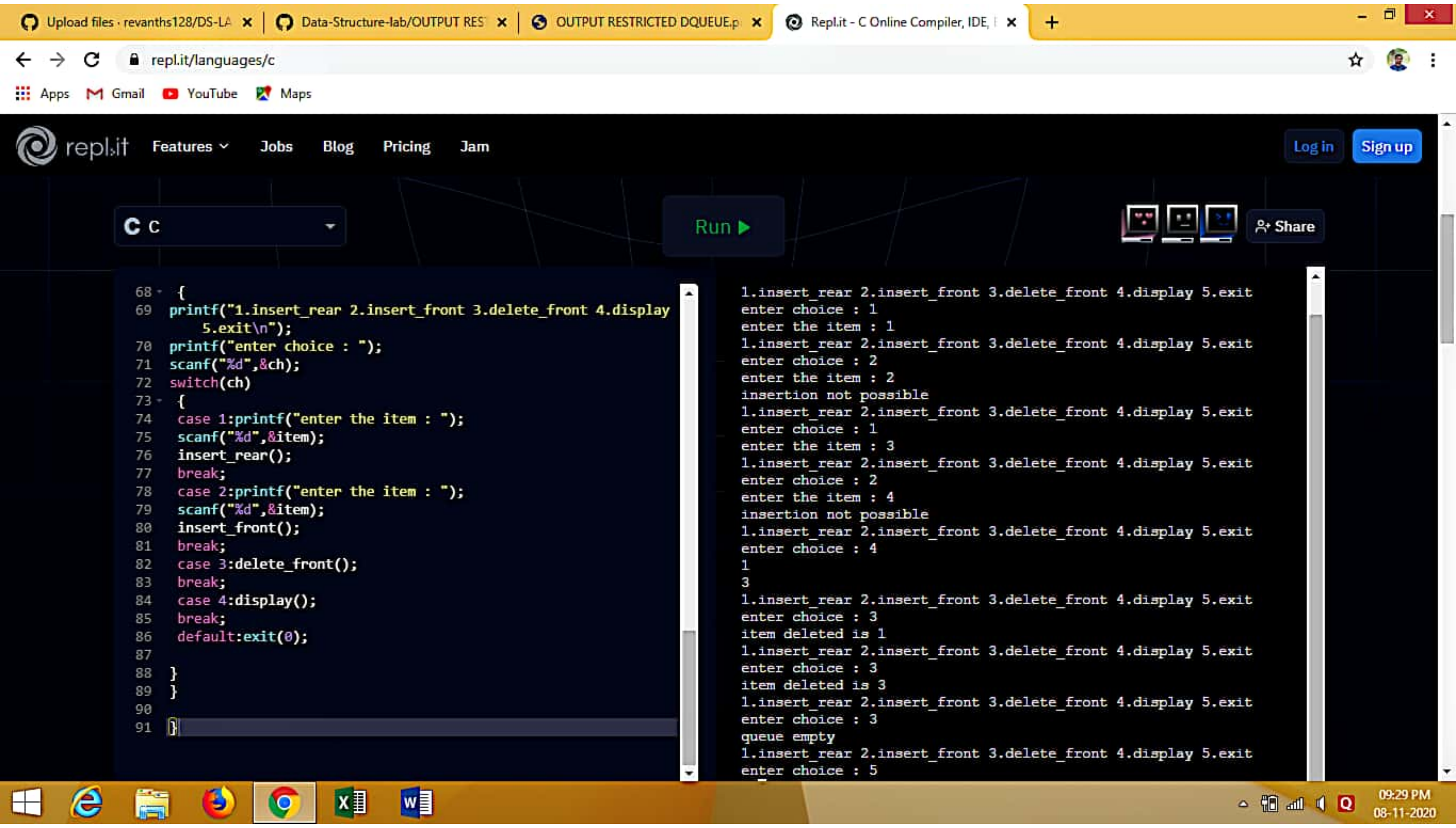
```
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 5
```



Upload files · revanths128/DS-LA
Data-Structure-lab/OUTPUT RES
OUTPUT RESTRICTED DQUEUE.p
Repl.it - C Online Compiler, IDE,
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:28 PM
08-11-2020

```
 }
 else
  printf("insertion not possible\n");
  }
 void display()
  {
  int i;
  if(isempty())
 {
  printf("queue empty\n");
  return;
 }
  for(i=f;i<=r;i++)
 printf("%d\n",q[i]);
  }
 int main()
 {
  for(;;)
  {
 printf("1.insert_rear 2.insert_front 3.delete_front 4.display
     5.exit\n");
 printf("enter choice : ");
 scanf("%d",&ch);
 switch(ch)
  {
  case 1:printf("enter the item : ");
  scanf("%d",&item);
  insert_rear();
```

```
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 5
```



Upload files · revanths128/DS-LA
Data-Structure-lab/OUTPUT REST
OUTPUT RESTRICTED DQUEUE.p
Repl.it - C Online Compiler, IDE,
repl.it/languages/c
Apps
Gmail
YouTube
Maps
repl.it
Features
Jobs
Blog
Pricing
Jam
Log in
Sign up
C
Run ▶
Share
09:29 PM
08-11-2020

```c
 {
printf("1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit\n");
printf("enter choice : ");
scanf("%d",&ch);
switch(ch)
 {
 case 1:printf("enter the item : ");
 scanf("%d",&item);
 insert_rear();
 break;
 case 2:printf("enter the item : ");
 scanf("%d",&item);
 insert_front();
 break;
 case 3:delete_front();
 break;
 case 4:display();
 break;
 default:exit(0);

}
}

}
```

```
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 5
```