DATE PAGE

POOJA.K

IBMI9CS111

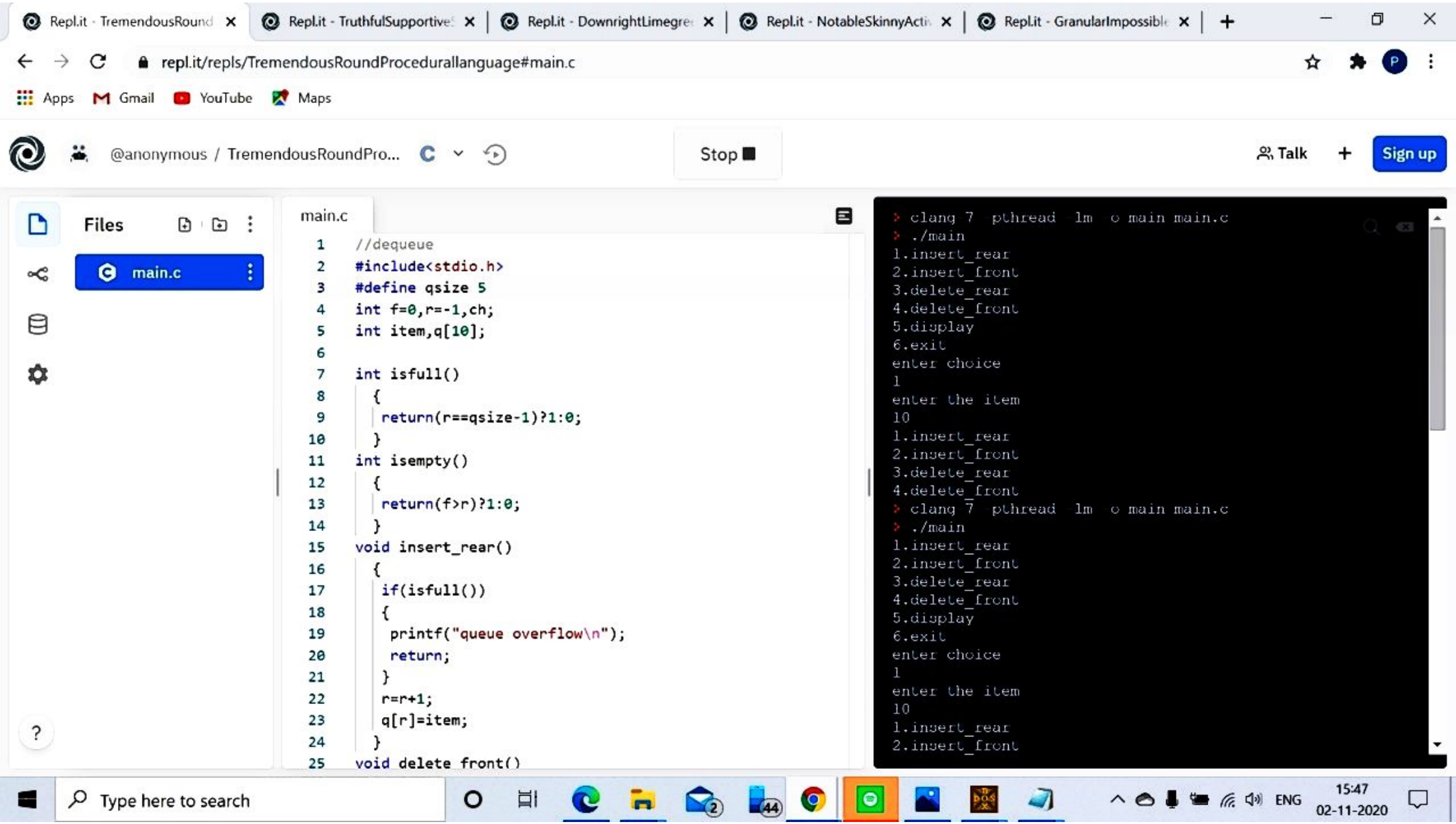
// Dequeue program

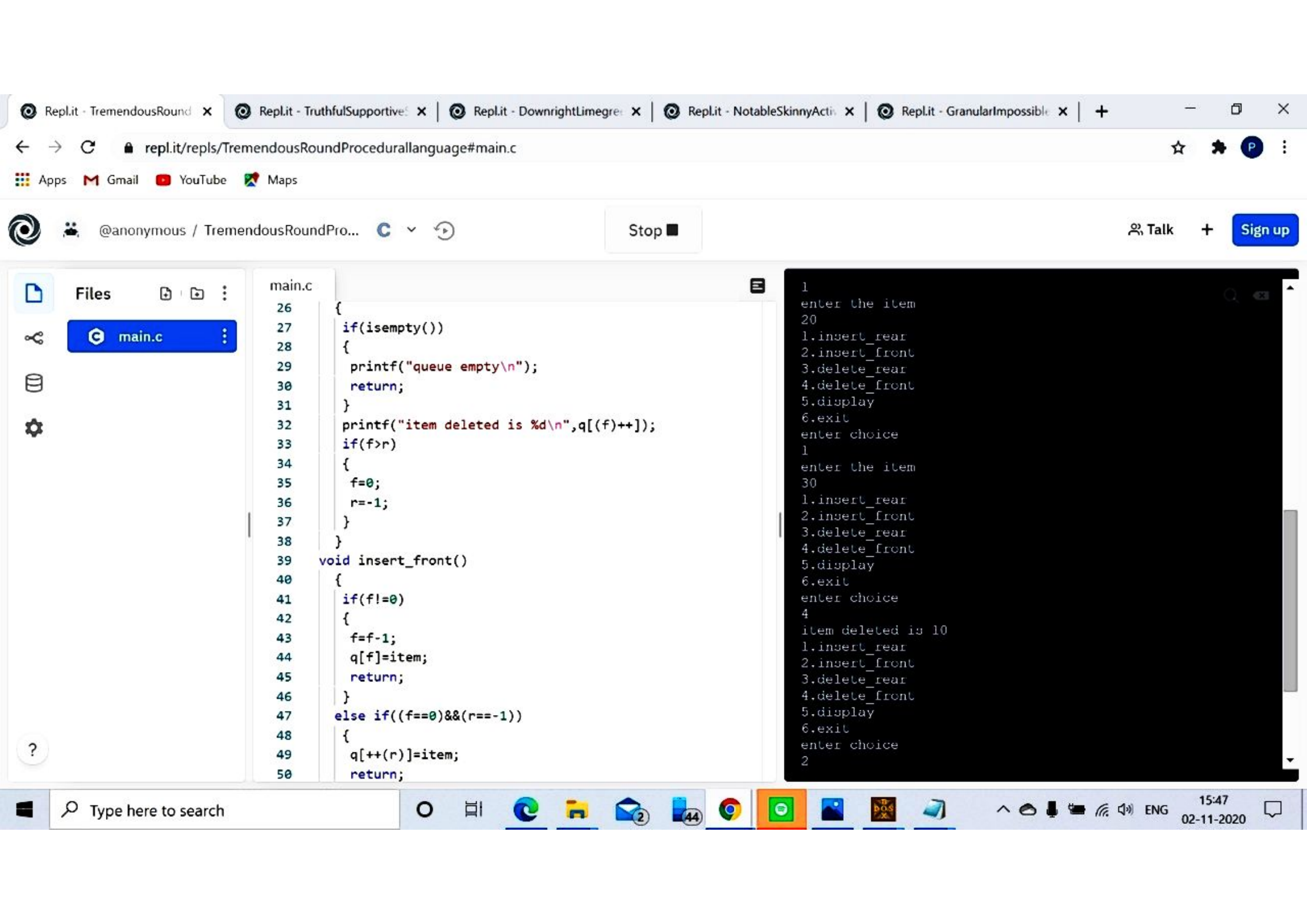
```
#include<stdio.h>
#include<stdlib.h>
#define qsize 5
int f=0, r=-1, ch;
int item, q[0];

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

int main()
 {
  for(;;)
```

```
{
printf("1.insert_rear\n 2.insert_front\n
3.delete_rear\n 4.delete_front\n 5.display
\n 6.exit\n");
printf("enter choice");
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);
    }
  }
  return 0;
}
```


Repl.it - TremendousRound
Repl.it - TruthfulSupportive
Repl.it - DownrightLimegre
Repl.it - NotableSkinnyActi
Repl.it - GranularImpossible
repl.it/repls/TremendousRoundProceduralLanguage#main.c
Apps
Gmail
YouTube
Maps
@anonymous / TremendousRoundPro...
Stop
Talk
Sign up
Files
main.c
//dequeue
#include<stdio.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()
> clang 7 pthread lm o main main.c
> ./main
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
> clang 7 pthread lm o main main.c
> ./main
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
Type here to search
ENG
15:47
02-11-2020


Repl.it - TremendousRound
Repl.it - TruthfulSupportive
Repl.it - DownrightLimegre
Repl.it - NotableSkinnyActi
Repl.it - GranularImpossible
repl.it/repls/TremendousRoundProceduralLanguage#main.c
Apps
Gmail
YouTube
Maps
@anonymous / TremendousRoundPro...
Stop
Talk
Sign up
Files
main.c
main.c
26 {
27 if(isempty())
28 {
29 printf("queue empty\n");
30 return;
31 }
32 printf("item deleted is %d\n",q[(f)++]);
33 if(f>r)
34 {
35 f=0;
36 r=-1;
37 }
38 }
39 void insert_front()
40 {
41 if(f!=0)
42 {
43 f=f-1;
44 q[f]=item;
45 return;
46 }
47 else if((f==0)&&(r==-1))
48 {
49 q[++(r)]=item;
50 return;
1
enter the item
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
Type here to search
ENG
15:47
02-11-2020

Repl.it - TremendousRound | Repl.it - TruthfulSupportive | Repl.it - DownrightLimegre | Repl.it - NotableSkinnyActi | Repl.it - GranularImpossible

repl.it/repls/TremendousRoundProceduralLanguage#main.c

Apps Gmail YouTube Maps

@anonymous / TremendousRoundPro... Stop Talk Sign up

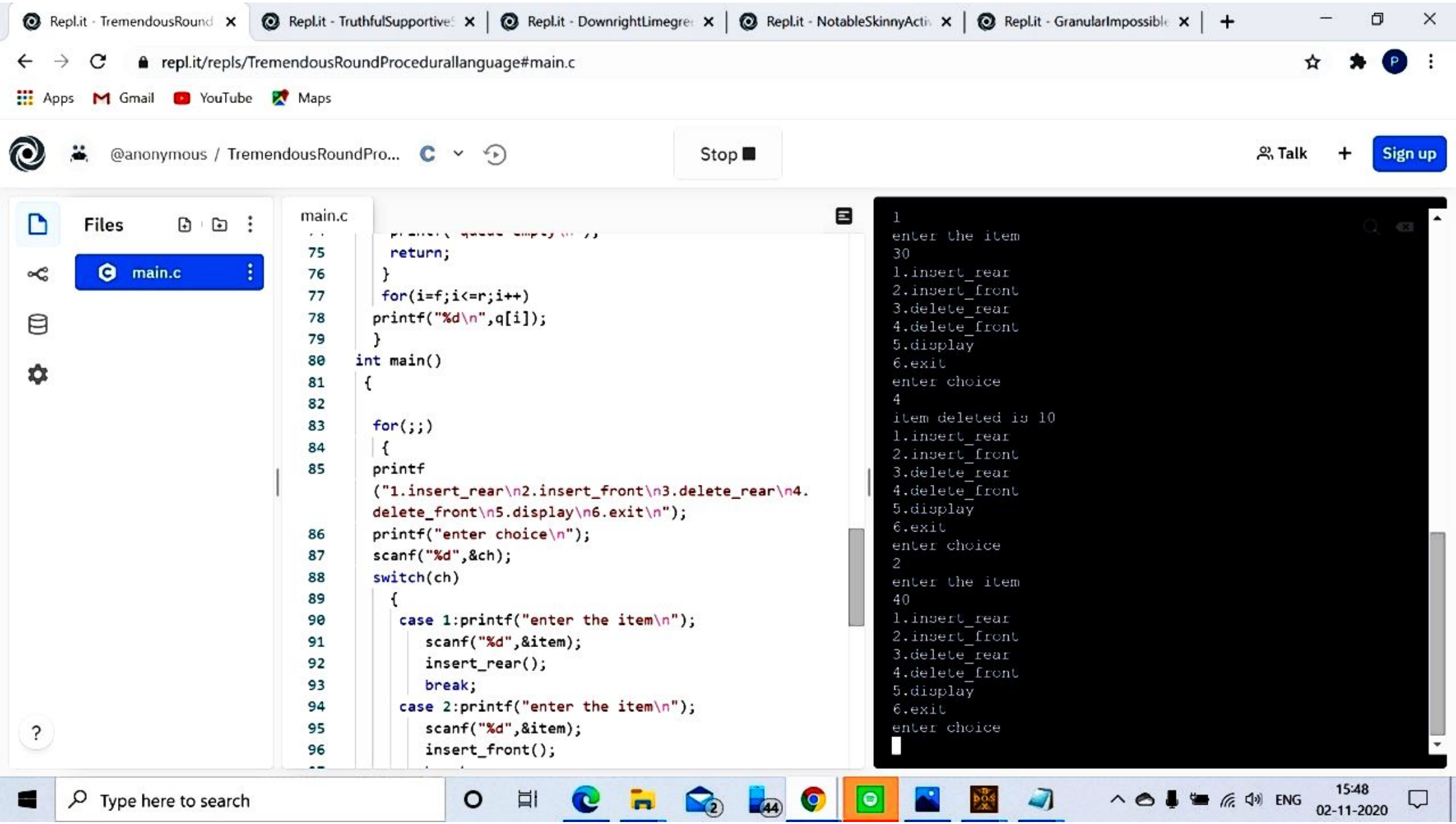
Files

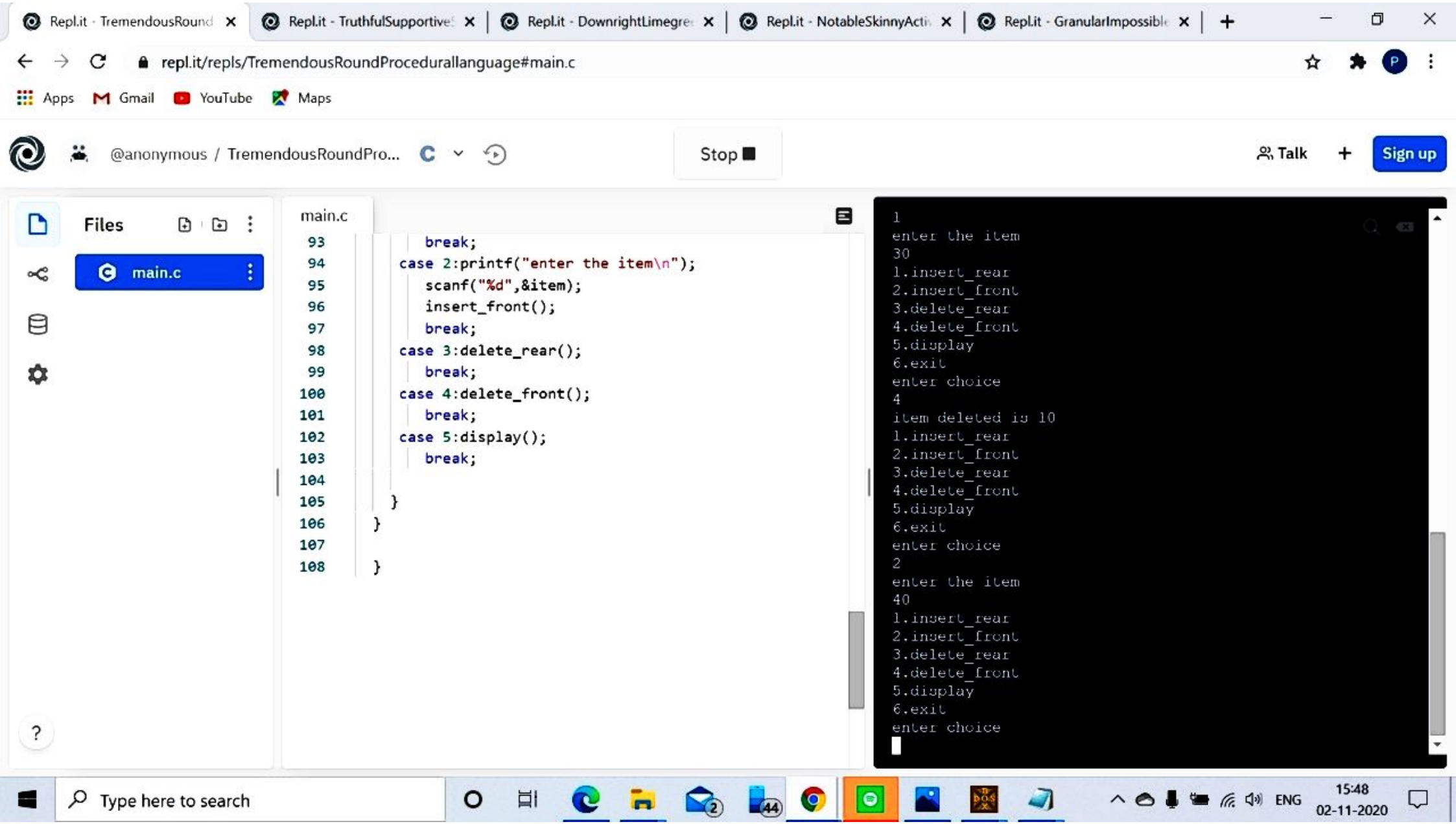
main.c

```c
  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;
   }
  }
void display()
  {
   int i;
   if(isempty())
   {
    printf("queue empty\n");
    return;
```

```
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

Type here to search ENG 15:47 02-11-2020


Repl.it - TremendousRound
Repl.it - TruthfulSupportive
Repl.it - DownrightLimegre
Repl.it - NotableSkinnyActi
Repl.it - GranularImpossible
repl.it/repls/TremendousRoundProceduralLanguage#main.c
Apps
Gmail
YouTube
Maps
@anonymous / TremendousRoundPro...
Stop
Talk
Sign up
Files
main.c
return;
}
for(i=f;i<=r;i++)
printf("%d\n",q[i]);
}
int main()
{
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();
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
Type here to search
ENG
15:48
02-11-2020


Repl.it - TremendousRound
Repl.it - TruthfulSupportive
Repl.it - DownrightLimegre
Repl.it - NotableSkinnyActi
Repl.it - GranularImpossible
repl.it/repls/TremendousRoundProceduralLanguage#main.c
Apps
Gmail
YouTube
Maps
@anonymous / TremendousRoundPro...
Stop
Talk
Sign up
Files
main.c
main.c
93 break;
94 case 2:printf("enter the item\n");
95 scanf("%d",&item);
96 insert_front();
97 break;
98 case 3:delete_rear();
99 break;
100 case 4:delete_front();
101 break;
102 case 5:display();
103 break;
104
105 }
106 }
107
108 }
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
Type here to search
ENG
15:48
02-11-2020