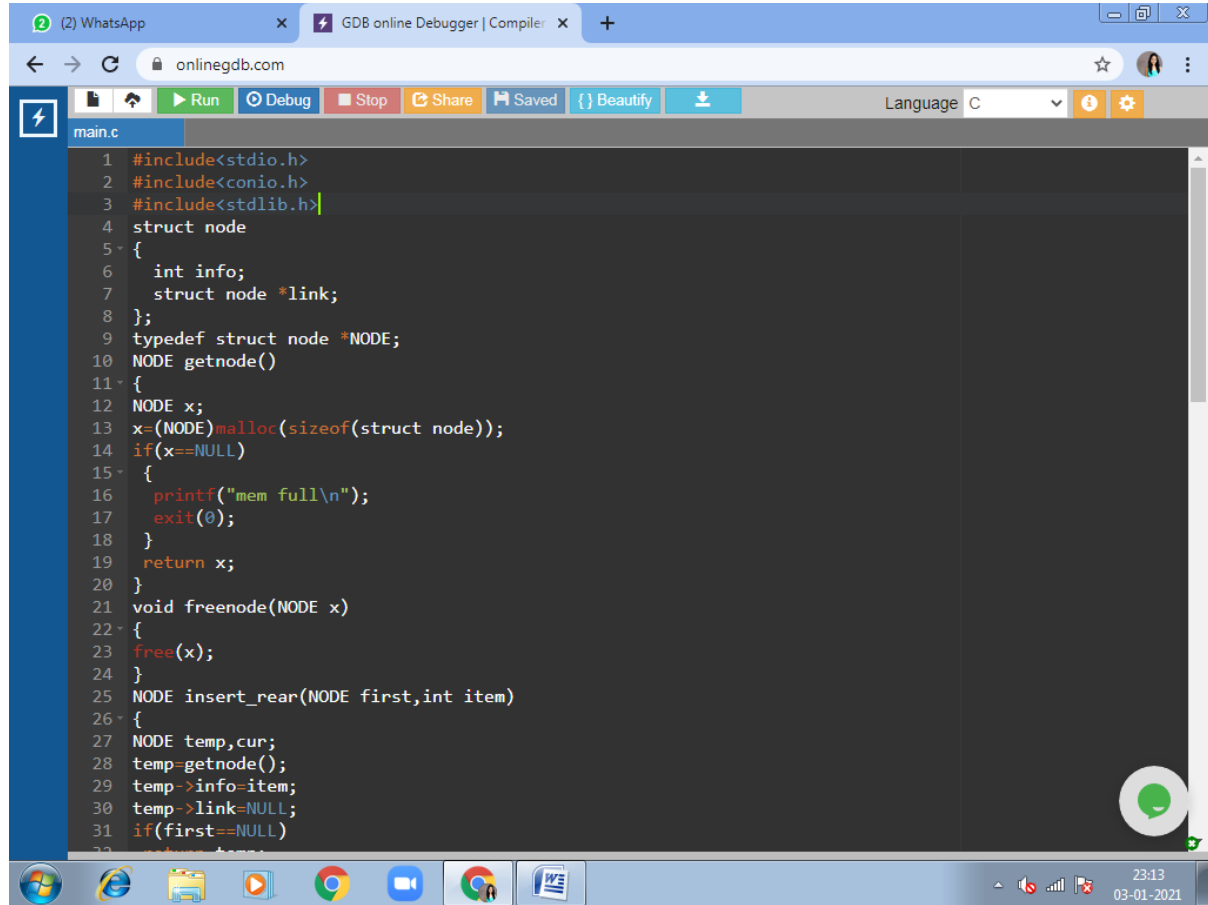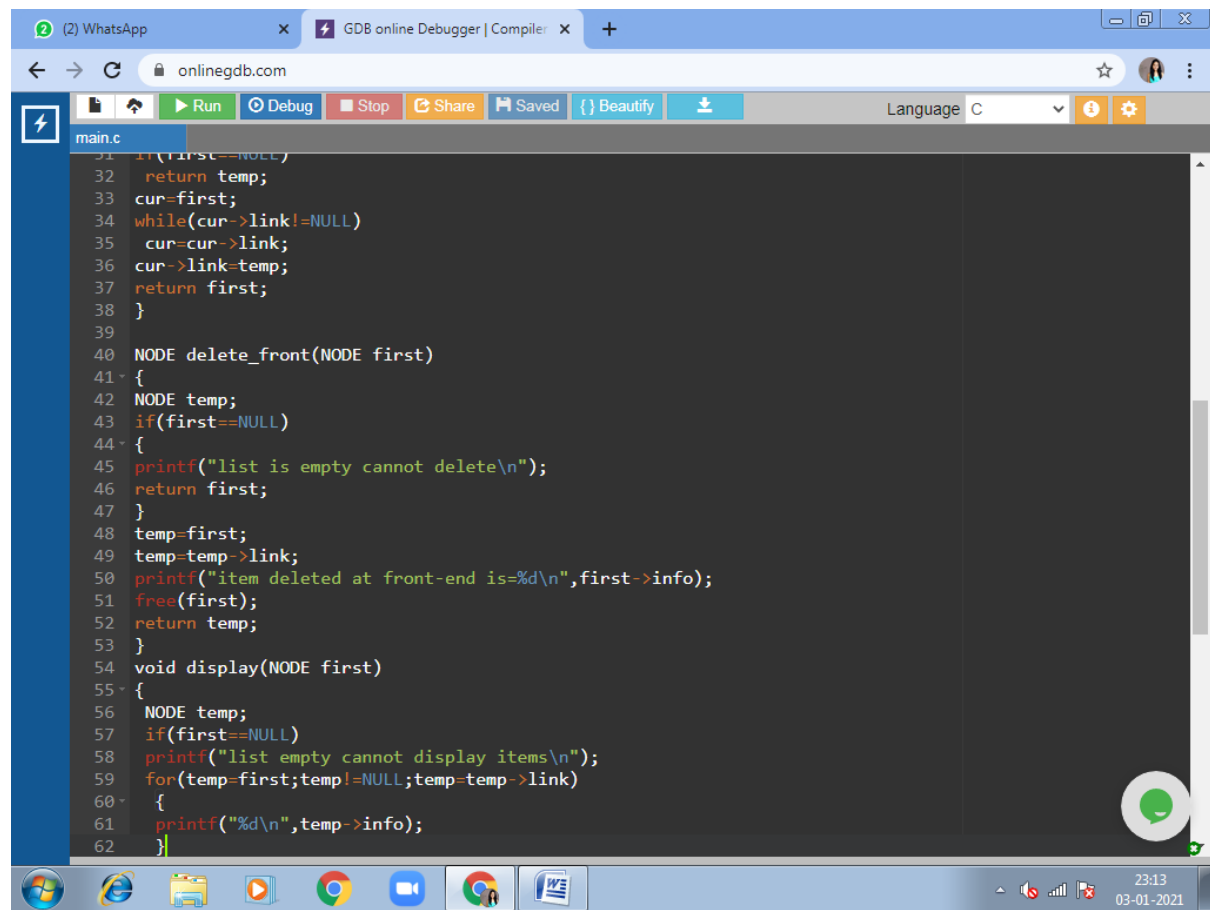# Q1)wap to implement queues using linked lists:

main.c
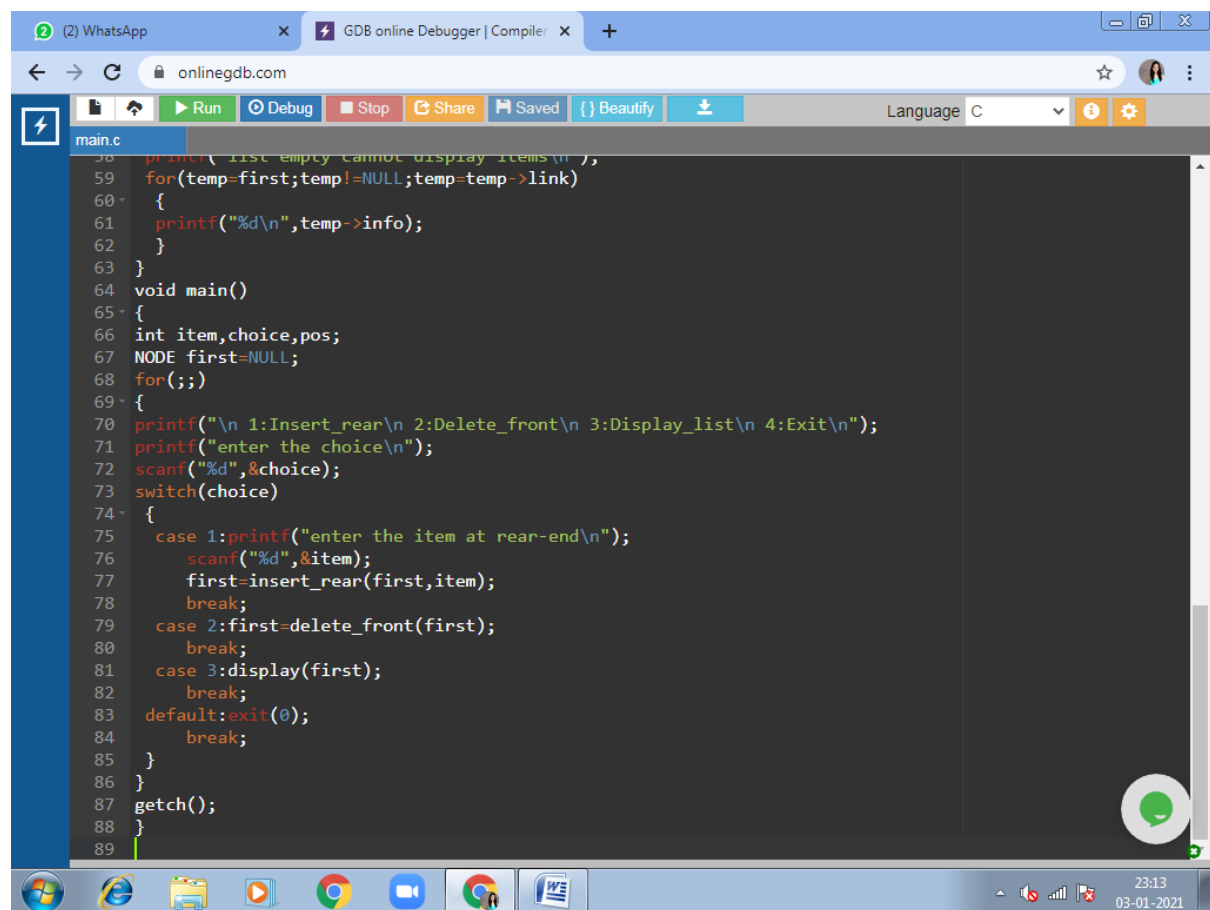
```c
1   #include<stdio.h>
2   #include<conio.h>
3   #include<stdlib.h>
4   struct node
5   {
6      int info;
7      struct node *link;
8   };
9   typedef struct node *NODE;
10  NODE getnode()
11  {
12  NODE x;
13  x=(NODE)malloc(sizeof(struct node));
14  if(x==NULL)
15   {
16    printf("mem full\n");
17    exit(0);
18   }
19   return x;
20  }
21  void freenode(NODE x)
22  {
23  free(x);
24  }
25  NODE insert_rear(NODE first,int item)
26  {
27  NODE temp,cur;
28  temp=getnode();
29  temp->info=item;
30  temp->link=NULL;
31  if(first==NULL)
```

```c
   if(first==NULL)
32   return temp;
33 cur=first;
34 while(cur->link!=NULL)
35   cur=cur->link;
36 cur->link=temp;
37 return first;
38 }
39
40 NODE delete_front(NODE first)
41 {
42 NODE temp;
43 if(first==NULL)
44 {
45 printf("list is empty cannot delete\n");
46 return first;
47 }
48 temp=first;
49 temp=temp->link;
50 printf("item deleted at front-end is=%d\n",first->info);
51 free(first);
52 return temp;
53 }
54 void display(NODE first)
55 {
56   NODE temp;
57   if(first==NULL)
58   printf("list empty cannot display items\n");
59   for(temp=first;temp!=NULL;temp=temp->link)
60   {
61   printf("%d\n",temp->info);
62   }
```

```c
     printf("list empty cannot display items\n");
59   for(temp=first;temp!=NULL;temp=temp->link)
60   {
61   printf("%d\n",temp->info);
62   }
63 }
64 void main()
65 {
66 int item,choice,pos;
67 NODE first=NULL;
68 for(;;)
69 {
70 printf("\n 1:Insert_rear\n 2:Delete_front\n 3:Display_list\n 4:Exit\n");
71 printf("enter the choice\n");
72 scanf("%d",&choice);
73 switch(choice)
74   {
75     case 1:printf("enter the item at rear-end\n");
76         scanf("%d",&item);
77         first=insert_rear(first,item);
78         break;
79     case 2:first=delete_front(first);
80         break;
81     case 3:display(first);
82         break;
83   default:exit(0);
84         break;
85   }
86 }
87 getch();
88 }
89
```

input

```
 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at rear-end
1

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at rear-end
2

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at rear-end
3

 1:Insert_rear
 2:Delete_front
 3:Display_list
```

input

```
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
3
1
2
3

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=1

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=2

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
3
```
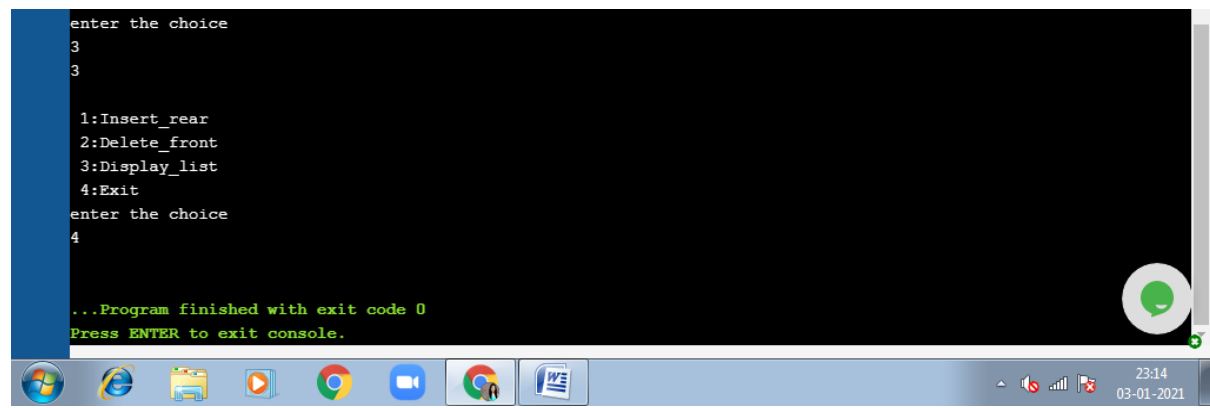
Q2)wap to implement stacks using linked lists:
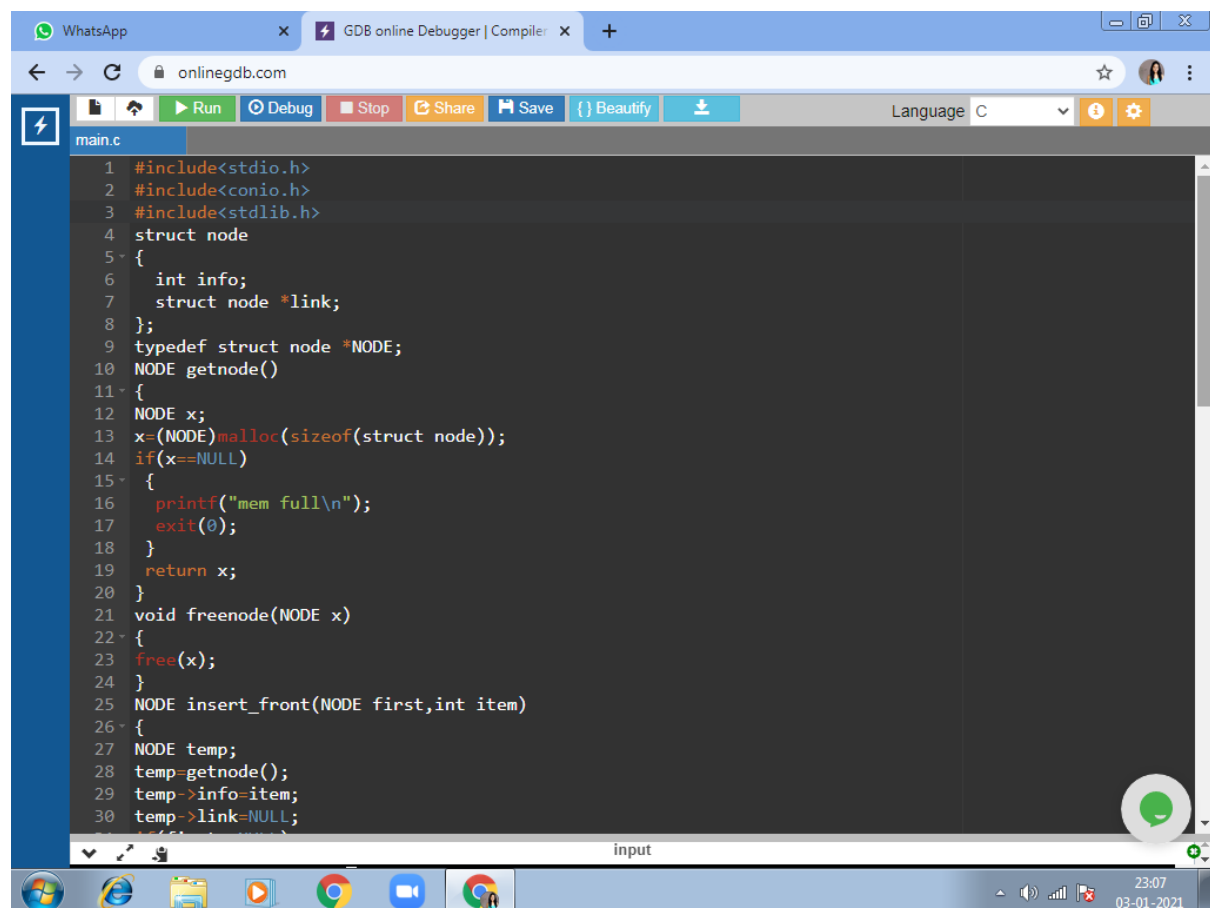
```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
   int info;
   struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
  {
   printf("mem full\n");
   exit(0);
  }
 return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_front(NODE first,int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
```

input

```c
30  temp->link=NULL;
31  if(first==NULL)
32  return temp;
33  temp->link=first;
34  first=temp;
35  return first;
36  }
37  NODE delete_front(NODE first)
38  {
39  NODE temp;
40  if(first==NULL)
41  {
42  printf("stack is empty cannot delete\n");
43  return first;
44  }
45  temp=first;
46  temp=temp->link;
47  printf("item deleted at front-end is=%d\n",first->info);
48  free(first);
49  return temp;
50  }
51  void display(NODE first)
52  {
53    NODE temp;
54    if(first==NULL)
55    printf("stack empty cannot display items\n");
56    for(temp=first;temp!=NULL;temp=temp->link)
57    {
58    printf("%d\n",temp->info);
59    }
```

```c
55    printf("stack empty cannot display items\n");
56    for(temp=first;temp!=NULL;temp=temp->link)
57    {
58    printf("%d\n",temp->info);
59    }
60  }
61  void main()
62  {
63  int item,choice,pos;
64  NODE first=NULL;
65  for(;;)
66  {
67  printf("\n 1:Insert_front[PUSH]\n 2:Delete_front[pop]\n 3:Display_list\n 4:Exit\n");
68  printf("enter the choice\n");
69  scanf("%d",&choice);
70  switch(choice)
71  {
72    case 1:printf("enter the item at front-end\n");
73        scanf("%d",&item);
74        first=insert_front(first,item);
75        break;
76    case 2:first=delete_front(first);
77        break;
78    case 3:display(first);
79        break;
80  default:exit(0);
81        break;
82  }
83  }
84  }
```

input

```
1:Insert_front[PUSH]
2:Delete_front[pop]
3:Display_list
4:Exit
enter the choice
1
enter the item at front-end
1

1:Insert_front[PUSH]
2:Delete_front[pop]
3:Display_list
4:Exit
enter the choice
1
enter the item at front-end
2

1:Insert_front[PUSH]
2:Delete_front[pop]
3:Display_list
4:Exit
enter the choice
1
enter the item at front-end
3

1:Insert_front[PUSH]
2:Delete_front[pop]
3:Display_list
```
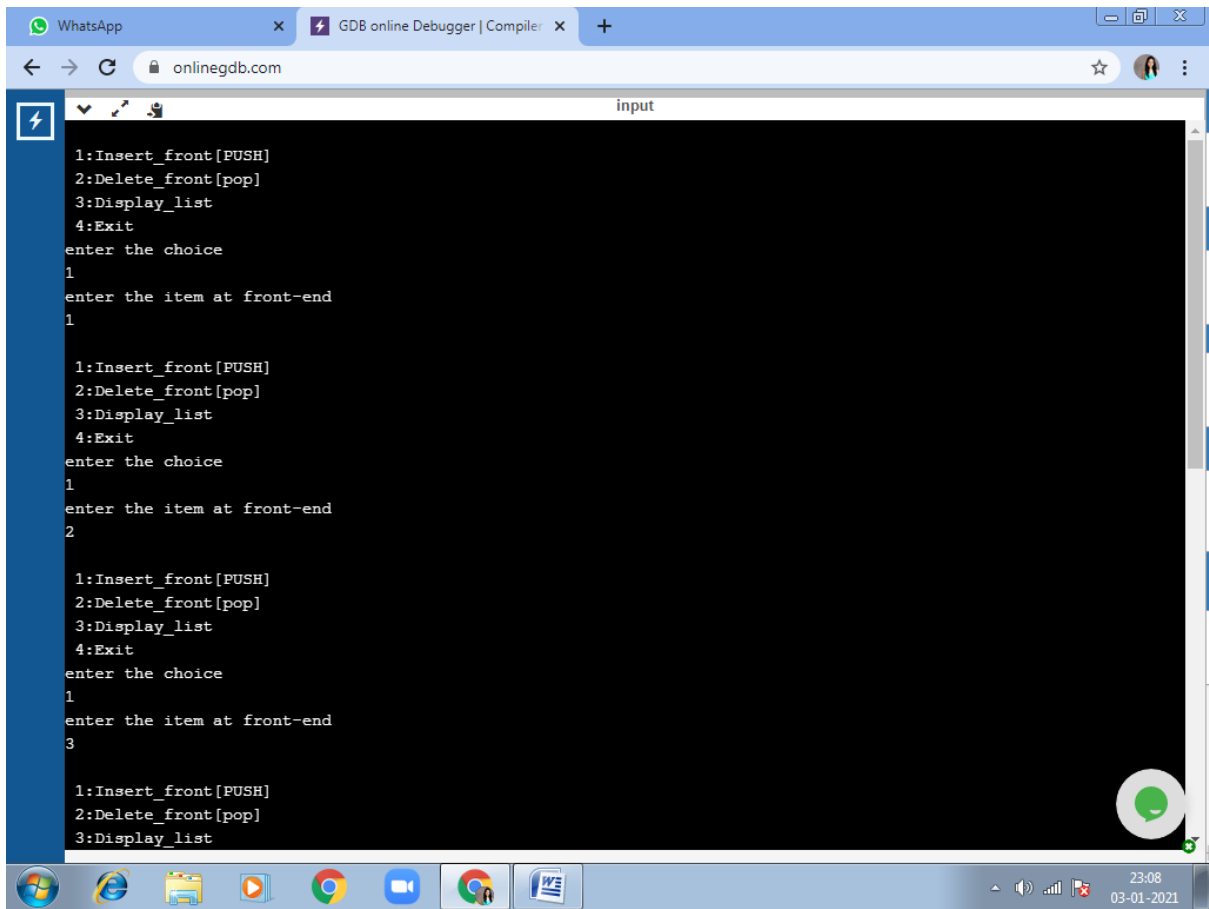
input

```
3
2
1

 1:Insert_front[PUSH]
 2:Delete_front[pop]
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=3

 1:Insert_front[PUSH]
 2:Delete_front[pop]
 3:Display_list
 4:Exit
enter the choice
3
2
1

 1:Insert_front[PUSH]
 2:Delete_front[pop]
 3:Display_list
 4:Exit
enter the choice
4


...Program finished with exit code 0
Press ENTER to exit console.
```