Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

Practical File

Data Structure and Algorithm (DSA)

(PCS 302)

2021-22

Submitted to:

Dr. Rakesh Patra

GEHU, D. Dun

Submitted by:

Vaibhav Kumar Kapriyal

University Roll. No.: 2018837
ClassRoll.No./Section: 60/A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

ACKNOWLEDGMENT

I would like to particularly thank my Data Science using OOPs with C++ Lab Faculty **Dr. Rakesh Patra** for his patience, support and encouragement throughout the completion of this Term work.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped me during this course.

Vaibhav Kumar Kapriyal
University. Roll No.- 2018837
B.Tech CSE-D-V Sem
Session: 2021-22
GEHU, Dehradun

B.Tech. CSE
STUDENT LAB REPORT SHEET

Photograph
Passport Size

Name of Student ……………………………………………Mob.No…………………………………………..

Address Permanent ………………………………………………………………………………………….

Father's Name ……………………….Occupation …………………………MoNo…….…………….…

Mother's Name ………………… …..Occupation…………………………MoNo……………………..

Section …………….Branch……………………Semester……………..Class Roll No……………… Grade A B C

Local Address…………………………………………….Email……………………………………. Marks 5 3 1

S.No.	Practical	D.O.P.	Date of Submission	Grade (Viva)	Grade (Report File)	Total Marks (out of 10)	Student's Signature	Teacher's Signature
1	Question-01							
2	Question-02							
3	Question-03							
4	Question-04							
5	Question-05							
6	Question-06							
7	Question-07							
8	Question-08							
9	Question-09							
10	Question-10							
11	Question-11							
12	Question-12							
13	Question-13							
14	Question-14							

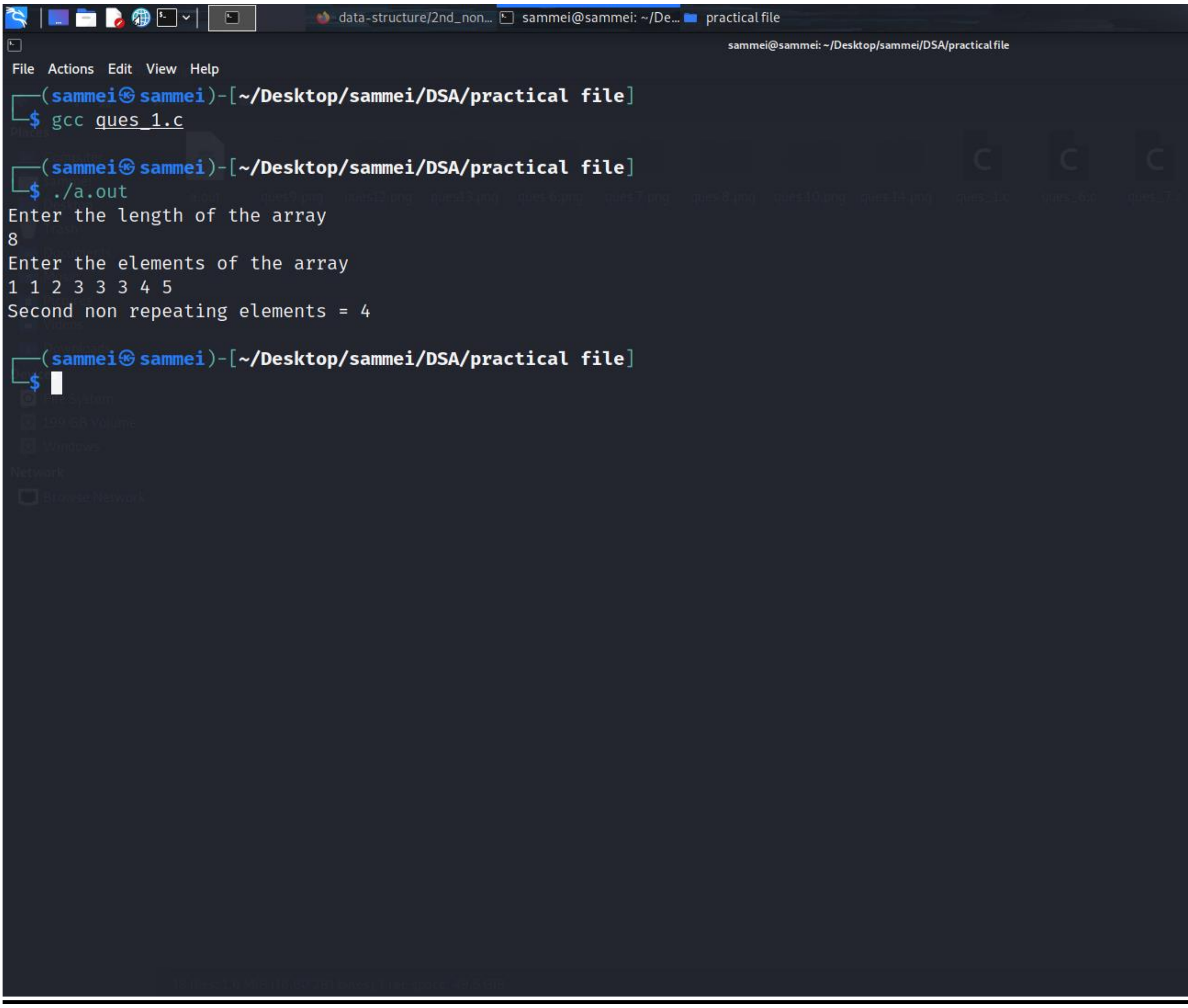
Q1. Write the C program to create an array by inserting N elements in it then find second non repeating element from the array

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>

            void check(int *a,int n)
            {
                   int i,t=0,c=0;
                   for(i=0;i<n;i++)
                   {
                          t=0;
                          for(int j=0;j<n;j++)
                          {
                                 if(a[i]==a[j])
                                 {
                                        t++;
                                 }
                          }
                          if(t==1)
                          {
                                 c++;
                          }
                          if(c==2)
                          {
                                 printf("Second non repeating elements = %d\n",a[i]);
                                 break;
                          }
                   }
            }

            int main()
            {
                   int n,i;
                   printf("Enter the length of the array\n");
                   scanf("%d",&n);
                   int a[n];
                   printf("Enter the elements of the array\n");
                   for(i=0;i<n;i++)
                   {
                          scanf("%d",&a[i]);
```

```
        }
        check(a,n);
        return 0;
}
```

Output:



data-structure/2nd_non...
sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File
Actions
Edit
View
Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ gcc ques_1.c

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
Enter the length of the array
8
Enter the elements of the array
1 1 2 3 3 3 4 5
Second non repeating elements = 4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

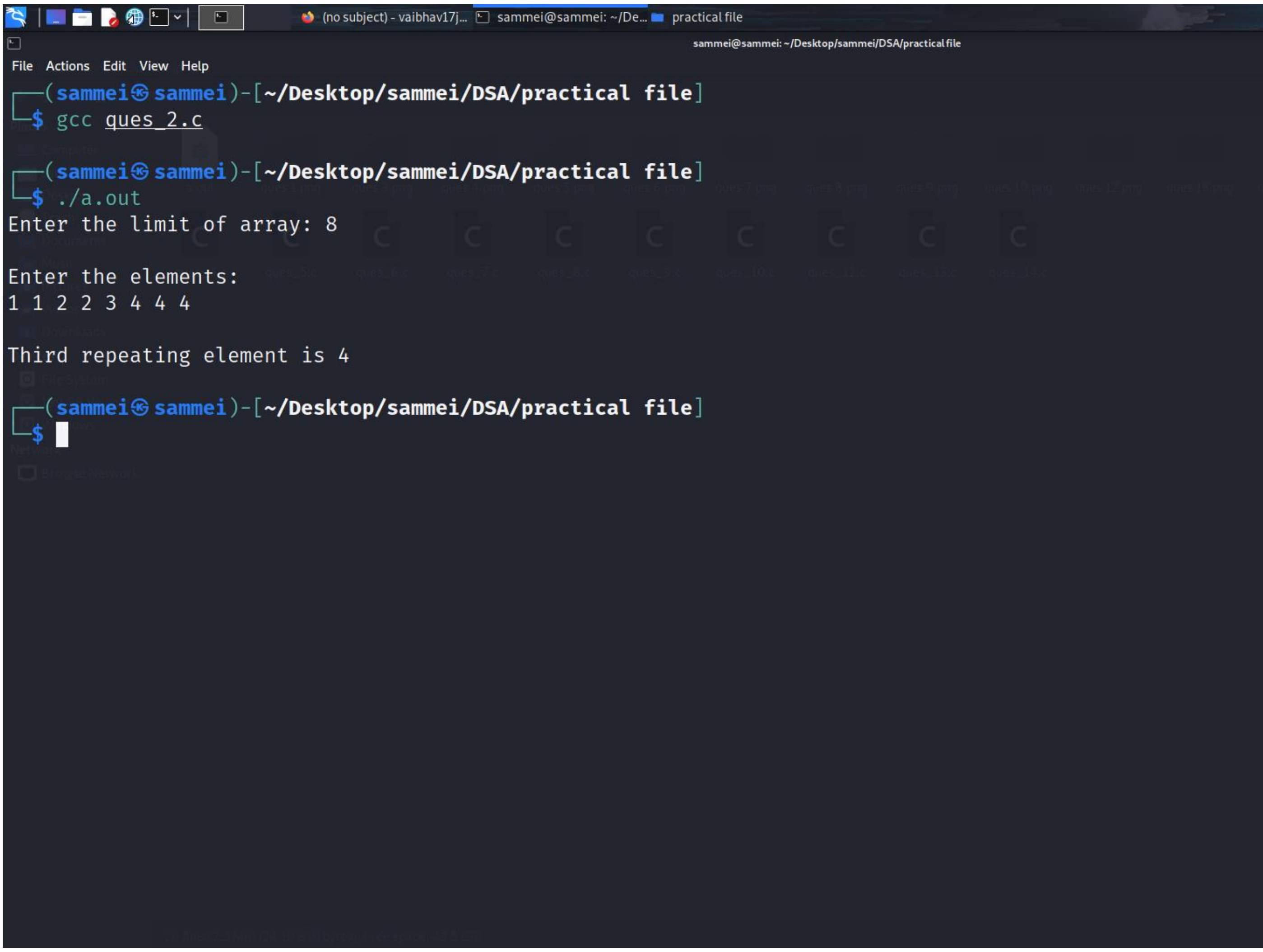
Q2. Write the C program to create an array by inserting N elements in it then find third repeating element from the array.

Solution:

```
#include<stdio.h>
                    int main()
                    {
                        int n,i,j,c,d=0;
                        printf("Enter the limit of array: ");
                        scanf("%d",&n);
                        int a[n];
                        printf("\nEnter the elements:\n");
                        for(i=0;i<n;i++)
                        {
                            scanf("%d",&a[i]);
                        }
                        for(i=0;i<n;i++)
                        {
                        c=0;
                        for(j=0;j<n;j++)
                        {
                        if(a[j]!=-1&&i!=j)
                        {
                        if(a[i]==a[j]&&i!=j)
                        {
                            c++;
                            a[j]=-1;
                        }
                        }
                        }
                            if(c>0)
                                d++;

                            if(d==3)
                            {
                                break;
                            }
                        }

                        if(d==3)
                            printf("\nThird repeating element is %d",a[i]);
                        else
                            printf("\nThird repeating element not found");
```

```
            return 0;

        }
```

Output:



(no subject) - vaibhav17j...
sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ gcc ques_2.c

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
Enter the limit of array: 8

Enter the elements:
1 1 2 2 3 4 4 4

Third repeating element is 4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

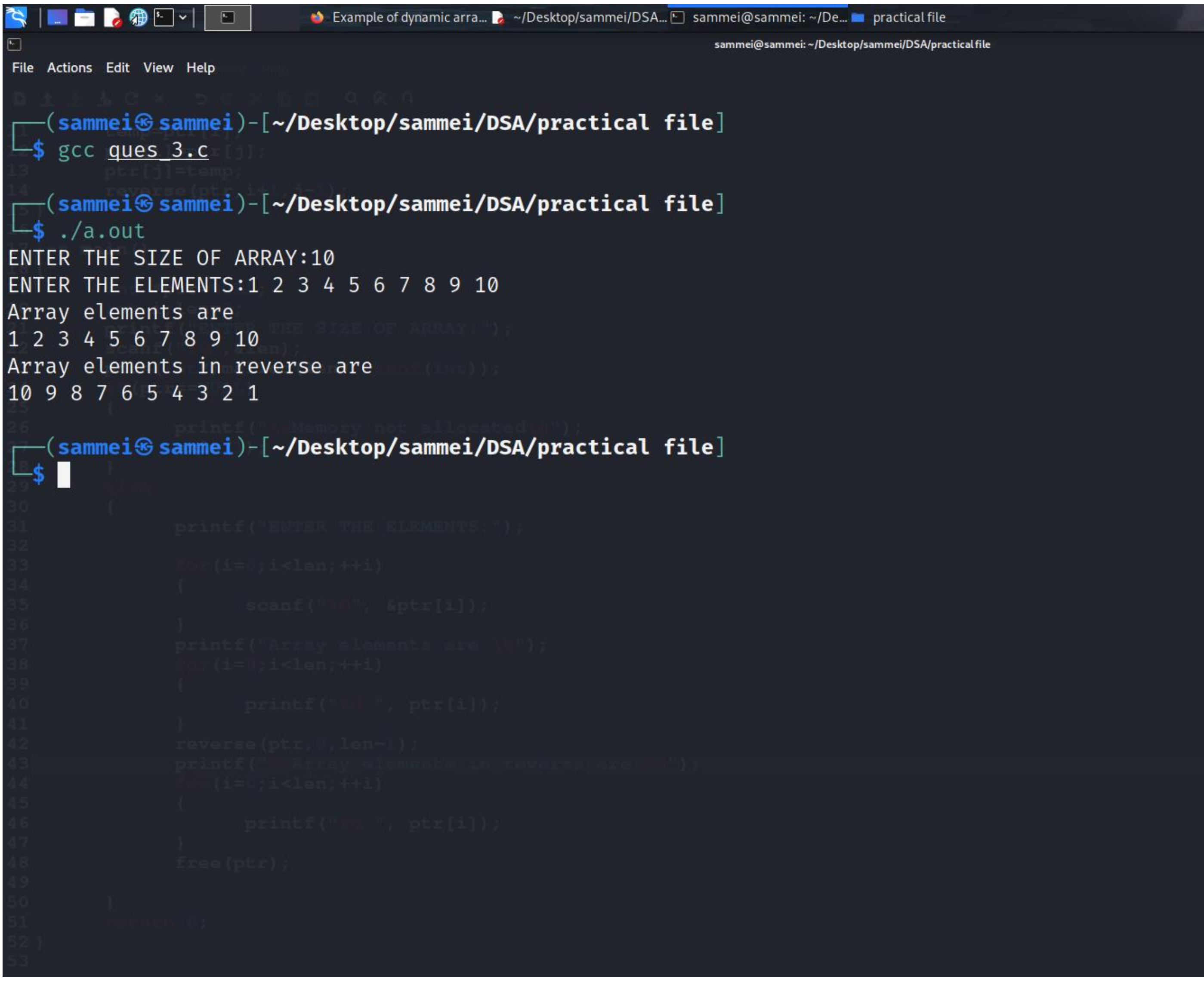
Q3. Write a C program Create a Dynamic array and then Reverse the array using recursion and then finally print the array.

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>

            void reverse(int *ptr,int i,int j)
            {
                   int temp;
                   if(i>=j)
                   {
                          return;
                   }
                   temp=ptr[i];
                   ptr[i]=ptr[j];
                   ptr[j]=temp;
                   reverse(ptr,i+1,j-1);
            }

            int main()
            {
                   int *ptr=NULL;
                   int i,len=0;
                   printf("ENTER THE SIZE OF ARRAY:");
                   scanf("%d",&len);
                   ptr=(int*)malloc(len*sizeof(int));
                   if(ptr==NULL)
                   {
                          printf("\nMemory not allocated\n");
                          exit(1);
                   }
                   else
                   {
                          printf("ENTER THE ELEMENTS:");

                          for(i=0;i<len;++i)
                          {
                                 scanf("%d", &ptr[i]);
                          }
                          printf("Array elements are \n");
                          for(i=0;i<len;++i)
```

```
                {
                        printf("%d ", ptr[i]);
                }
                reverse(ptr,0,len-1);
                printf("\nArray elements in reverse are \n");
                for(i=0;i<len;++i)
                {
                        printf("%d ", ptr[i]);
                }
                free(ptr);

        }
        return 0;
}
```

Output:



```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ gcc ques_3.c

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
ENTER THE SIZE OF ARRAY:10
ENTER THE ELEMENTS:1 2 3 4 5 6 7 8 9 10
Array elements are
1 2 3 4 5 6 7 8 9 10
Array elements in reverse are
10 9 8 7 6 5 4 3 2 1

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

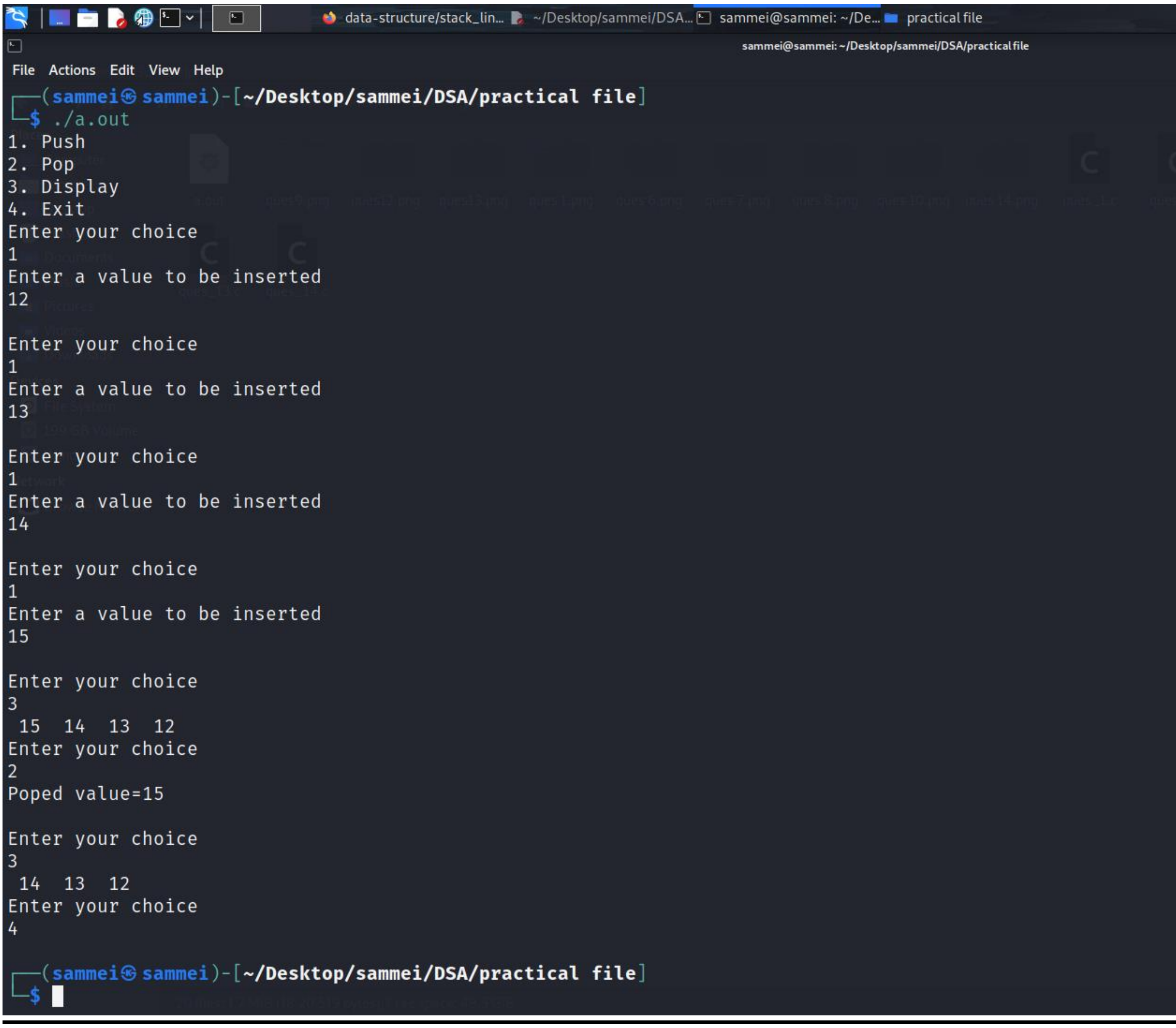
Q4. Write a C Program implement STACK using array in menu driven form.

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>


            typedef struct node
            {
                   int info;
                   struct node *next;
            }node;

            void push(node **top,int value)
            {
                   node *temp;
                   temp=(node*)malloc(sizeof(node));
                   temp->info=value;
                   temp->next=NULL;
                   if(*top==NULL)
                          *top=temp;
                   else
                   {
                          temp->next=*top;
                          *top=temp;
                   }
            }

            void display(node **top)
            {
                   node *temp=*top;
                   while(temp!=NULL)
                   {
                          printf(" %d ",temp->info);
                          temp=temp->next;
                   }
            }

            void pop(node **top)
            {
                   node *temp=*top;
```

```
        *top=temp->next;
        temp->next=NULL;
        printf("Poped value=%d\n",temp->info);
        free(temp);
}

int main()
{
      node *top=NULL;
        int c=0,value;
        printf("1. Push\n");
      printf("2. Pop\n");
      printf("3. Display\n");
      printf("4. Exit\n");
        while(c!=4)
        {
         printf("Enter your choice\n");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:
                        {
                                printf("Enter a value to be inserted\n");
                                scanf("%d",&value);
                                push(&top,value);
                                printf("\n");
                                break;
                        }
                        case 2:
                        {
                                pop(&top);
                                printf("\n");
                                break;
                        }
                        case 3:
                        {
                                display(&top);
                                printf("\n");
                                break;
                        }
                }
        }
        return 0;
}
```

Output:


data-structure/stack_lin...
~/Desktop/sammei/DSA...
sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

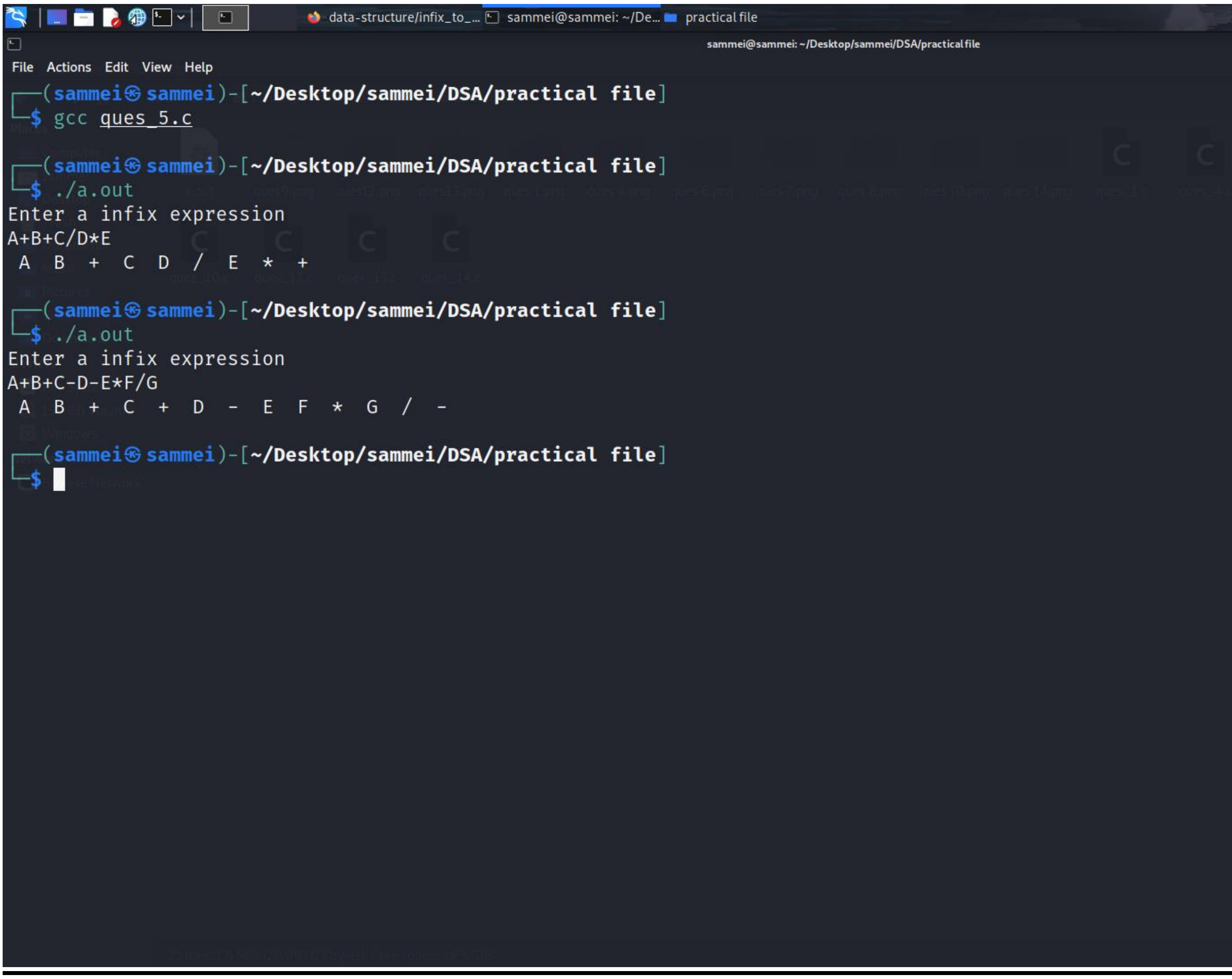
```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
1. Push
2. Pop
3. Display
4. Exit
Enter your choice
1
Enter a value to be inserted
12

Enter your choice
1
Enter a value to be inserted
13

Enter your choice
1
Enter a value to be inserted
14

Enter your choice
1
Enter a value to be inserted
15

Enter your choice
3
 15  14  13  12
Enter your choice
2
Poped value=15

Enter your choice
3
 14  13  12
Enter your choice
4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ▮
```

Q5. Write a C Program to Convert Infix to Postfix Expression using Stack.

Solution:

```
#include
<stdio.h
>

           int push(char *stack,int top,int max,char c)
           {
                  if(stack[top]==max-1)
                         printf("Stack is Full\n");
                  else
                  {
                         top=top+1;
                         stack[top]=c;
                  }
                  return top;
           }

           int pop(char *stack,int top)
           {
                  if(top==-1)
                         printf("Stack is empty\n");
                  else
                  {
                         printf(" %c ",stack[top]);
                         top=top-1;
                  }
                  return top;
           }

           int main()
           {
                  int top=-1,i=0,c,max=20;
                  char input[max],stack[max];
                  printf("Enter a infix expression\n");
                  scanf("%s",input);
                  while(input[i]!='\0')
                  {
                         if(input[i]=='+'||input[i]=='-')
                         {
                                if(top==-1||stack[top]=='(')
                        {
                                top=push(stack,top,max,input[i]);
                        }
                        else
if(stack[top]=='+'||stack[top]==''||stack[top]=='*'||stack[top]=='/'||stack[t
op]=='^')
                        {
                                top=pop(stack,top);
                                i-=1;;
                        }
                }
                else if(input[i]=='*'||input[i]=='/')
                {

        if(top==1||stack[top]=='('||stack[top]=='+'||stack[top]=='-')
                        {
                                top=push(stack,top,max,input[i]);
                        }
                        else
if(stack[top]=='*'||stack[top]=='/'||stack[top]=='^')
                        {
                                top=pop(stack,top);
                                i-=1;
                        }
                }
                else if(input[i]=='^')
                {
                        top=push(stack,max,top,input[i]);
                }
                else if(input[i]=='(')
                {
                        top=push(stack,max,top,input[i]);
                }
                else if(input[i]==')')
                {
                        while(top!=-1 && stack[top]!='(')
                                top=pop(stack,top);
                        if(top==-1 || stack[top]!='(')
                                printf("Error.......( missing\n");
                        else
                                top=pop(stack,top);
                }
                else
                        printf(" %c ",input[i]);
                i++;
        }
        while(top!=-1)
                top=pop(stack,top);
        printf("\n");
        return 0;
}
```

Output:



data-structure/infix_to_...
sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ gcc ques_5.c

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
Enter a infix expression
A+B+C/D*E
 A  B  +  C  D  /  E  *  +

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out
Enter a infix expression
A+B+C-D-E*F/G
 A  B  +  C  +  D  -  E  F  *  G  /  -

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

Q6. Write a C Program to create singly linked list by adding nodes in the right-hand side and delete alternate node from the list and then print the final list.

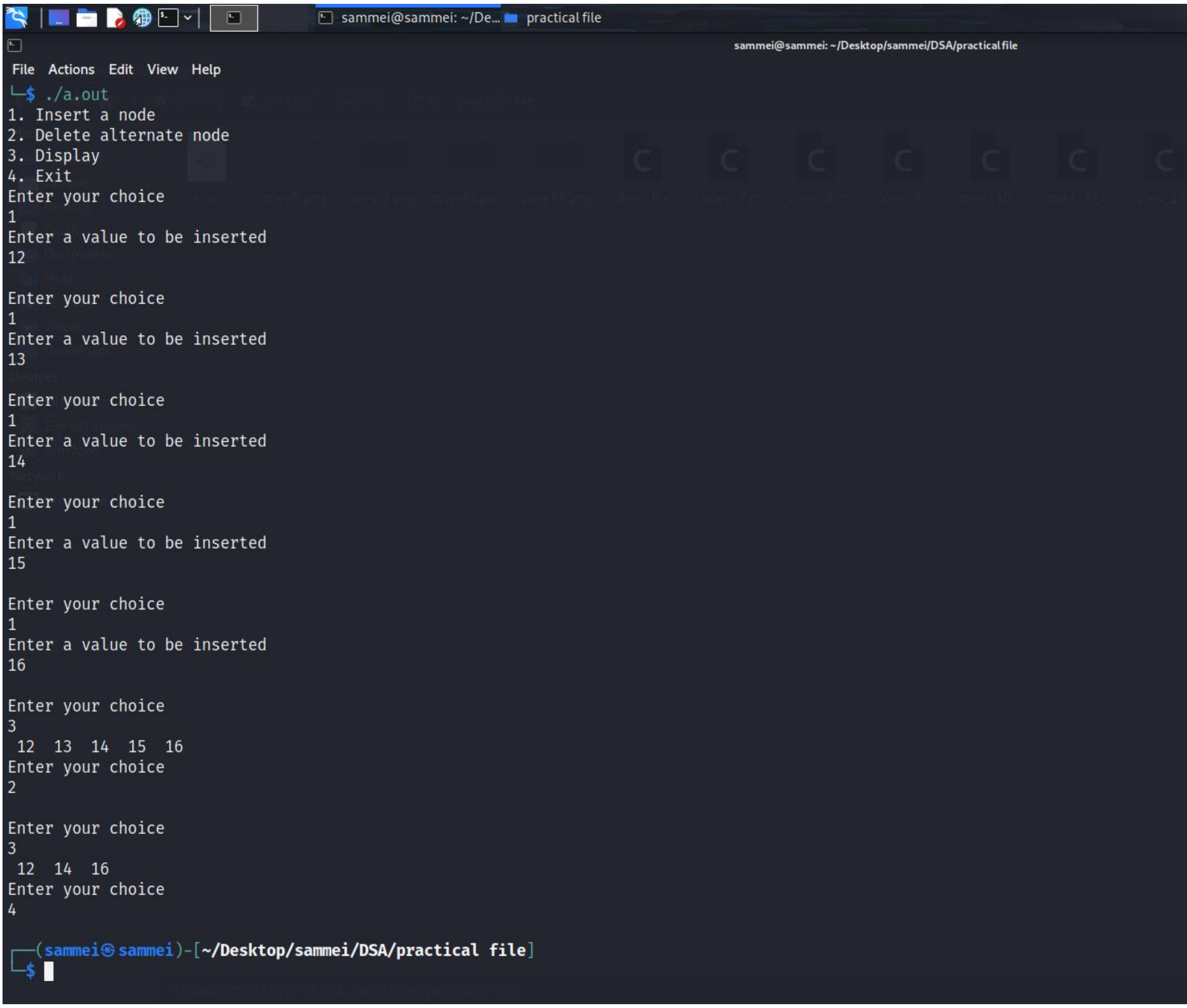
Solution:

```
#include
<stdio.h>
            #include <stdlib.h>


            typedef struct node
            {
                   int info;
                   struct node *next;
            }node;

            void insert(node **head,int value)
            {
                   node *temp;
                   temp=(node*)malloc(sizeof(node));
                   temp->info=value;
                   temp->next=NULL;
                   if(*head==NULL)
                          *head=temp;
                   else
                   {
                          node *last=*head;
                          while(last->next!=NULL)
                                 last=last->next;
                          last->next=temp;
                          last=temp;
                   }
            }

            void display(node *head)
            {
                   node *temp=head;
                   while(temp!=NULL)
                   {
                          printf(" %d ",temp->info);
                          temp=temp->next;
                   }
            }
```

```
void deletenode(node *head)
{
    if (head == NULL)
        return;
    node *prev = head;
    node *node = head->next;

    while (prev != NULL && node != NULL)
    {
        prev->next = node->next;

        prev = prev->next;
        if (prev != NULL)
            node = prev->next;
    }
}
int main()
{
      node *head=NULL;
       int c=0,value;
       printf("1. Insert a node\n");
      printf("2. Delete alternate node\n");
      printf("3. Display\n");
      printf("4. Exit\n");
       while(c!=4)
       {
        printf("Enter your choice\n");
               scanf("%d",&c);
               switch(c)
               {
                      case 1:
                      {
                             printf("Enter a value to be inserted\n");
                             scanf("%d",&value);
                             insert(&head,value);
                             printf("\n");
                             break;
                      }
                      case 2:
                      {
                             deletenode(head);
                             printf("\n");
                             break;
                      }
                      case 3:
```

```
            {
                display(head);
                printf("\n");
                break;
            }
        }
    }
    return 0;
}
```

Output:



```
└─$ ./a.out
1. Insert a node
2. Delete alternate node
3. Display
4. Exit
Enter your choice
1
Enter a value to be inserted
12

Enter your choice
1
Enter a value to be inserted
13

Enter your choice
1
Enter a value to be inserted
14

Enter your choice
1
Enter a value to be inserted
15

Enter your choice
1
Enter a value to be inserted
16

Enter your choice
3
 12  13  14  15  16
Enter your choice
2

Enter your choice
3
 12  14  16
Enter your choice
4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ▌
```

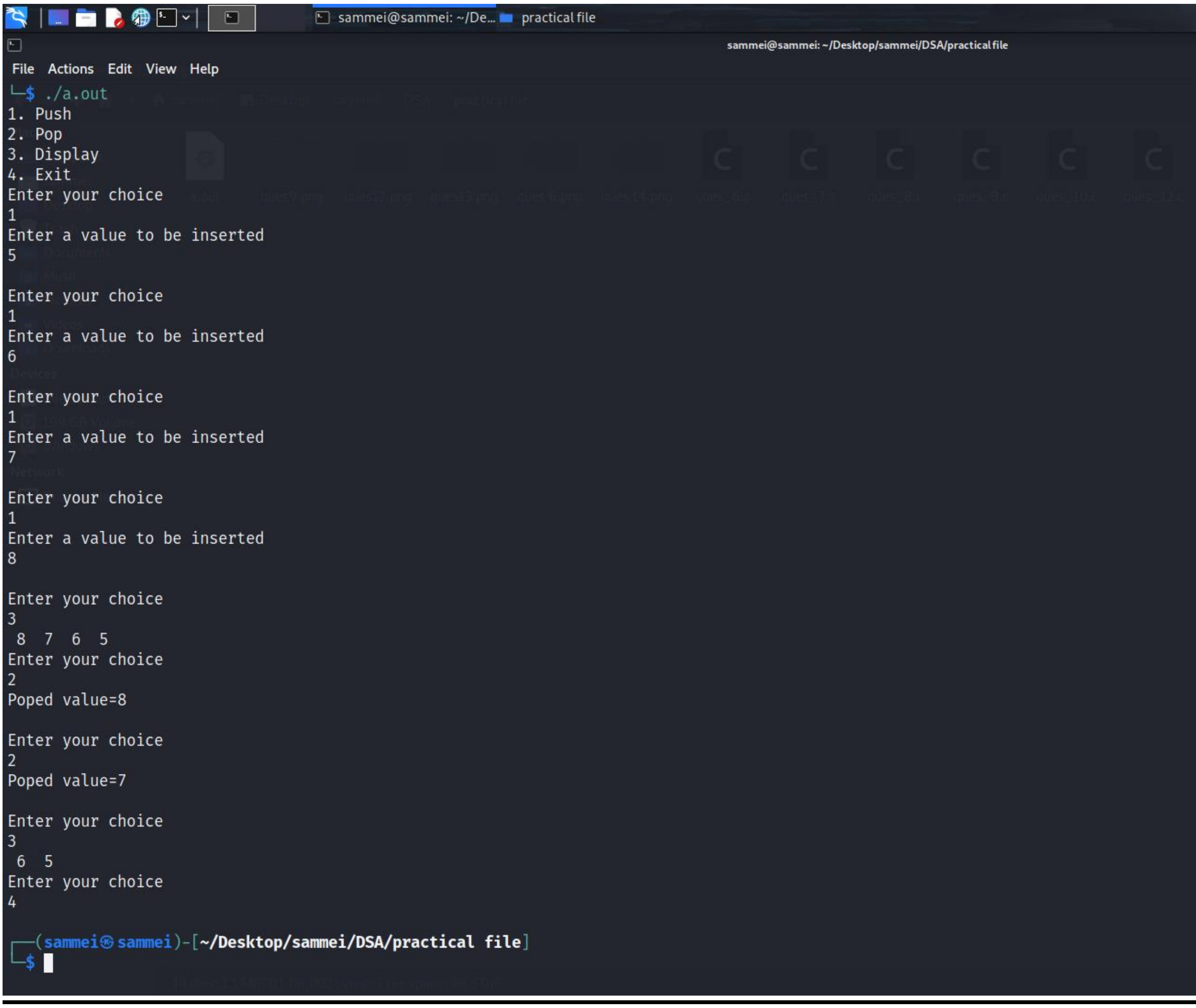
Q7. Write a C Program implement STACK using Link List in menu driven form.

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>


            typedef struct node
            {
                   int info;
                   struct node *next;
            }node;

            void push(node **top,int value)
            {
                   node *temp;
                   temp=(node*)malloc(sizeof(node));
                   temp->info=value;
                   temp->next=NULL;
                   if(*top==NULL)
                          *top=temp;
                   else
                   {
                          temp->next=*top;
                          *top=temp;
                   }
            }

            void display(node **top)
            {
                   node *temp=*top;
                   while(temp!=NULL)
                   {
                          printf(" %d ",temp->info);
                          temp=temp->next;
                   }
            }

            void pop(node **top)
            {
                   node *temp=*top;
```

```
        *top=temp->next;
        temp->next=NULL;
        printf("Poped value=%d\n",temp->info);
        free(temp);
}

int main()
{
    node *top=NULL;
        int c=0,value;
        printf("1. Push\n");
      printf("2. Pop\n");
      printf("3. Display\n");
      printf("4. Exit\n");
        while(c!=4)
        {
         printf("Enter your choice\n");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:
                        {
                                printf("Enter a value to be inserted\n");
                                scanf("%d",&value);
                                push(&top,value);
                                printf("\n");
                                break;
                        }
                        case 2:
                        {
                                pop(&top);
                                printf("\n");
                                break;
                        }
                        case 3:
                        {
                                display(&top);
                                printf("\n");
                                break;
                        }
                }
        }
        return 0;
}
```

Output:



sammei@sammei: ~/De... practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
└─$ ./a.out
1. Push
2. Pop
3. Display
4. Exit
Enter your choice
1
Enter a value to be inserted
5

Enter your choice
1
Enter a value to be inserted
6

Enter your choice
1
Enter a value to be inserted
7

Enter your choice
1
Enter a value to be inserted
8

Enter your choice
3
 8  7  6  5
Enter your choice
2
Poped value=8

Enter your choice
2
Poped value=7

Enter your choice
3
 6  5
Enter your choice
4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

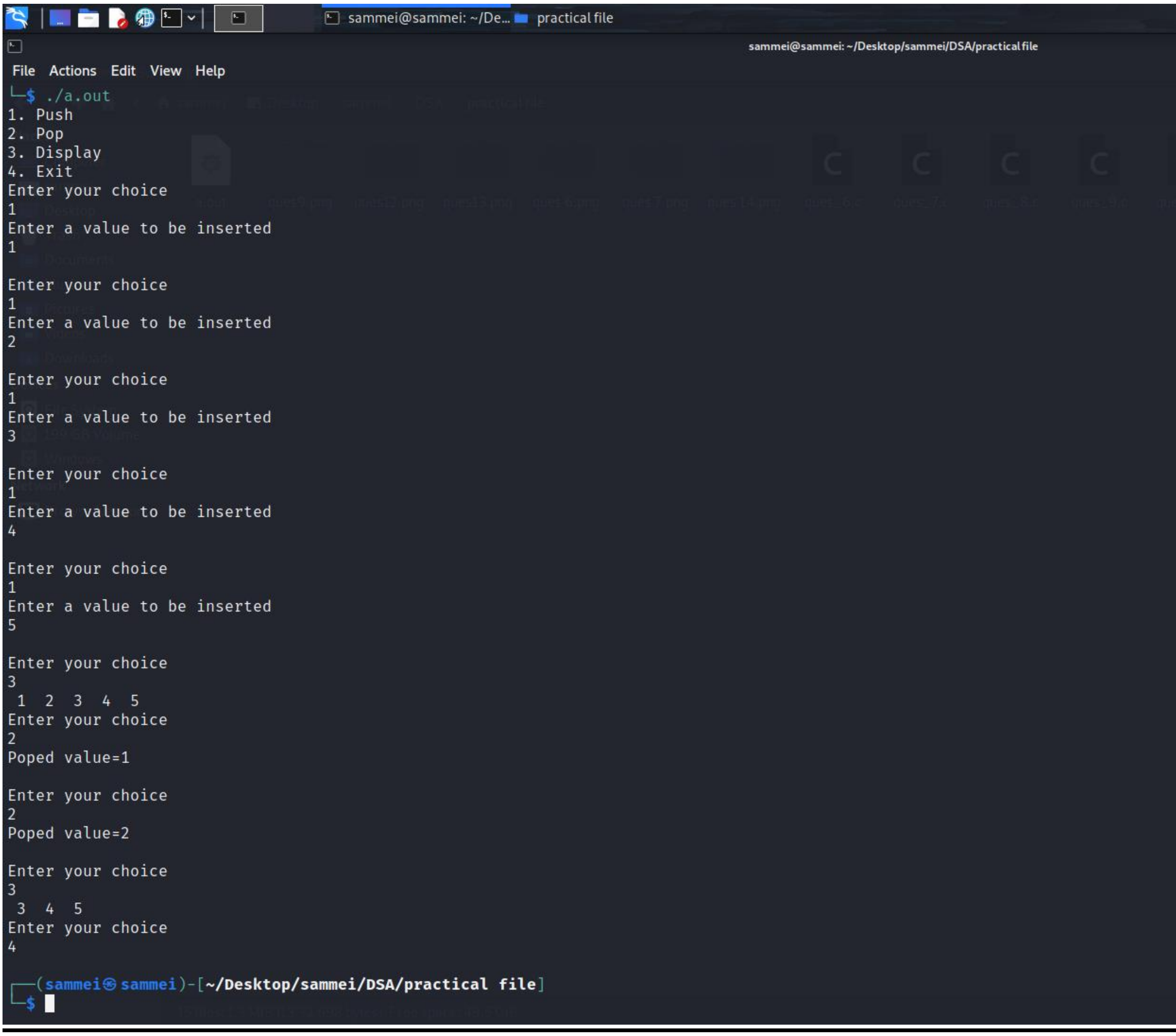
Q8. Write a C Program implement QUEUE using Link List in menu driven form.

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>


            typedef struct node
            {
                   int info;
                   struct node *next;
            }node;

            void push(node **front,int value)
            {
                   node *temp;
                   temp=(node*)malloc(sizeof(node));
                   temp->info=value;
                   temp->next=NULL;
                   if(*front==NULL)
                          *front=temp;
                   else
                   {
                          node *last=*front;
                          while(last->next!=NULL)
                                 last=last->next;
                          last->next=temp;
                          last=temp;
                   }
            }

            void display(node **front)
            {
                   node *temp=*front;
                   while(temp!=NULL)
                   {
                          printf(" %d ",temp->info);
                          temp=temp->next;
                   }
            }
```

```
void pop(node **rear)
{
        node *temp=*rear;
        *rear=temp->next;
        temp->next=NULL;
        printf("Poped value=%d\n",temp->info);
        free(temp);
}

int main()
{
    node *head=NULL;
        int c=0,value;
        printf("1. Push\n");
      printf("2. Pop\n");
      printf("3. Display\n");
      printf("4. Exit\n");
        while(c!=4)
        {
         printf("Enter your choice\n");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:
                        {
                                printf("Enter a value to be inserted\n");
                                scanf("%d",&value);
                                push(&head,value);
                                printf("\n");
                                break;
                        }
                        case 2:
                        {
                                pop(&head);
                                printf("\n");
                                break;
                        }
                        case 3:
                        {
                                display(&head);
                                printf("\n");
                                break;
                        }
                }
        }
        return 0;
}
```

Output:



sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
└─$ ./a.out
1. Push
2. Pop
3. Display
4. Exit
Enter your choice
1
Enter a value to be inserted
1

Enter your choice
1
Enter a value to be inserted
2

Enter your choice
1
Enter a value to be inserted
3

Enter your choice
1
Enter a value to be inserted
4

Enter your choice
1
Enter a value to be inserted
5

Enter your choice
3
 1  2  3  4  5
Enter your choice
2
Poped value=1

Enter your choice
2
Poped value=2

Enter your choice
3
 3  4  5
Enter your choice
4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

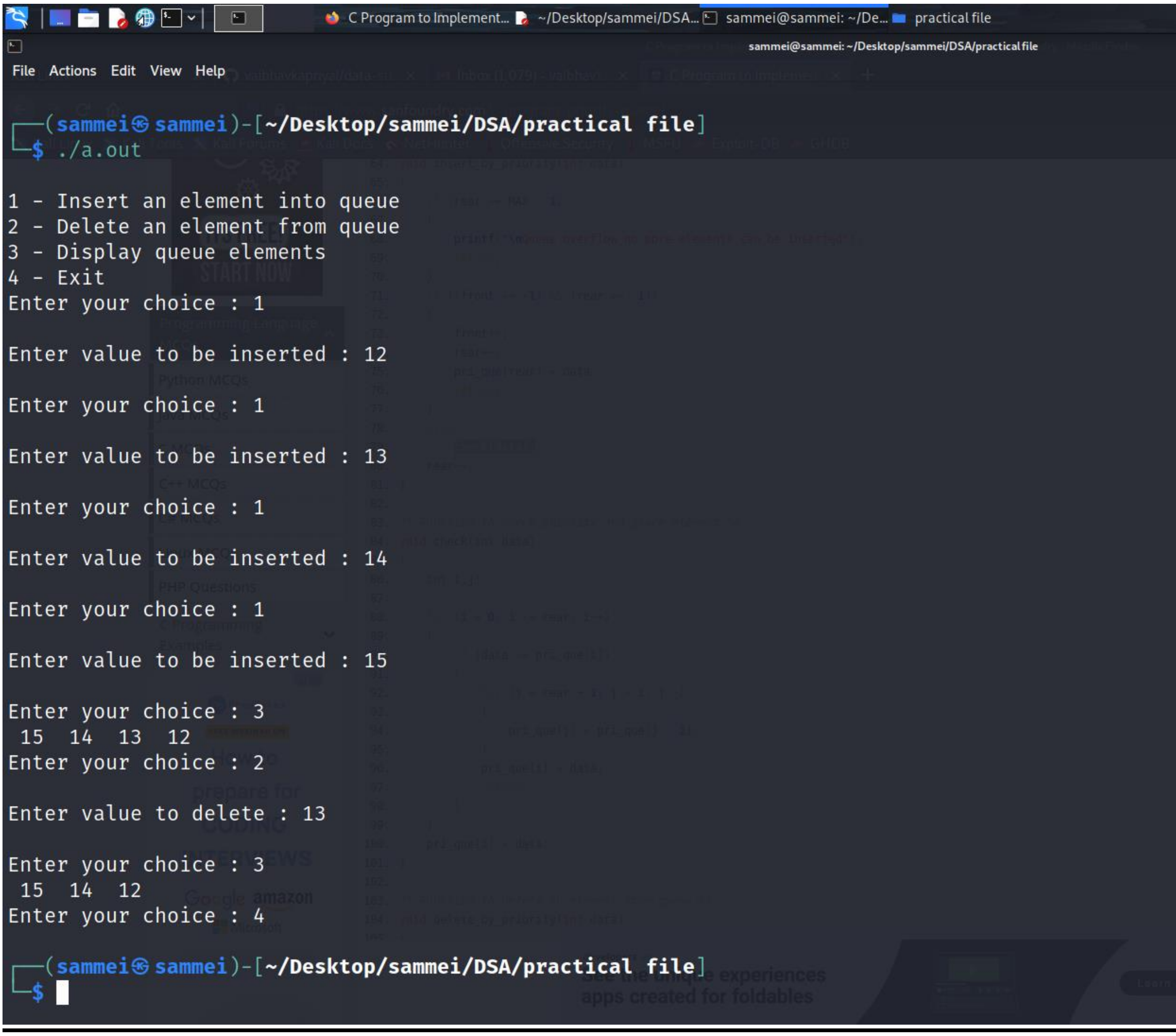
Q9. Write a C Program implement priority QUEUE using array in menu driven form.

<u>Solution:</u>

```
#include
<stdio.h>
            #include <stdlib.h>
            #define MAX 5

            int pri_que[MAX];
            int front, rear;

            void create()
            {
                front = rear = -1;
            }

            void insert_by_priority(int data)
            {
                if (rear >= MAX - 1)
                {
                    printf("\nQueue overflow no more elements can be inserted");
                    return;
                }
                if ((front == -1) && (rear == -1))
                {
                    front++;
                    rear++;
                    pri_que[rear] = data;
                    return;
                }
                else
                {
                   check(data);
                }
                rear++;
            }

            void check(int data)
            {
                int i,j;
                for (i = 0; i <= rear; i++)
                {
```

```
        if (data >= pri_que[i])
        {
            for (j = rear + 1; j > i; j--)
            {
                pri_que[j] = pri_que[j - 1];
            }
            pri_que[i] = data;
            return;
        }
    }
    pri_que[i] = data;
}

void delete_by_priority(int data)
{
    int i;
    if ((front==-1) && (rear==-1))
    {
        printf("\nQueue is empty no elements to delete");
        return;
    }
   for (i = 0; i <= rear; i++)
    {
        if (data == pri_que[i])
        {
            for (; i < rear; i++)
            {
                pri_que[i] = pri_que[i + 1];
            }
        pri_que[i] = -99;
        rear--;
       if (rear == -1)
            front = -1;
        return;
        }
    }
    printf("\n%d not found in queue to delete", data);
}

void display_pqueue()
{
    if ((front == -1) && (rear == -1))
    {
        printf("\nQueue is empty");
        return;
    }
    for (; front <= rear; front++)
    {
        printf(" %d ", pri_que[front]);
    }
    front = 0;
}

void main()
{
    int n, ch;
    printf("\n1 - Insert an element into queue");
    printf("\n2 - Delete an element from queue");
    printf("\n3 - Display queue elements");
    printf("\n4 - Exit");
    create();
    while (1)
    {
        printf("\nEnter your choice : ");
        scanf("%d", &ch);
      switch (ch)
        {
        case 1:
            printf("\nEnter value to be inserted : ");
            scanf("%d",&n);
            insert_by_priority(n);
            break;
        case 2:
            printf("\nEnter value to delete : ");
            scanf("%d",&n);
            delete_by_priority(n);
            break;
        case 3:
            display_pqueue();
            break;
        case 4:
            exit(0);
        default:
            printf("\nChoice is incorrect, Enter a correct choice");
        }
    }
}
```

Output:



C Program to Implement...
~/Desktop/sammei/DSA...
sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out

1 - Insert an element into queue
2 - Delete an element from queue
3 - Display queue elements
4 - Exit
Enter your choice : 1

Enter value to be inserted : 12

Enter your choice : 1

Enter value to be inserted : 13

Enter your choice : 1

Enter value to be inserted : 14

Enter your choice : 1

Enter value to be inserted : 15

Enter your choice : 3
 15   14   13   12
Enter your choice : 2

Enter value to delete : 13

Enter your choice : 3
 15   14   12
Enter your choice : 4

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ 
```

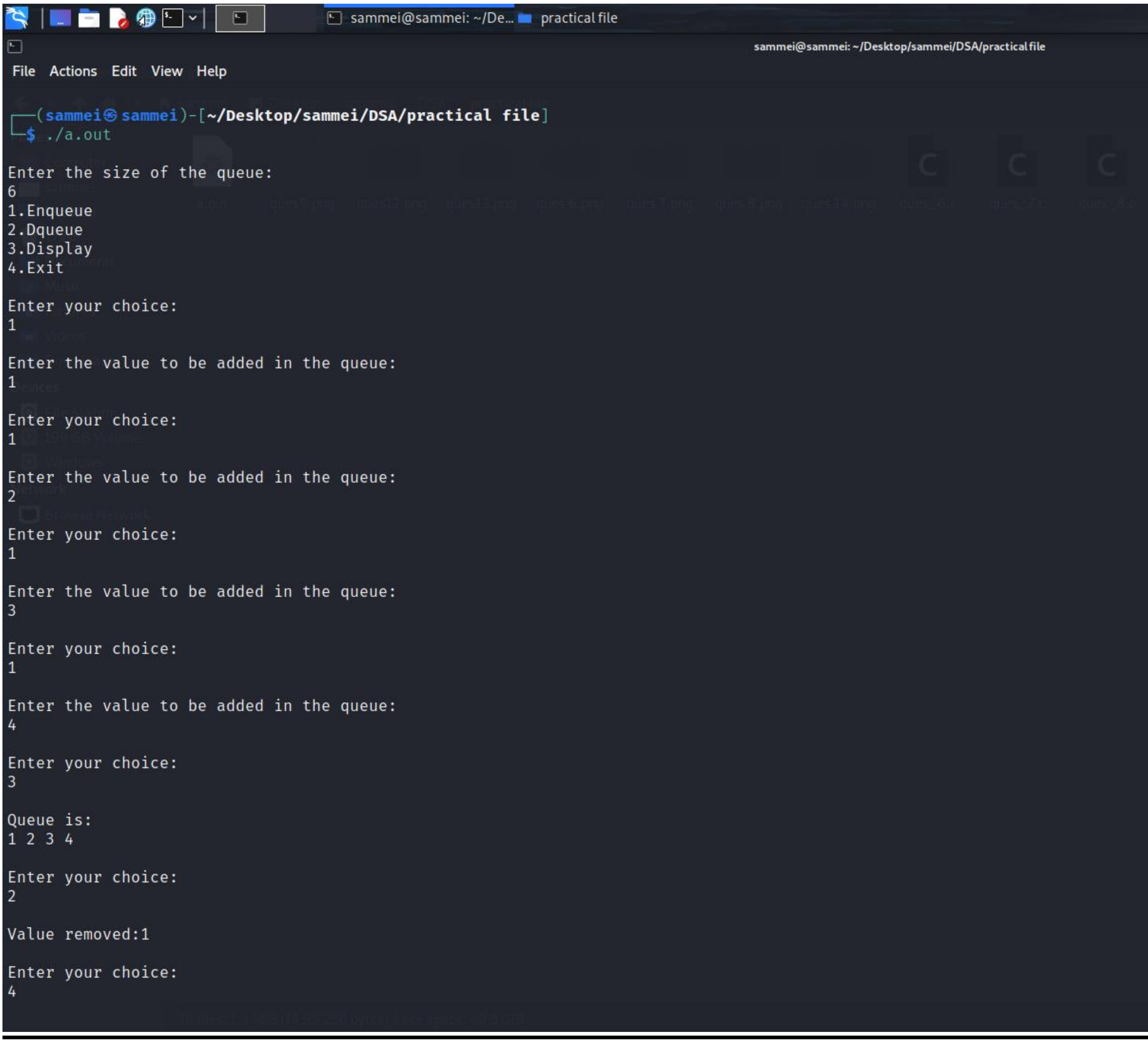
Q10. Write a C Program implement QUEUE using array in menu driven form.

Solution:

```
#include
<stdio.h>

            void Enqueue(int *a,int *f,int *r,int n)
            {
                    int val;
                    printf("\nEnter the value to be added in the queue:\n");
                    scanf("%d",&val);
                    if(*r==n-1)
                            printf("\nOverflow\n");
                    else
                    {
                            *r=*r+1;
                            a[*r]=val;
                            if(*f==-1)
                                    *f=0;
                    }
            }
            void Dqueue(int *a,int *f,int *r)
            {
                    if(*f==-1||*f>*r)
                            printf("\nUnderflow\n");
                    else
                    {
                            printf("\nValue removed:%d\n",a[*f]);
                            *f=*f+1;
                            if(*f>*r)
                                    *f=*r=-1;
                    }
            }
            void display(int *a,int *f,int *r)
            {
                    if(*f==-1)
                            printf("\nQueue is Empty\n");
                    else
                    {
                            printf("\nQueue is:\n");
                            for(int i=*f;i<=*r;i++)
                                    printf("%d ",a[i]);
```

```
                printf("\n");
        }
}
int main()
{
        int a[20];
        int f=-1,r=-1,c=0,n=0;
        printf("\nEnter the size of the queue:\n");
        scanf("%d",&n);
        printf("1.Enqueue\n");
        printf("2.Dqueue\n");
        printf("3.Display\n");
        printf("4.Exit\n");
        while(c!=4)
        {
                printf("\nEnter your choice:\n");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:
                        {
                                Enqueue(a,&f,&r,n);
                                break;
                        }
                        case 2:
                        {
                                Dqueue(a,&f,&r);
                                break;
                        }
                        case 3:
                        {
                                display(a,&f,&r);
                                break;
                        }
                        default:
                        {
                                printf("\nInvalid choice\n");
                                break;
                        }

                }
        }
}
```

Output:


sammei@sammei: ~/De...
practical file
sammei@sammei: ~/Desktop/sammei/DSA/practical file
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA/practical file]
└─$ ./a.out

Enter the size of the queue:
6
1.Enqueue
2.Dqueue
3.Display
4.Exit

Enter your choice:
1

Enter the value to be added in the queue:
1

Enter your choice:
1

Enter the value to be added in the queue:
2

Enter your choice:
1

Enter the value to be added in the queue:
3

Enter your choice:
1

Enter the value to be added in the queue:
4

Enter your choice:
3

Queue is:
1 2 3 4

Enter your choice:
2

Value removed:1

Enter your choice:
4
```

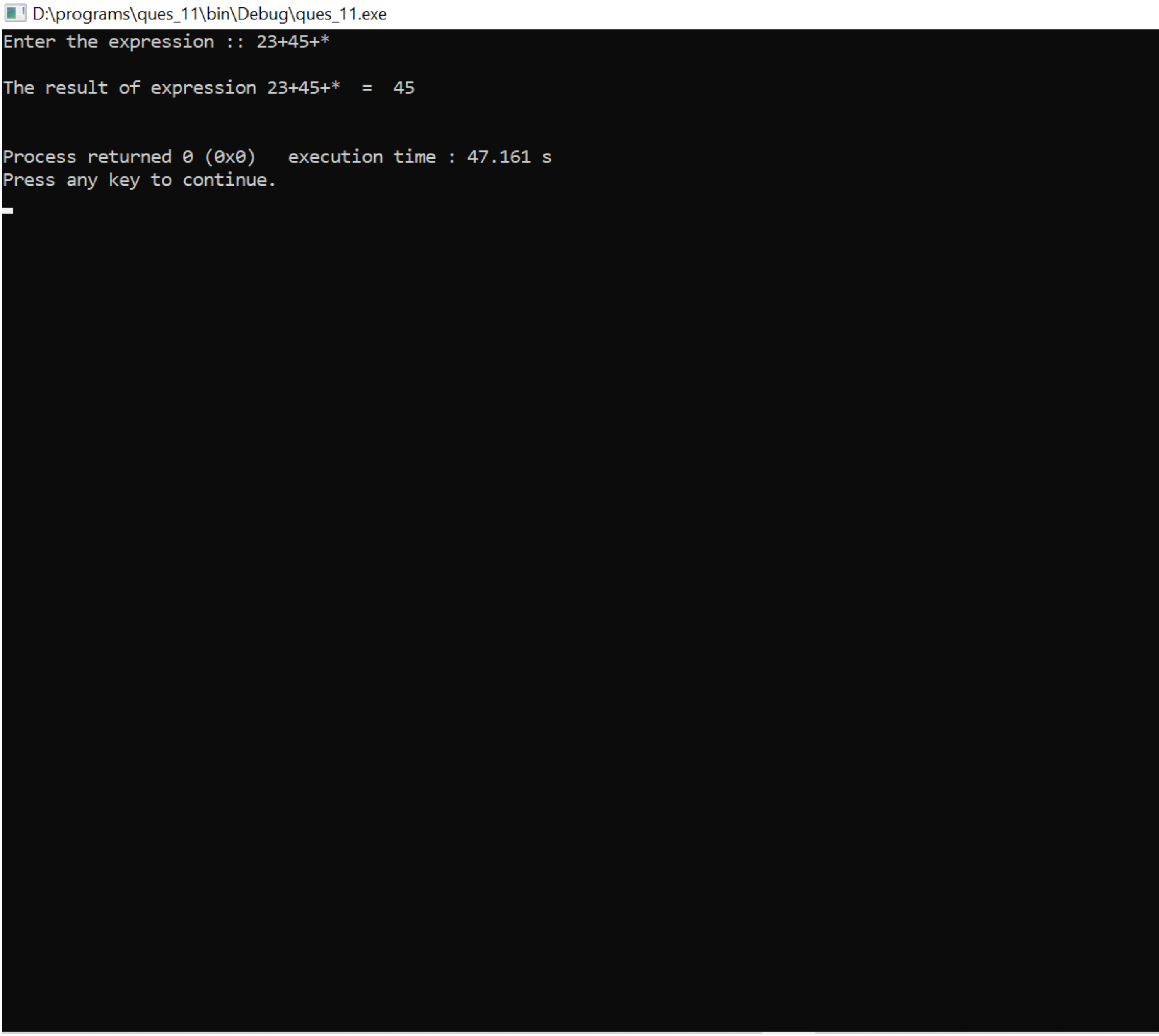
Q11. Write a C program to Evaluate Postfix Expression using Stack

Solution:

```
#include<stdio.h>
                    int stack[20];
                    int top = -1;

                    void push(int x)
                    {
                        stack[++top] = x;
                    }

                    int pop()
                    {
                        return stack[top--];
                    }

                    int main()
                    {
                        char exp[20];
                        int i=0;
                        int n1,n2,n3,num;
                        printf("Enter the expression :: ");
                        scanf("%s",exp);
                        while(exp[i] != '\0')
                        {
                            if(isdigit(exp[i]))
                            {
                                num = exp[i] - 48;
                                push(num);
                            }
                            else
                            {
                                n1 = pop();
                                n2 = pop();
                                switch(exp[i])
                                {
                                case '+':
                                {
                                    n3 = n1 + n2;
                                    break;
                                }
                                case '-':
```

```
            {
                n3 = n2 - n1;
                break;
            }
            case '*':
            {
                n3 = n1 * n2;
                break;
            }
            case '/':
            {
                n3 = n2 / n1;
                break;
            }
            }
            push(n3);
        }
        i++;
    }
    printf("\nThe result of expression %s  =  %d\n\n",exp,pop());
    return 0;
}
```

Output:



D:\programs\ques_11\bin\Debug\ques_11.exe

```
Enter the expression :: 23+45+*

The result of expression 23+45+*  =  45


Process returned 0 (0x0)   execution time : 47.161 s
Press any key to continue.
```

Q 12. Write a C program to create TWO singly linked list L1 and L2 and sort both the list and finally merge both the list such that L2 comes after L1. [use double pointer]

Solution:

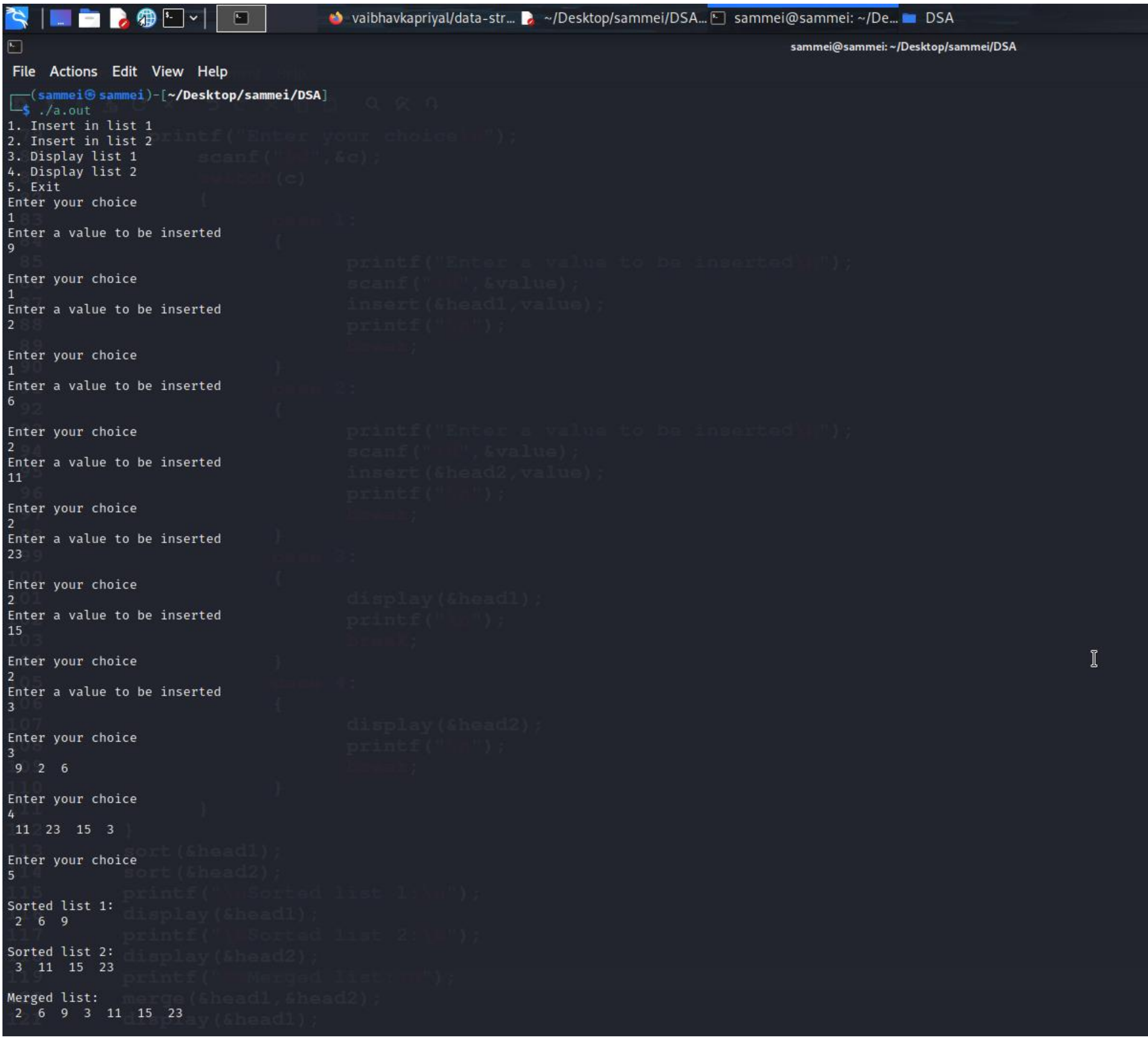
```
#include
<stdio.h>
            #include <stdlib.h>


            typedef struct node
            {
                    int info;
                    struct node *next;
            }node;

            void insert(node **head,int value)
            {
```

```
        node *temp;
        temp=(node*)malloc(sizeof(node));
        temp->info=value;
        temp->next=NULL;
        if(*head==NULL)
                *head=temp;
        else
        {
                node *last=*head;
                while(last->next!=NULL)
                        last=last->next;
                last->next=temp;
                last=temp;
        }
}

void display(node **head)
{
        node *temp=*head;
        while(temp!=NULL)
        {
                printf(" %d ",temp->info);
                temp=temp->next;
        }
        printf("\n");
}

void sort(node **head)
{
        int t;
        node *temp1=*head;
        while(temp1!=NULL)
        {
                node *temp2=*head;
                while(temp2!=NULL)
                {
                        if(temp1->info<temp2->info)
                        {
                                t=temp1->info;
                                temp1->info=temp2->info;
                                temp2->info=t;
                        }
                        temp2=temp2->next;
                }
                temp1=temp1->next;
        }
```

```
}
void merge(node **head1,node **head2)
{
       node *last=*head1;
       while(last->next!=NULL)
              last=last->next;
       last->next=*head2;
}
int main()
{
      node *head1=NULL;
      node *head2=NULL;
       int c=0,value;
       printf("1. Insert in list 1\n");
      printf("2. Insert in list 2\n");
      printf("3. Display list 1\n");
      printf("4. Display list 2\n");
      printf("5. Exit\n");
       while(c!=5)
       {
        printf("Enter your choice\n");
              scanf("%d",&c);
              switch(c)
              {
                     case 1:
                     {
                            printf("Enter a value to be inserted\n");
                            scanf("%d",&value);
                            insert(&head1,value);
                            printf("\n");
                            break;
                     }
                     case 2:
                     {
                            printf("Enter a value to be inserted\n");
                            scanf("%d",&value);
                            insert(&head2,value);
                            printf("\n");
                            break;
                     }
                     case 3:
                     {
                            display(&head1);
                            printf("\n");
                            break;
                     }
```

```
                case 4:
                {
                        display(&head2);
                        printf("\n");
                        break;
                }
        }
    }
    sort(&head1);
    sort(&head2);
    printf("\nSorted list 1:\n");
    display(&head1);
    printf("\nSorted list 2:\n");
    display(&head2);
    printf("\nMerged list:\n");
    merge(&head1,&head2);
    display(&head1);

    return 0;
}
```

Output:


vaibhavkapriyal/data-str...
~/Desktop/sammei/DSA...
sammei@sammei: ~/De...
DSA
sammei@sammei: ~/Desktop/sammei/DSA
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ ./a.out
1. Insert in list 1
2. Insert in list 2
3. Display list 1
4. Display list 2
5. Exit
Enter your choice
1
Enter a value to be inserted
9

Enter your choice
1
Enter a value to be inserted
2

Enter your choice
1
Enter a value to be inserted
6

Enter your choice
2
Enter a value to be inserted
11

Enter your choice
2
Enter a value to be inserted
23

Enter your choice
2
Enter a value to be inserted
15

Enter your choice
2
Enter a value to be inserted
3

Enter your choice
3
 9  2  6

Enter your choice
4
 11  23  15  3

Enter your choice
5

Sorted list 1:
 2  6  9

Sorted list 2:
 3  11  15  23

Merged list:
 2  6  9  3  11  15  23
```

Q 13. Write C program to create a doubly link list by adding the node right hand side and then check list is in palindrome form or not.

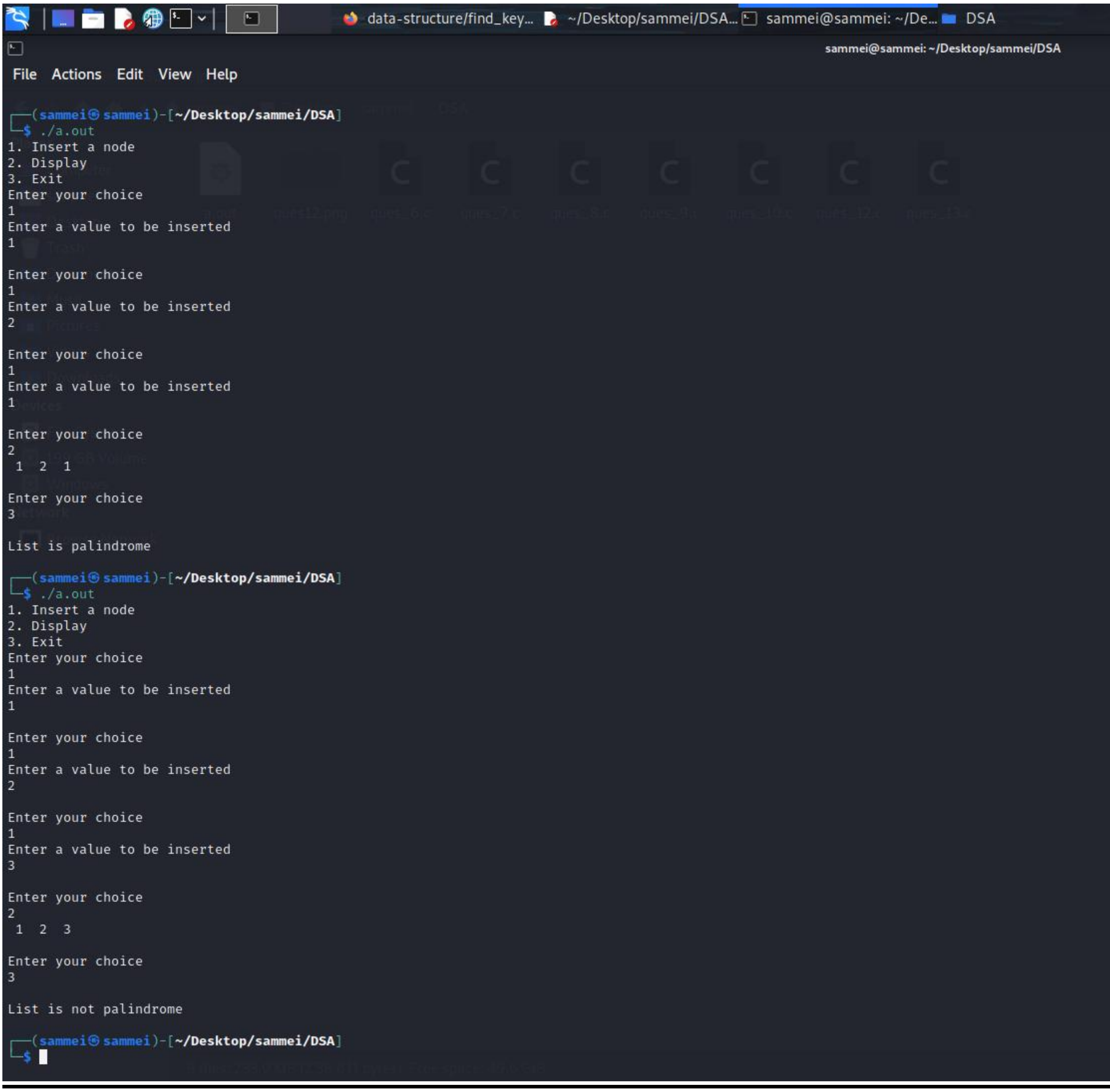
Solution:

```
#include
<stdio.h>
            #include <stdlib.h>

            typedef struct node{
                    int info;
                    struct node *next;
                    struct node *prev;
            }node;

            void insert(node**head,node **tail,int value)
            {
                    node *temp;
                    temp=(node*)malloc(sizeof(node));
                    temp->info=value;
                    temp->next=NULL;
                    if(*tail==NULL)
                    {
                            *head=temp;
                            *tail=temp;
                            temp->prev=NULL;
                    }
                    else
                    {
                            (*tail)->next=temp;
                            temp->prev=*tail;
                            *tail=temp;
                    }
            }

            void display(node **head)
            {
                    node *temp=*head;
                    while(temp!=NULL)
                    {
                            printf(" %d ",temp->info);
                            temp=temp->next;
                    }
                    printf("\n");
            }

            void checkpalin(node **head,node **tail)
```

```
{
        node *temp1=*head;
        node *temp2=*tail;
        while(temp1->next!=NULL&&temp2->prev!=NULL)
        {
                if(temp1->info!=temp2->info)
                {
                        printf("\nList is not palindrome\n");
                        return;
                }
                temp2=temp2->prev;
                temp1=temp1->next;
        }
        printf("\nList is palindrome\n");
}

int main()
{
        node *head=NULL;
        node *tail=NULL;
        int c=0,value;
        printf("1. Insert a node\n");
       printf("2. Display\n");
       printf("3. Exit\n");
        while(c!=3)
        {
         printf("Enter your choice\n");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:
                        {
                                printf("Enter a value to be inserted\n");
                                scanf("%d",&value);
                                insert(&head,&tail,value);
                                printf("\n");
                                break;
                        }
                        case 2:
                        {
                                display(&head);
                                printf("\n");
                                break;
                        }
                }
        }
```

```
            checkpalin(&head,&tail);
            return 0;
    }
```

Output:



data-structure/find_key...
~/Desktop/sammei/DSA...
sammei@sammei: ~/De...
DSA
sammei@sammei: ~/Desktop/sammei/DSA
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ ./a.out
1. Insert a node
2. Display
3. Exit
Enter your choice
1
Enter a value to be inserted
1

Enter your choice
1
Enter a value to be inserted
2

Enter your choice
1
Enter a value to be inserted
1

Enter your choice
2
 1  2  1

Enter your choice
3

List is palindrome

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ ./a.out
1. Insert a node
2. Display
3. Exit
Enter your choice
1
Enter a value to be inserted
1

Enter your choice
1
Enter a value to be inserted
2

Enter your choice
1
Enter a value to be inserted
3

Enter your choice
2
 1  2  3

Enter your choice
3

List is not palindrome

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ 
```

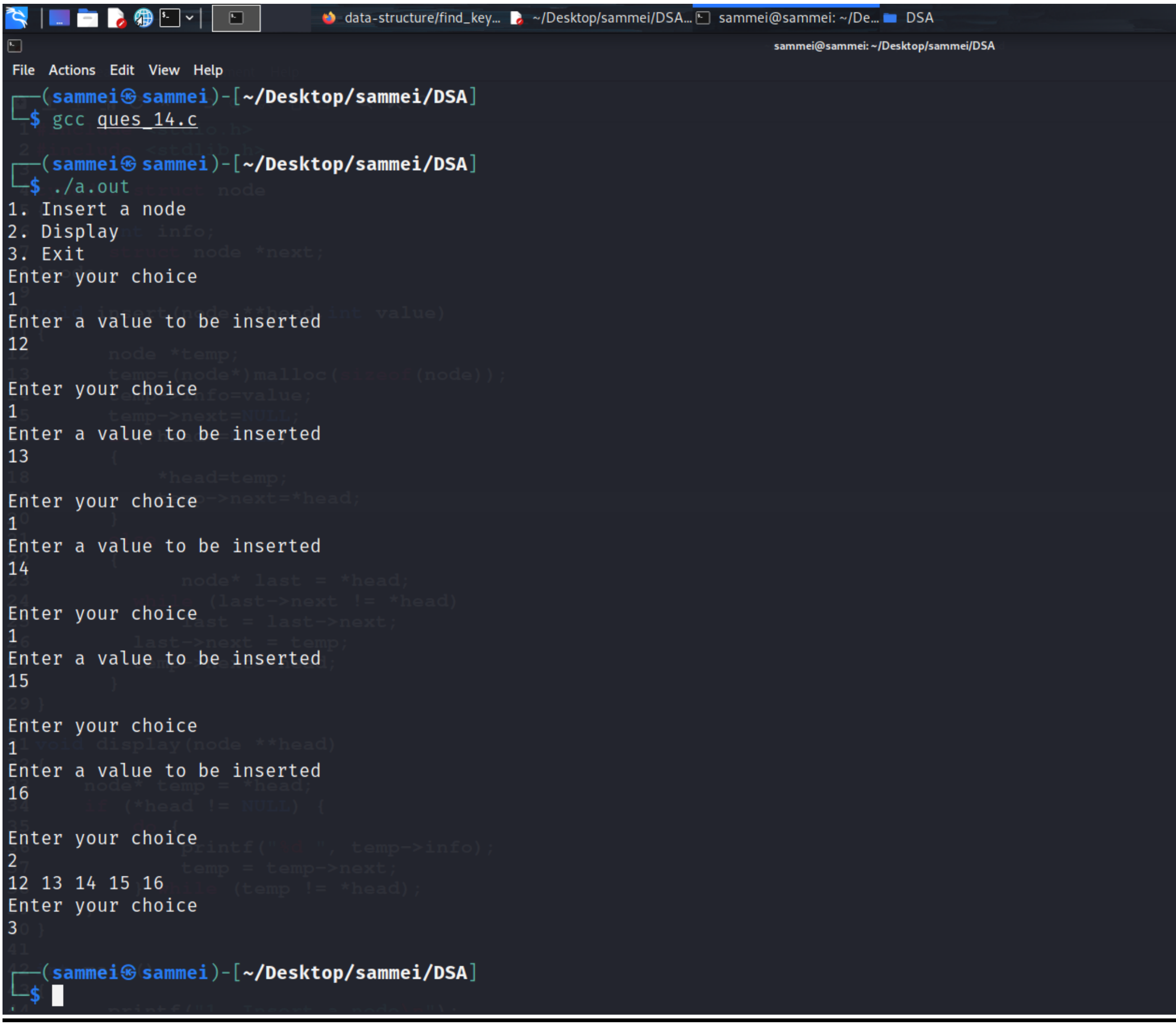
Q14. Write a C program to create a circular link list by adding the nodes in right hand side and then print the list.

Solution:

```
#include
<stdio.h>
            #include <stdlib.h>

            typedef struct node
            {
                   int info;
                   struct node *next;
            }node;

            void insert(node **head,int value)
            {
                   node *temp;
                   temp=(node*)malloc(sizeof(node));
                   temp->info=value;
                   temp->next=NULL;
                   if(*head==NULL)
                   {
                      *head=temp;
                       temp->next=*head;
                   }
                   else
                   {
                          node* last = *head;
                    while (last->next != *head)
                        last = last->next;
                    last->next = temp;
                    temp->next=*head;
                   }
            }

            void display(node **head)
            {
                node* temp = *head;
                if (*head != NULL) {
                    do {
                        printf("%d ", temp->info);
                        temp = temp->next;
                    } while (temp != *head);
```

```
    }
}

int main()
{
       printf("1. Insert a node\n");
      printf("2. Display\n");
      printf("3. Exit\n");
       node *head=NULL;
       int c=0,value;
       while(c!=3)
       {
        printf("Enter your choice\n");
              scanf("%d",&c);
              switch(c)
              {
                     case 1:
                     {
                            printf("Enter a value to be inserted\n");
                            scanf("%d",&value);
                            insert(&head,value);
                            printf("\n");
                            break;
                     }
                     case 2:
                     {
                            display(&head);
                            printf("\n");
                            break;
                     }
              }
       }
       return 0;
}
```

Output:



data-structure/find_key...
~/Desktop/sammei/DSA...
sammei@sammei: ~/De...
DSA
sammei@sammei: ~/Desktop/sammei/DSA
File Actions Edit View Help

```
┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ gcc ques_14.c

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ ./a.out
1. Insert a node
2. Display
3. Exit
Enter your choice
1
Enter a value to be inserted
12

Enter your choice
1
Enter a value to be inserted
13

Enter your choice
1
Enter a value to be inserted
14

Enter your choice
1
Enter a value to be inserted
15

Enter your choice
1
Enter a value to be inserted
16

Enter your choice
2
12 13 14 15 16
Enter your choice
3

┌──(sammei㉿sammei)-[~/Desktop/sammei/DSA]
└─$ 
```