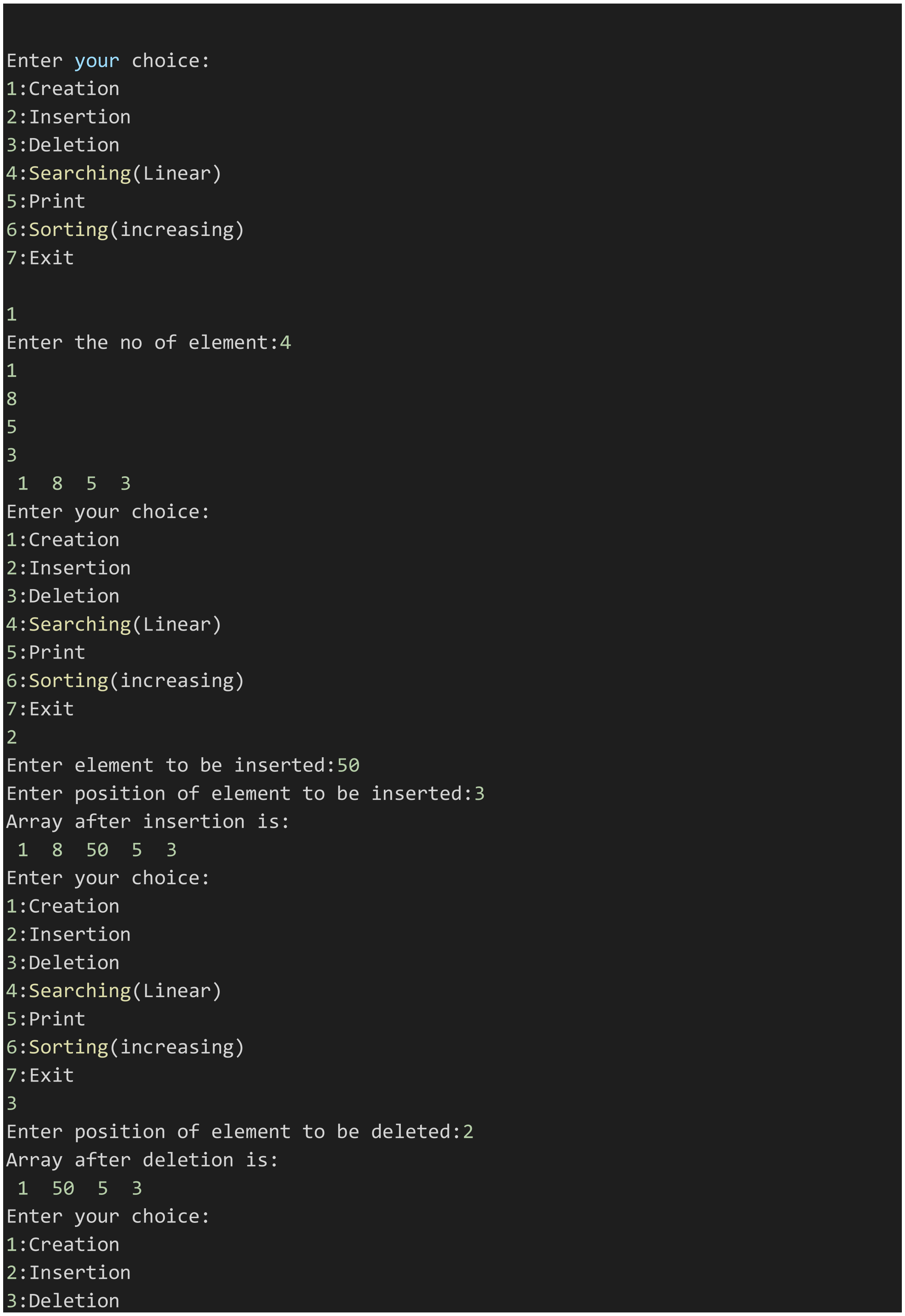
Q1-

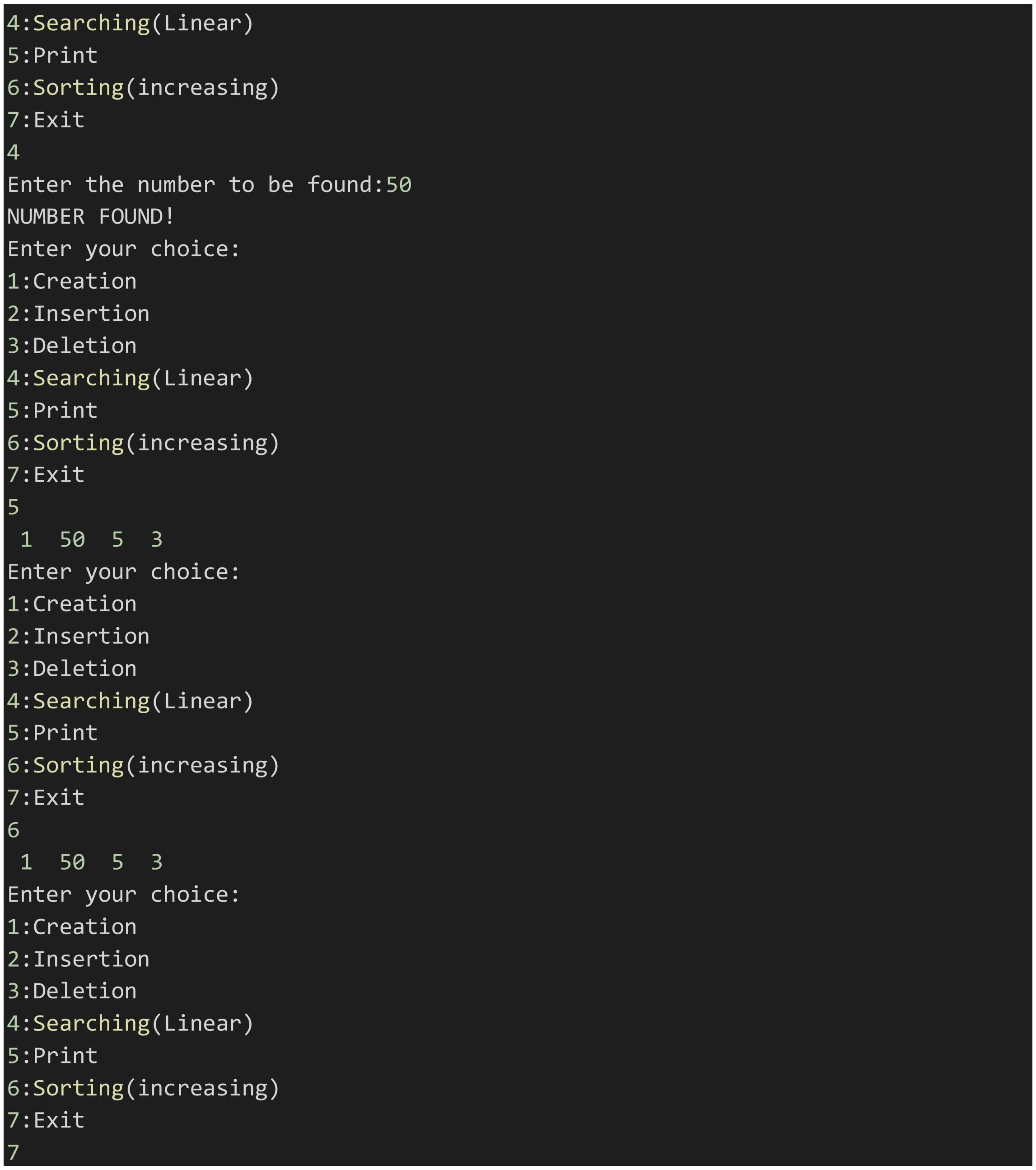
```c
#include<stdio.h>
#include<stdlib.h>

void creation(int a[],int n)
{
int i;
for(i=0;i<n;i++)
scanf("%d",&a[i]);
}

void print(int a[],int n)
{
int i;
for(i=0;i<=n-1;i++)
printf(" %d ",a[i]);
}

void insert(int a[],int n)
{
int i,ele,pos;
printf("Enter element to be inserted:");
scanf("%d",&ele);
printf("Enter position of element to be inserted:");
scanf("%d",&pos);
if(pos>n)
{
printf("Insertion not possible");
}
else
{
for(i=n-1;i>=pos-1;i--)
{
a[i+1]=a[i];
}
a[pos-1]=ele;
printf("Array after insertion is:\n");
for(i=0;i<=n;i++)
{
printf(" %d ",a[i]);
}
}
}

void delete(int a[],int n)
{
int i,pos;
printf("Enter position of element to be deleted:");
```

```c
scanf("%d",&pos);
if(pos>n+1)
{
printf("Deletion not possible");
}
else
{
for(i=pos-1;i<n;i++)
{
a[i]=a[i+1];
}
printf("Array after deletion is:\n");
for(i=0;i<n;i++)
{
printf(" %d ",a[i]);
}
}
}

void search(int a[], int n)
{
int i,num,f=0;
printf("Enter the number to be found:");
scanf("%d",&num);
for(i=0;i<=n-1;i++)
{
if(a[i]==num)
{
printf("NUMBER FOUND!");
f++;
break;
}
}
if(f==0)
{
printf("NOT FOUND!");
}
}

void sort(int a[],int n){
    int i,j,t;
    for(i=0;i<n-1;i++){
        for(j=0;j<=n-1-i;j++){
            if(a[j]>a[j+1]){
                t=a[j+1];
                a[j+1]=a[j];
                a[j]=t;
            }
        }
    }
    for(i=0;i<n;i++){
        printf(" %d ",a[i]);
    }
}

int main()
{
int arry[10],ch,n;
do
{
printf("\nEnter your choice: \n1:Creation \n2:Insertion \n3:Deletion \n4:Searching(Linear) \n5:Print \n6:Sorting(increasing) \n7:Exit\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("Enter the no of element:");
scanf("%d",&n);
creation(arry,n);
print(arry,n);
break;
case 2:
insert(arry,n);
break;
case 3:
delete(arry,n);
break;
case 4:
search(arry,n);
break;
case 5:
print(arry,n);
break;
case 6:
sort(arry,n);
break;
case 7:
exit(0);
}
}while(ch<=6);
return 0;
}
```

Output:-



```
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit

1
Enter the no of element:4
1
8
5
3
 1  8  5  3
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
2
Enter element to be inserted:50
Enter position of element to be inserted:3
Array after insertion is:
 1  8  50  5  3
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
3
Enter position of element to be deleted:2
Array after deletion is:
 1  50  5  3
Enter your choice:
1:Creation
2:Insertion
3:Deletion
```


```
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
4
Enter the number to be found:50
NUMBER FOUND!
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
5
 1  50  5  3
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
6
 1  50  5  3
Enter your choice:
1:Creation
2:Insertion
3:Deletion
4:Searching(Linear)
5:Print
6:Sorting(increasing)
7:Exit
7
```

Q2-

```c
#include<stdio.h>
#include<stdlib.h>
void display(int a[],int n);
void bubble_sort(int a[],int n);
void selection_sort(int a[],int n);
void insertion_sort(int a[],int n);
//-----------------Main Function----------------------

int main()
{
   int n,choice,i;
   char ch[20];
   printf("Enter no. of elements u want to sort : ");
   scanf("%d",&n);
   int arr[5];
   for(i=0;i<n;i++)
   {
        printf("Enter %d Element : ",i+1);
        scanf("%d",&arr[i]);
   }
   printf("Please select any option Given Below for Sorting : \n");

while(1)
```

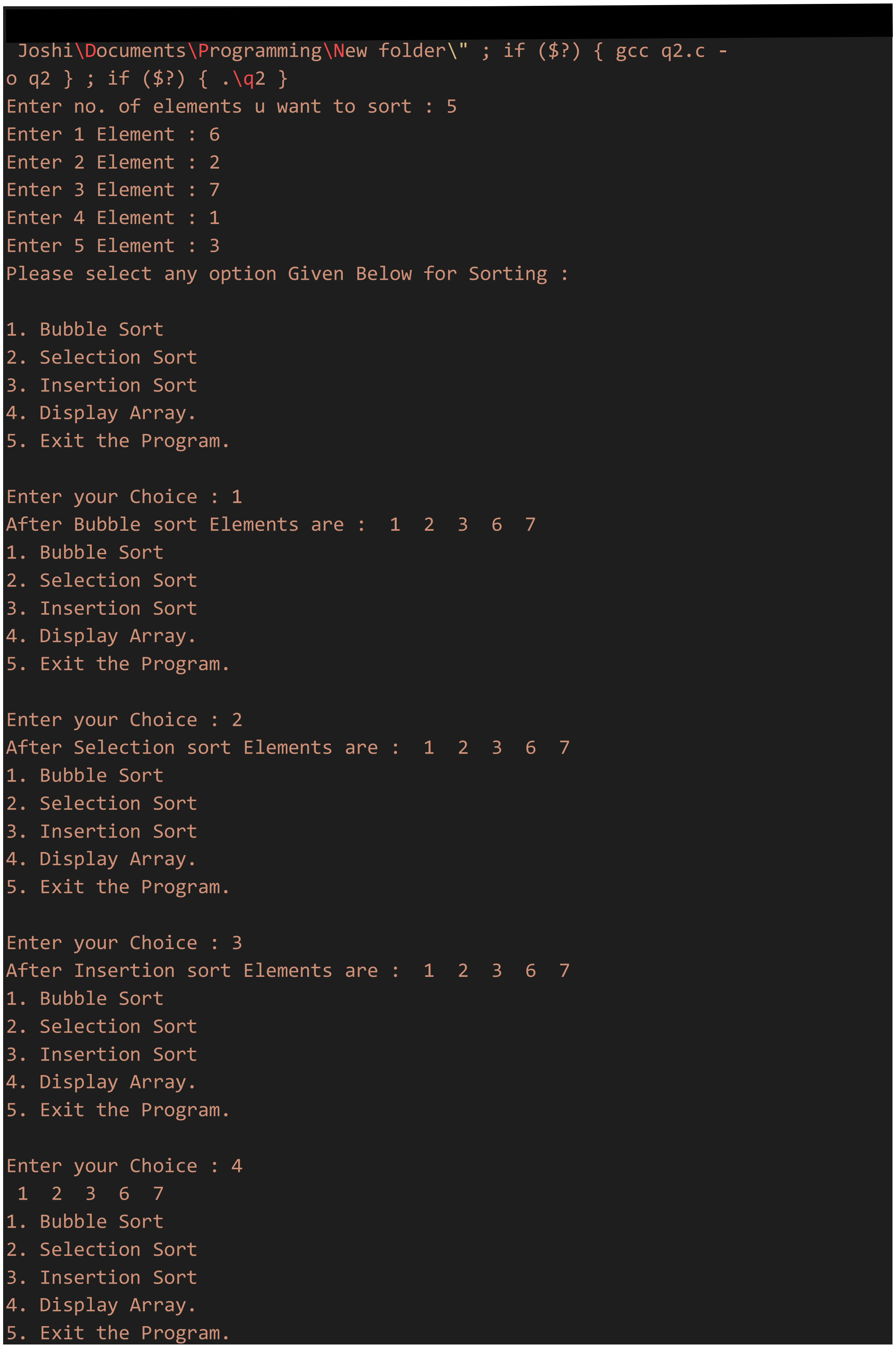
```c
  {


   printf("\n1. Bubble Sort\n2. Selection Sort\n3. Insertion Sort\n4. Display
Array.\n5. Exit the Program.\n");

   printf("\nEnter your Choice : ");

   scanf("%d",&choice);


   switch(choice)

   {

   case 1:

       bubble_sort(arr,n);

       break;

   case 2:

       selection_sort(arr,n);

       break;

   case 3:

       insertion_sort(arr,n);

       break;

   case 4:


       display(arr,n);

       break;


   case 5:
        return 0;

    default:

        printf("\nPlease Select only 1-5 option ----\n");

    }

}

return 0;

}



//-----------End of main function---------------------



//-------------------Display Function-----------------



void display(int arr[],int n)

{

    for(int i=0;i<n;i++)

   {

        printf(" %d ",arr[i]);

   }



}



//---------------------Bubble Sort Function-----------



void bubble_sort(int arr[],int n)
```

```
{
  int i,j,temp;
  for(i=0;i<n;i++)
  {
      for(j=0;j<n-i-1;j++)
      {
          if(arr[j]>arr[j+1])
          {
             temp=arr[j];
             arr[j]=arr[j+1];
             arr[j+1]=temp;
          }
      }
  }
printf("After Bubble sort Elements are : ");
display(arr,n);
}


//------------------Selection Sort Function---------


void selection_sort(int arr[],int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
```

```c
    {
        for(j=i+1;j<n;j++)
        {
            if(arr[i]>arr[j])
            {
             temp=arr[i];
             arr[i]=arr[j];
             arr[j]=temp;
            }
        }

    }
printf("After Selection sort Elements are : ");
display(arr,n);
}

//---------------Insertion Sort Function------------------

void insertion_sort(int arr[],int n)
{
    int i,j,min;
    for(i=1;i<n;i++)
    {
        min=arr[i];
        j=i-1;
        while(min<arr[j] && j>=0)
        {
            arr[j+1]=arr[j];
            j=j-1;
        }
        arr[j+1]=min;
    }
printf("After Insertion sort Elements are : ");
display(arr,n);
}
```

Output-

```
Enter no. of elements u want to sort : 5
Enter 1 Element : 6
Enter 2 Element : 2
Enter 4 Element : 1
Enter 5 Element : 2
Please select any option Given Below for Sorting :

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice :
Please Select only 1-5 option ----

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.
[redacted]
```



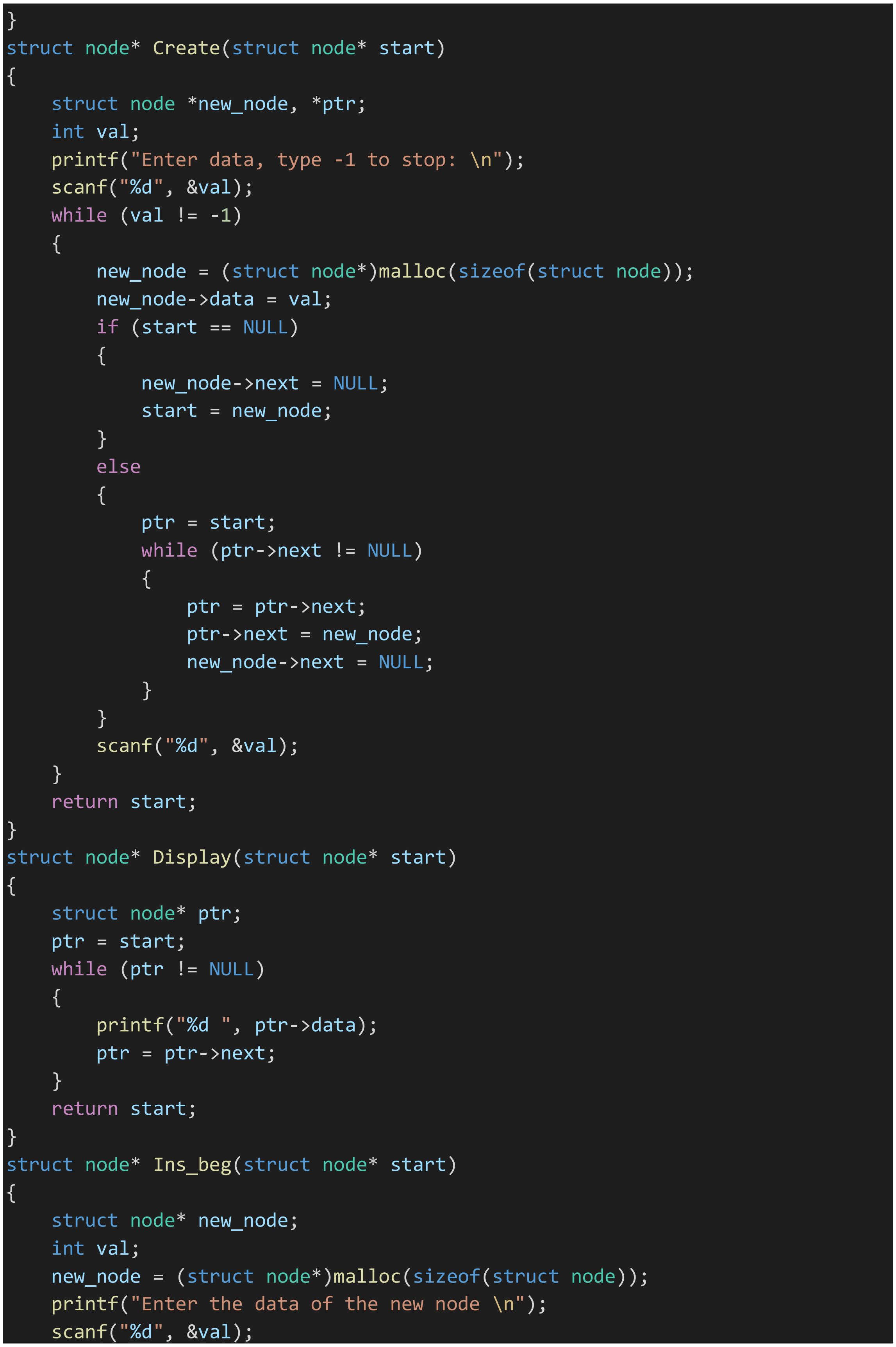
[redacted]

```
 Joshi\Documents\Programming\New folder\" ; if ($?) { gcc q2.c -
o q2 } ; if ($?) { .\q2 }
Enter no. of elements u want to sort : 5
Enter 1 Element : 6
Enter 2 Element : 2
Enter 3 Element : 7
Enter 4 Element : 1
Enter 5 Element : 3
Please select any option Given Below for Sorting :

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 1
After Bubble sort Elements are :  1   2   3   6   7
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 2
After Selection sort Elements are :  1   2   3   6   7
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 3
After Insertion sort Elements are :  1   2   3   6   7
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 4
 1   2   3   6   7
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.
```

```
Enter your Choice : 5
```

Q3.

```
#include<stdio.h>
#include<malloc.h>
struct node
{
    int data;
    struct node* next;
};
struct node* start = NULL;
struct node* Create(struct node*);
struct node* Display(struct node*);
struct node* Ins_beg(struct node*);
struct node* Del_end(struct node*);
void main()
{
    int ch;
    do {
        printf("\n Enter choice: \n 1. Create a linked list \n 2. Display list
 \n 3. Insert node at the beginning \n 4. Delete node at the end \n 5. Exit \n
");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:  start = Create(start);
                 printf("Linked list has been created \n");
                 break;

        case 2:  printf("Here is the list \n");
                 start = Display(start);
                 break;

        case 3:  start = Ins_beg(start);
                 printf("Node has been inserted at the beginning \n");
                 break;

        case 4:  start = Del_end(start);
                 printf("Node has been deleted from the end \n");
                 break;

        case 5:  break;

        default: printf("Invalid choice \n");
        }
    } while (ch != 5);
```

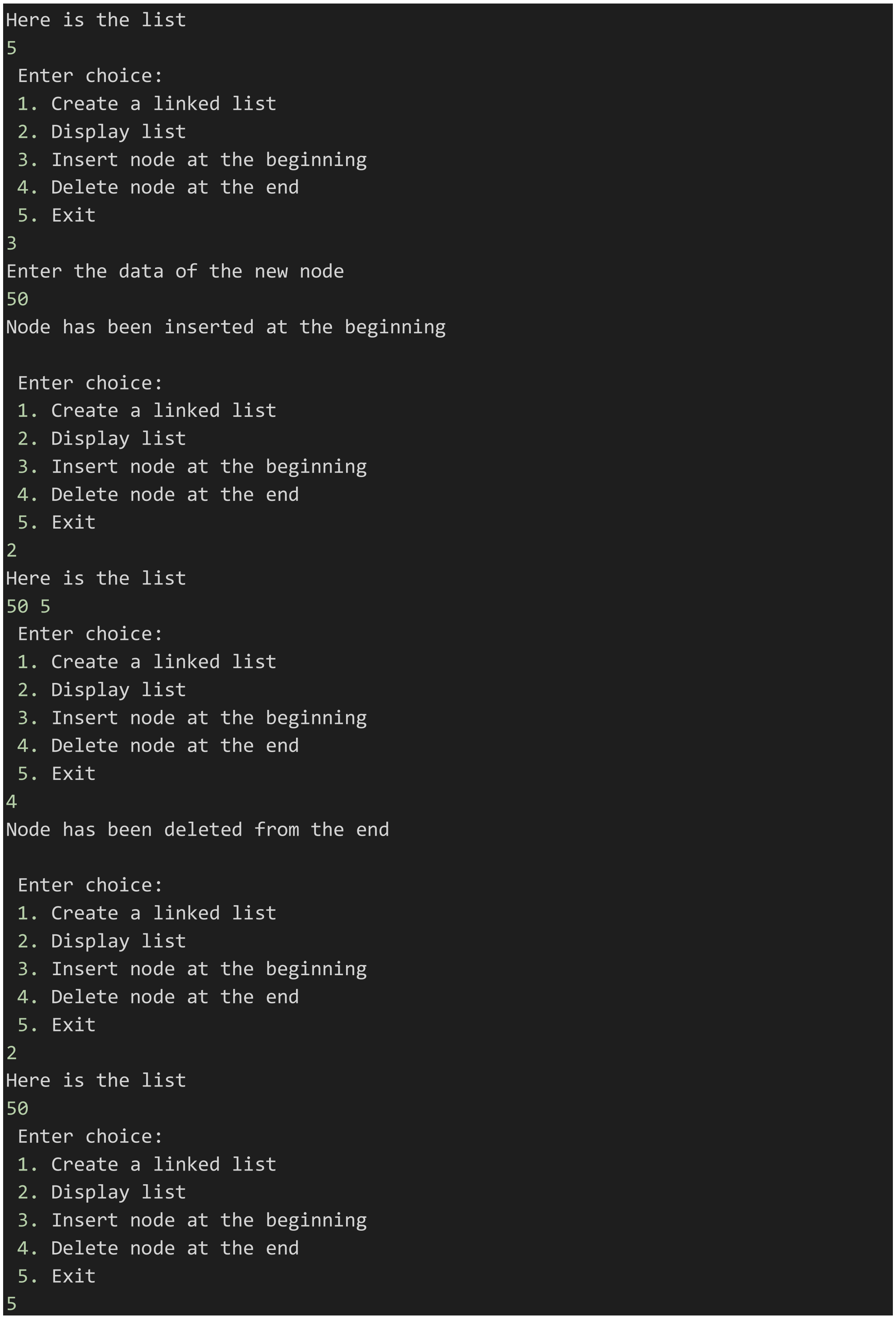


```
}
struct node* Create(struct node* start)
{
    struct node *new_node, *ptr;
    int val;
    printf("Enter data, type -1 to stop: \n");
    scanf("%d", &val);
    while (val != -1)
    {
        new_node = (struct node*)malloc(sizeof(struct node));
        new_node->data = val;
        if (start == NULL)
        {
            new_node->next = NULL;
            start = new_node;
        }
        else
        {
            ptr = start;
            while (ptr->next != NULL)
            {
                ptr = ptr->next;
                ptr->next = new_node;
                new_node->next = NULL;
            }
        }
        scanf("%d", &val);
    }
    return start;
}
struct node* Display(struct node* start)
{
    struct node* ptr;
    ptr = start;
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    return start;
}
struct node* Ins_beg(struct node* start)
{
    struct node* new_node;
    int val;
    new_node = (struct node*)malloc(sizeof(struct node));
    printf("Enter the data of the new node \n");
    scanf("%d", &val);
```

```c
    new_node->data = val;
    new_node->next = start;
    start = new_node;
    return start;
}
struct node* Del_end(struct node* start)
{
    struct node* ptr, * preptr;
    preptr = start;
    ptr = start;
    while (ptr->next != NULL)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = NULL;
    free(ptr);
    return start;
}
```

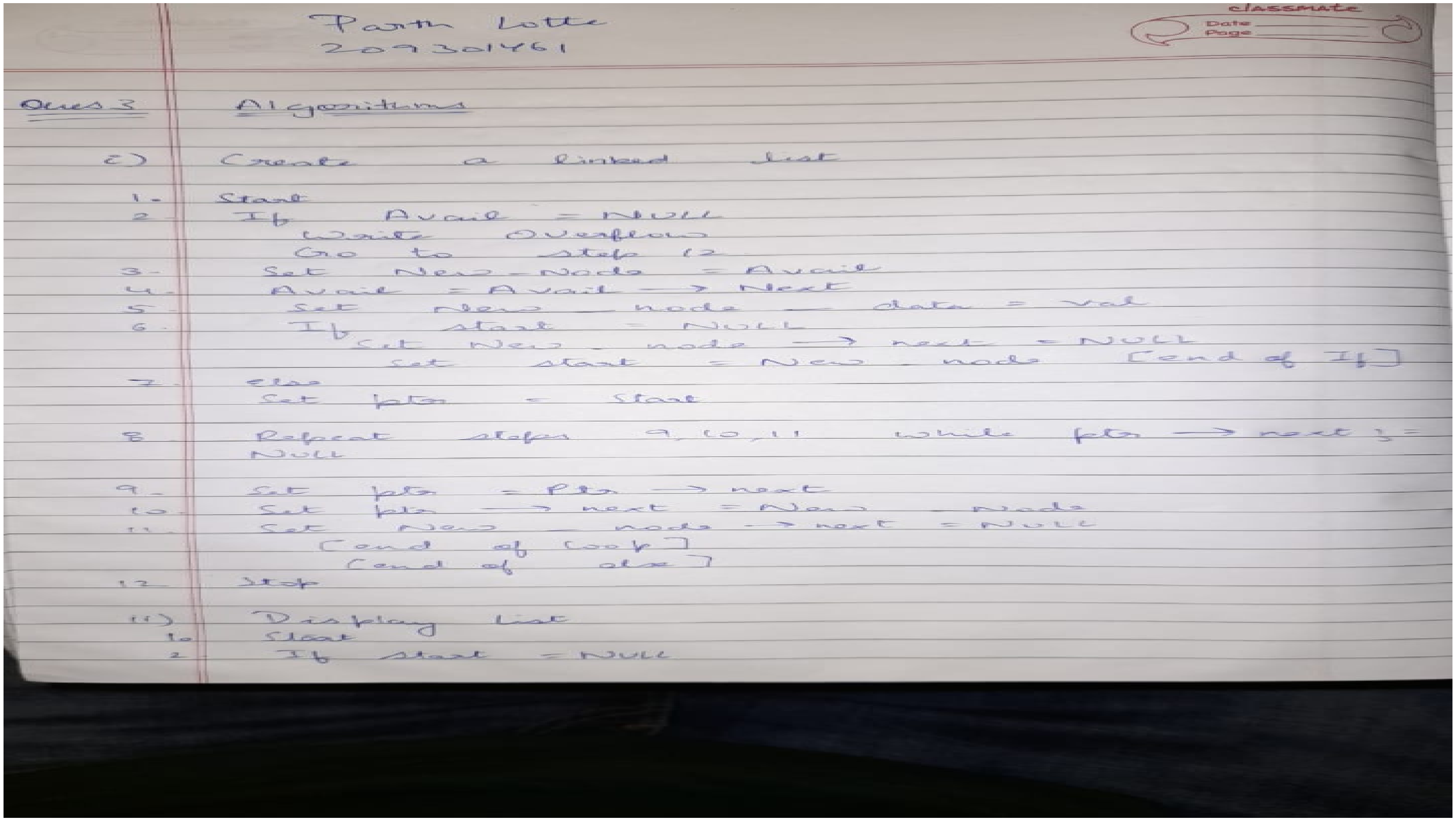
Output –

```
 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
1
Enter data, type -1 to stop:
5
6
Enter data, type -1 to stop:
7
5
Enter data, type -1 to stop:
2
-1
Linked list has been created

 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
2
```



```
Here is the list
5
 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
3
Enter the data of the new node
50
Node has been inserted at the beginning

 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
2
Here is the list
50 5
 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
4
Node has been deleted from the end

 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
2
Here is the list
50
 Enter choice:
 1. Create a linked list
 2. Display list
 3. Insert node at the beginning
 4. Delete node at the end
 5. Exit
5
```

Algorithm :-



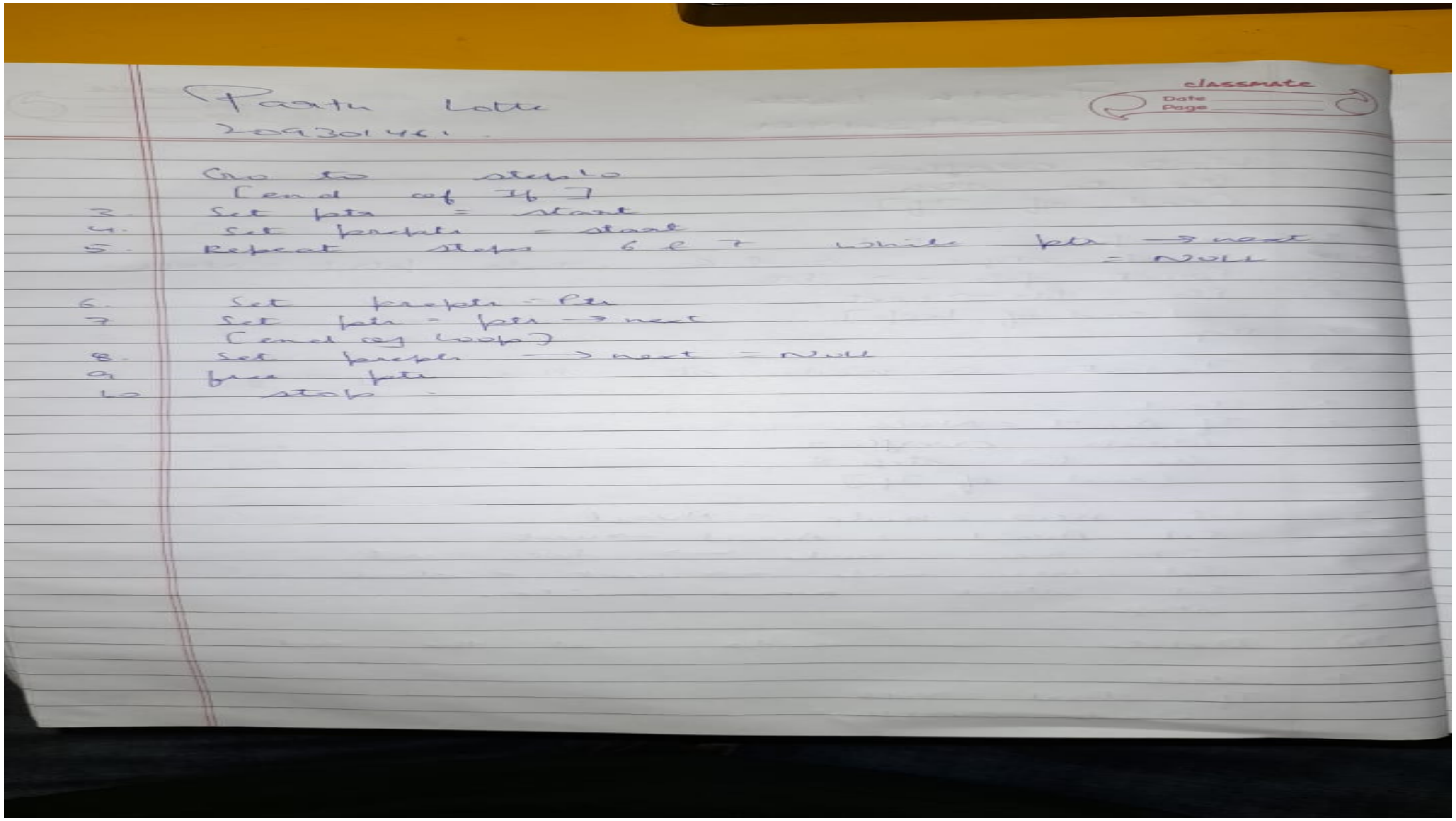
Parth Lotte
209301461

Ques 3 Algorithms

```
i) Create a linked list

1.  Start
2.  If Avail = NULL
        Write Overflow
        Go to step 12
3.  Set New_Node = Avail
4.  Avail = Avail → Next
5.  Set New_node — data = val
6.  If start = NULL
        Set New_node → next = NULL
        Set start = New_node [end of If]
7.  Else
    Set ptr = Start
8   Repeat steps 9, 10, 11 while ptr → next != Null
9.  Set ptr = Ptr → next
10. Set ptr → next = New_Node
11. Set New_node → next = NULL
        [end of loop]
        [end of else]
12. Stop

ii) Display list
1.  Start
2.  If start = NULL
```

Partha Lotte
2093014C1

```
     Write Overflow
     Go to step
     [end of If]
3.   Set ptr = start
4.   Repeat steps 5 & 6 while ptr != NULL
5.   Print ptr → data
6.   Ptr = ptr → next
       [end of Loop]
7.   stop.

(iii) Insert a node at the beginning.

1.   Start
2.   If Avail = NULL
       Write overflow
       Go to step 8
       [end of If]

3.   Set new_node = Avail
4.     Set Avail = Avail → next
5.     Set New_node → data = val
6.     Set new node → next = start
7.     Set start = New_node
8.     stop

iv)  Delete a node at the end

1.   Start
2.   If start = NULL
       Write overflow
```



Parth Lotte
20930146.

```
    Go to step 10
    [end of If]
3.  Set ptr = start
4.  Set preptr = start
5.  Repeat steps 6 & 7 while ptr → next
                                  = NULL

6.  Set preptr = ptr
7.  Set ptr = ptr → next
    [end of loop]
8.  Set preptr → next = NULL
9.  free ptr
10  stop.
```