Assignment -3

Que 1.

Input :- num1 = [4, 9, 5] , num 2 = [9, 4, 9, 8, 4]

Output :- [4, 9]

* Program :-

```
#include <bits/stdc++.h>
using namespace std;
void print Intersection (int num1[], int num2[], Int m, int n
{
  int i = 0, j = 0;
  while (i<m &&j <n){
      if (num1[i] < num2[j])
         i++;
      else if (num2[j] < num 1[i])
         j++;
      else
      {
         i++;
         j++;
      }
   }
}
 int main ()
 {
     int num1[] = {4, 9, 5};
     int num2[] = {9, 4, 9, 8, 4};
int m = sizeof(num1) / sizeof(num1[0]);
int n = sizeof(num2) / sizeof(num2[0]);

// function Calling

printIntersection(num1, num2, m, n);

return 0;
}
```

OUTPUT

```
4,9
```

Que 2.

Input :- Head of following linked list
1 -> 2 -> 3 -> 4 -> NULL

Output :- Linked list should be changed to:
4 -> 3 -> 2 -> 1 -> NULL

* Program

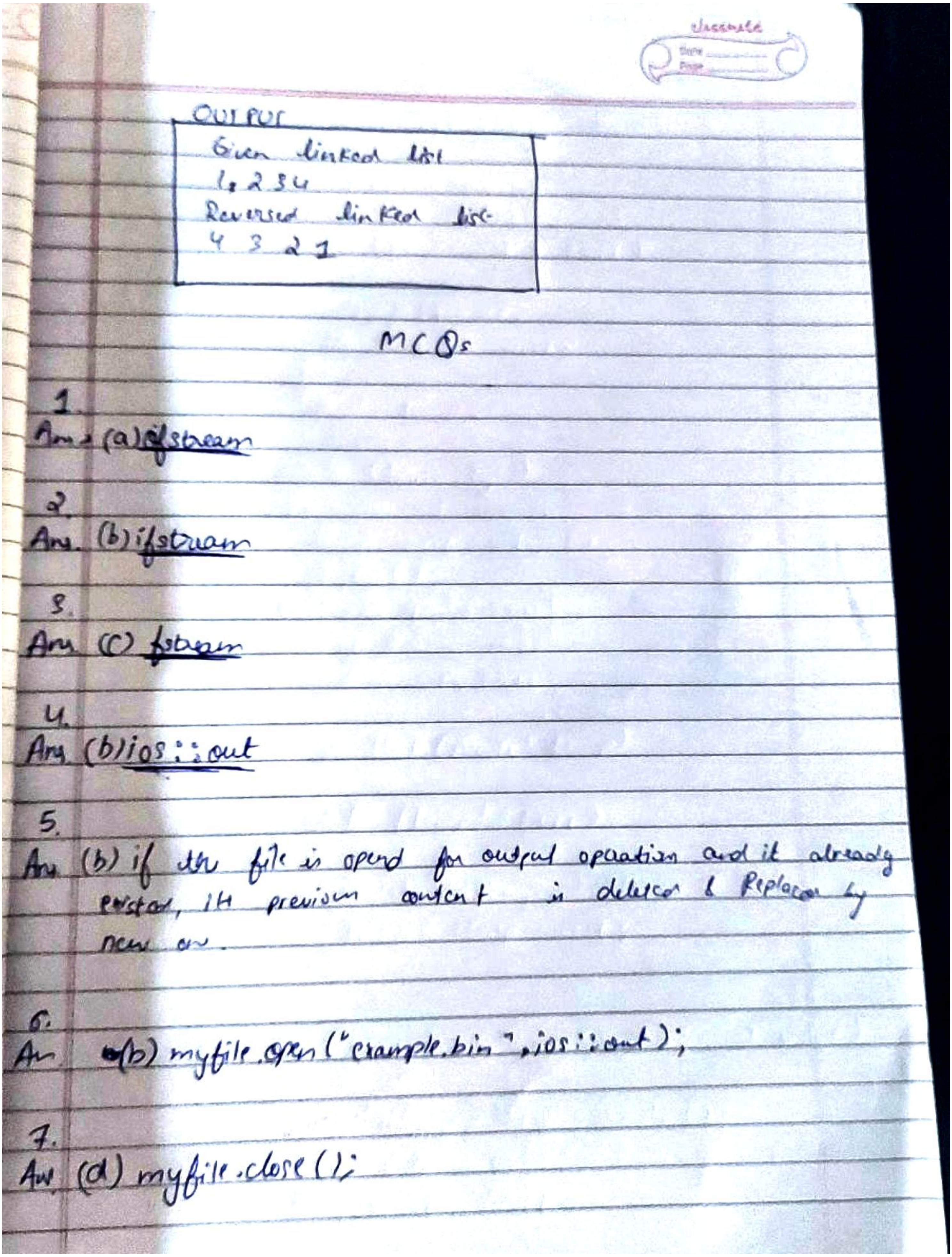
```
# include <bits/stdc++.h>
#include <iostream>

using namespace std;
    Class Node
    {
    public ;
     Node *next ;
    };
```

```
void reversLL (Node ** head)
{
   stack <Node *> S ;
   Node * temp = *head;
   while (temp->next != NULL)
   {
      S.push(temp);
      temp = temp->next;
   }
   *head = temp;

   while (!s.empty())
   {
      temp->next = s.top();
      S.pop();
      temp = temp->next;
   }
   temp->next = NULL;
}
void printlist (Node* temp)
{
   while (temp != NULL)
   {
      count << temp->data << " ";
      temp = temp->next;
   }
}
void insert_back (Node ** head, int value)
{
   Node * temp = new Node();
   temp->data = value;
   temp->next = NULL;
```

```
    if (*head == NULL)
      {
        *head = temp;
        return;
      }
    else
      {
      Node* last_node = *head;
      while (last_node->next != NULL)
        {
          last_node = last_node->next;
        }
      last_node->next = temp;
      return;
      }
  }
int main ()
{
  Node* head = NULL;
    insert_back (&head, 1);
    insert_back (&head, 2);
    insert_back (&head, 3);
    insert_back (&head, 4);
    cout << "Given linked list \n";
    printlist (head);
    reverseLL(&head);
    cout << "\n Reversed linked list \n";
    printlist (head);
    return 0;
}
```



OUTPUT

```
Given linked list
1, 2 3 4
Reversed linked list
4 3 2 1
```

MCQs

1.
Ans (a) ofstream

2.
Ans. (b) ifstream

3.
Ans (c) fstream

4.
Ans (b) ios::out

5.
Ans (b) if the file is opend for output opeation and it already existed, its previous content is deleted & Replaced by new one.

6.
An (b) myfile.open ("example.bin", ios::out);

7.
Ans (d) myfile.close();