Data Structures and Algorithms

TEAM ASSIGNMENT

Adhin Jibil (RA2011030010031)

Aditya A R (RA2011030010052)

Gokul M K (RA2011030010023)

QUESTION

Q.1) Create a student database with 5 students using doubly linked list in which each student data like roll number, name, marks of 5 subjects (marks between 0 and 100) are stored in the nodes by creating structures.

i) Print the students' details by traversing in both the direction.

ii) Calculate the average marks of the student and display his grade as per the grade mapping below along with name and roll number.

Mark Range	Grade
90 - 100	O
80 - 89	A
70 - 79	B
60 - 69	C
50 - 59	D
0 - 49	F

1.AIM:

To create a student database with 5 students using a doubly linked list in which each student data's are stored, and details are displayed by traversing in both the direction (forward and backward).

2.ALGORITHM

PUSHING DATA -push()

Step 1: IF ptr = NULL

Write OVERFLOW

Go to Step 9

[END OF IF]

Step 2: SET NEW_NODE = ptr

Step 3: SET ptr = ptr -> NEXT

Step 4: SET NEW_NODE -> DATA = VAL

Step 5: SET NEW_NODE -> PREV = NULL

Step 6: SET NEW_NODE -> NEXT = START

Step 7: SET head -> PREV = NEW_NODE

Step 8: SET head = NEW_NODE

Step 9: EXIT

GRADING MARKS -grading()

Step 1: Start

Step 2: Declare variable sum and avg.

Step 3: Add **temp->math + temp->dsa + temp->ade + temp->coa + temp-oodp** and assign the result to sum.

Step 4: Assign avg=sum/5

Step 5: if avg >= 90
assign temp->grade=O;

else if avg >=80 and avg<90
assign temp->grade=A;

else if avg >=70 and avg<80
assign temp->grade=B;

else if avg >=60 and avg<70
assign temp->grade=C;

else if avg >=50 and avg<60
assign temp->grade=D;

else
assign temp->grade=F;

Step 6: Stop

DISPLAYING DATA -print()

Step 1 : Check whether list is Empty (head == NULL)

Step 2 : If it is Empty, then display 'List is Empty!!!' and terminate the function.

Step 3 : If it is not Empty, then define a Node pointer 'temp' and initialize with head.

Step 4 : Display 'NULL <--- '.

Step 5 : Keep displaying temp → data with an arrow (<===>) until temp reaches to the last node

Step 6 : Finally, display temp → data with arrow pointing to NULL (temp → data ---> NULL).

FORWARD TRAVERSAL

Step 1:Start

Step 2: If (START is equal to NULL)

Step 3: Display “The list is Empty”

Step 4: Stop

Step 5: Initialize TEMP = START

Step 6: Repeat the step 5 and 6 until (TEMP == NULL)

Step 7: Display “TEMP → DATA”

Step 8:TEMP = TEMP → Next

Step 9: Stop

BACKWARD TRAVERSAL

Step 1: Start

Step 2: If (START is equal to NULL)

Step 3: Display “The list is Empty”

Step 4: Stop

Step 5: Initialize TEMP = TAIL

Step 6: Repeat the step 5 and 6 until (TEMP == NULL)

Step 7: Display “TEMP ⟶ DATA”

Step 8:TEMP = TEMP ⟶ Prev

Step 9: Stop

3.PROGRAM

```cpp
/*
        QUESTION NUMBER 1
                                                PROGRAM STARTS
*/

#include <iostream>                                                    //header files

using namespace std;
struct Node
{
    public:
    string name,reg;
    int math,dsa,ade,coa,oodp,avg;
    char grade;
    Node* next;
    Node* prev;
} *head=NULL;

void grading(Node* temp)                          //function for allotting grade
{
    int sum=0;
    sum=temp->math + temp->dsa + temp->ade + temp->coa + temp->oodp;
    temp->avg=sum/5;
    if(temp->avg>=90)
        temp->grade='O';
    else if(temp->avg>=80 && temp->avg<90)
        temp->grade='A';
    else if(temp->avg>=70 && temp->avg<80)
        temp->grade='B';
    else if(temp->avg>=60 && temp->avg<70)
        temp->grade='C';
    else if(temp->avg>=50 && temp->avg<60)
        temp->grade='D';
    else
        temp->grade='F';
}

void push()                                                    //function for pushing data
{
    Node* new_node = new Node();
    cout<<"\n Enter the following details of student";
    cout<<"\n\tName   : ";
    cin>>new_node->name;
    cout<<"\n\tRegistration number  : ";
    cin>>new_node->reg;
    cout<<"\n\tMarks obtained in Math [100]  : ";
    cin>>new_node->math;
    cout<<"\n\tMarks obtained in DSA [100]  : ";
    cin>>new_node->dsa;
    cout<<"\n\tMarks obtained in ADE [100]  : ";
    cin>>new_node->ade;
    cout<<"\n\tMarks obtained in COA [100]  : ";
```

```
    cin>>new_node->coa;
    cout<<"\n\tMarks obtained in OODP [100]  : ";
    cin>>new_node->oodp;
    new_node->next=NULL;
    if(head==NULL)
    {
        head=new_node;
        new_node->prev=NULL;
    }
    else
    {
        Node* temp = head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=new_node;
        new_node->prev=temp;
    }
    grading(new_node);
}

void print(Node *temp)                                          //function for
printing data
{
    cout<<"\n Name of the student: "<<temp->name;
    cout<<"\n Registration number: "<<temp->reg;
    cout<<"\n Marks obtained ";
    cout<<"\n\tMath = "<<temp->math;
    cout<<"\n\tDSA = "<<temp->dsa;
    cout<<"\n\tADE = "<<temp->ade;
    cout<<"\n\tCOA = "<<temp->coa;
    cout<<"\n\tOODP = "<<temp->oodp;
    cout<<"\n\tAverage = "<<temp->avg;
    cout<<"\n\tGrade = "<<temp->grade;
}


void printList()                                                //function for
printing list in each direction
{
    Node* temp = head;
    cout<<"\n Forward travesing\n";
    cout<<"   ............  ";
    while(temp->next!=NULL)     //  Start with the front node and visit all the nodes until
                                                                //the node becomes NULL.

    {
        print(temp);
        temp=temp->next;
    }
    print(temp);
    cout<<"\n\n Backward travesing\n";
```

```
    cout<<"   ..............    ";
    while(temp->prev!=NULL)              //Start with the end node and visit all the nodes
                                                    //until the node becomes NULL.
    {
        print(temp);
        temp=temp->prev;
    }
    print(temp);
}

int main()                                          //main function
{
    int n;
    cout<<"\n Enter the number of entries to be made: ";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        push();
        cout<<"\n";
    }
    printList();
    return 0;
}
                                        //PROGRAM ENDS
/*
            Done by team MASTERS
    Contributions:-
    Adhin Jibil (RA2011030010031): struct Node,push()
    Aditya A R (RA2011030010052): grading(),main()
    Gokul M K (RA2011030010023): print(),printList()
*/
```

4.OUTPUT

input

```
Enter the number of entries to be made: 5

Enter the following details of student 1
       Name    : Aditya

       Registration number   : 2011030010052

       Marks obtained in Math [100]   : 70

       Marks obtained in DSA [100]   : 40

       Marks obtained in ADE [100]   : 69

       Marks obtained in COA [100]   : 93

       Marks obtained in OODP [100]   : 84


Enter the following details of student 2
       Name    : Gokul

       Registration number   : 2011030010023

       Marks obtained in Math [100]   : 40

       Marks obtained in DSA [100]   : 60

       Marks obtained in ADE [100]   : 53

       Marks obtained in COA [100]   : 82

       Marks obtained in OODP [100]   : 90

Enter the following details of student 3
       Name    : Jibil

       Registration number   : 2011030010031

       Marks obtained in Math [100]   : 60

       Marks obtained in DSA [100]   : 43

       Marks obtained in ADE [100]   : 72

       Marks obtained in COA [100]   : 81

       Marks obtained in OODP [100]   : 40


Enter the following details of student 4
       Name    : Anand

       Registration number   : 2011030010111

       Marks obtained in Math [100]   : 20

       Marks obtained in DSA [100]   : 35

       Marks obtained in ADE [100]   : 25

       Marks obtained in COA [100]   : 30

       Marks obtained in OODP [100]   : 39


Enter the following details of student 5
       Name    : Aysha
```

```
       Registration number  : 2011030010112

       Marks obtained in Math [100]  : 90

       Marks obtained in DSA [100]  : 84

       Marks obtained in ADE [100]  : 93

       Marks obtained in COA [100]  : 92

       Marks obtained in OODP [100]  : 98


Forward travesing
  ............
Name of the student: Aditya
Registration number: 2011030010052
Marks obtained
       Math = 70
       DSA = 40
       ADE = 69
       COA = 93
       OODP = 84
       Average = 71
       Grade = B
Name of the student: Gokul
Registration number: 2011030010023
Marks obtained
       Math = 40
       DSA = 60
       ADE = 53
       COA = 82
       OODP = 90
       Average = 65
       Grade = C
Name of the student: Jibil
Registration number: 2011030010031
Marks obtained
       Math = 60
       DSA = 43
       ADE = 72
       COA = 81
       OODP = 40
       Average = 59
       Grade = D
Name of the student: Anand
Registration number: 2011030010111
Marks obtained
       Math = 20
       DSA = 35
       ADE = 25
       COA = 30
       OODP = 39
       Average = 29
       Grade = F
Name of the student: Aysha
Registration number: 2011030010112
Marks obtained
       Math = 90
       DSA = 84
       ADE = 93
       COA = 92
       OODP = 98
       Average = 91
       Grade = O

Backward travesing
  ............
Name of the student: Aysha
Registration number: 2011030010112
```

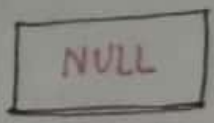
```
Marks obtained
       Math = 90
       DSA = 84
       ADE = 93
       COA = 92
       OODP = 98
       Average = 91
       Grade = O

Backward travesing
  ..............
Name of the student: Aysha
Registration number: 2011030010112
Marks obtained
       Math = 90
       DSA = 84
       ADE = 93
       COA = 92
       OODP = 98
       Average = 91
       Grade = O
Name of the student: Anand
Registration number: 2011030010111
Marks obtained
       Math = 20
       DSA = 35
       ADE = 25
       COA = 30
       OODP = 39
       Average = 29
       Grade = F
Name of the student: Jibil
Registration number: 2011030010031
Marks obtained
       Math = 60
       DSA = 43
       ADE = 72
       COA = 81
       OODP = 40
       Average = 59
       Grade = D
Name of the student: Gokul
Registration number: 2011030010023
Marks obtained
       Math = 40
       DSA = 60
       ADE = 53
       COA = 82
       OODP = 90
       Average = 65
       Grade = C
Name of the student: Aditya
Registration number: 2011030010052
Marks obtained
       Math = 70
       DSA = 40
       ADE = 69
       COA = 93
       OODP = 84
       Average = 71
       Grade = B

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

5.DRY RUN

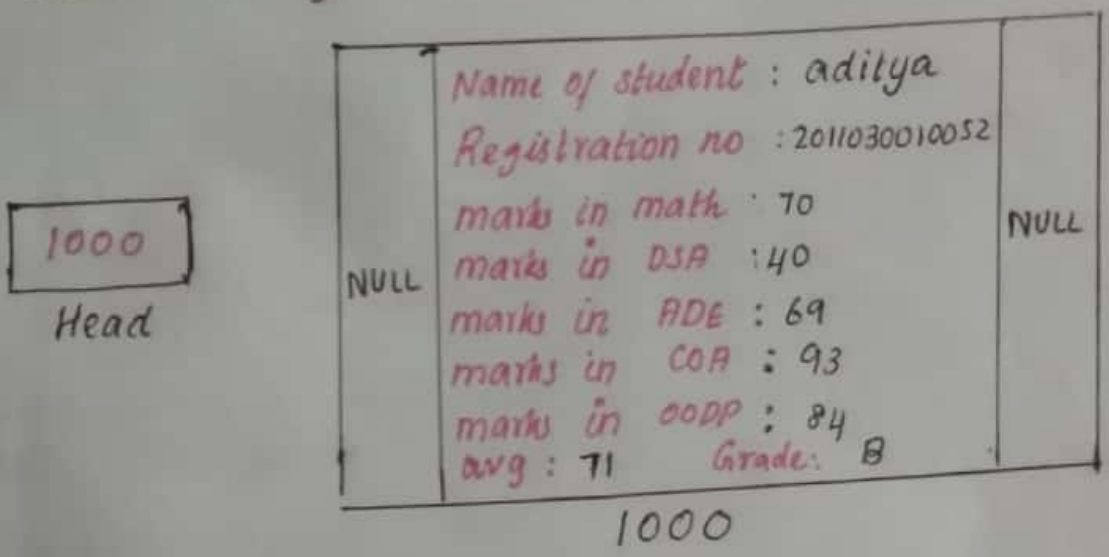
Student database with 5 students using doubly linked list.

Number of entries = 5

Initially head is declared as Null



// then creating a node called (new_node)



Here, the node for aditya is created.

[The address of the 1st node be 1000, if the head is null, the address of the first node stored in the head]

then grading function is called

temp→ math = 70, temp → DSA = 40, temp→ ADE = 69, temp → COA = 93, temp→ OOBP = 84.

The temp→ avg is 71, so temp→ grade = 'B'

// creating 4th node

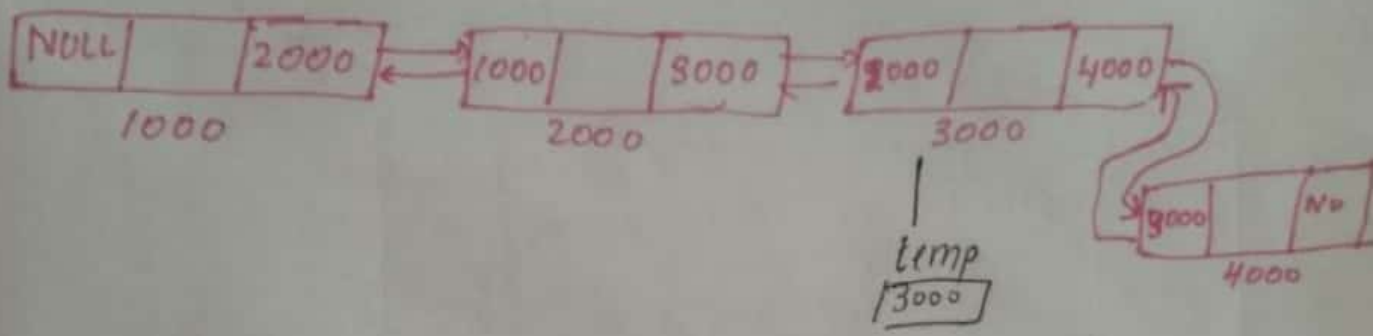
Now, temp should be pointed 3000

new_node = 4000

temp->next = 4000

new_node-> prev = 3000

new_node -> NULL



// creating 5th node..

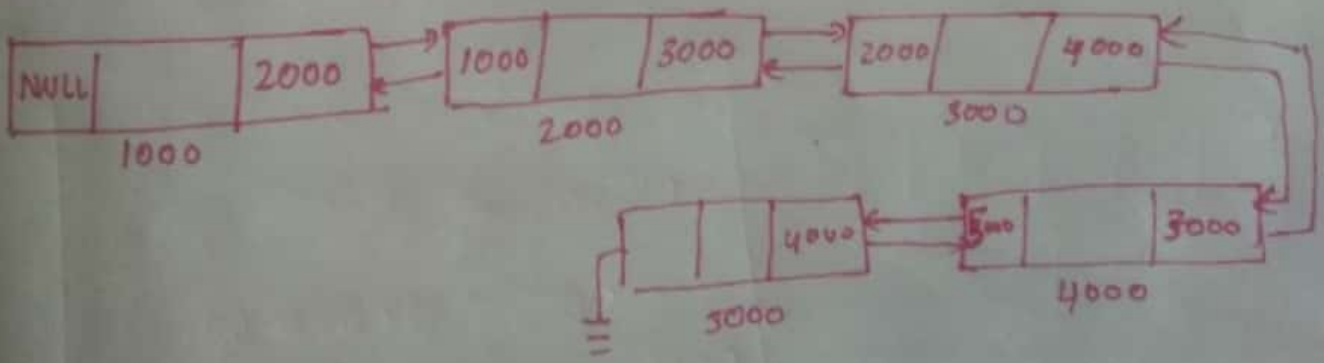
Now, temp should be pointed 4000.

new_node = 5000

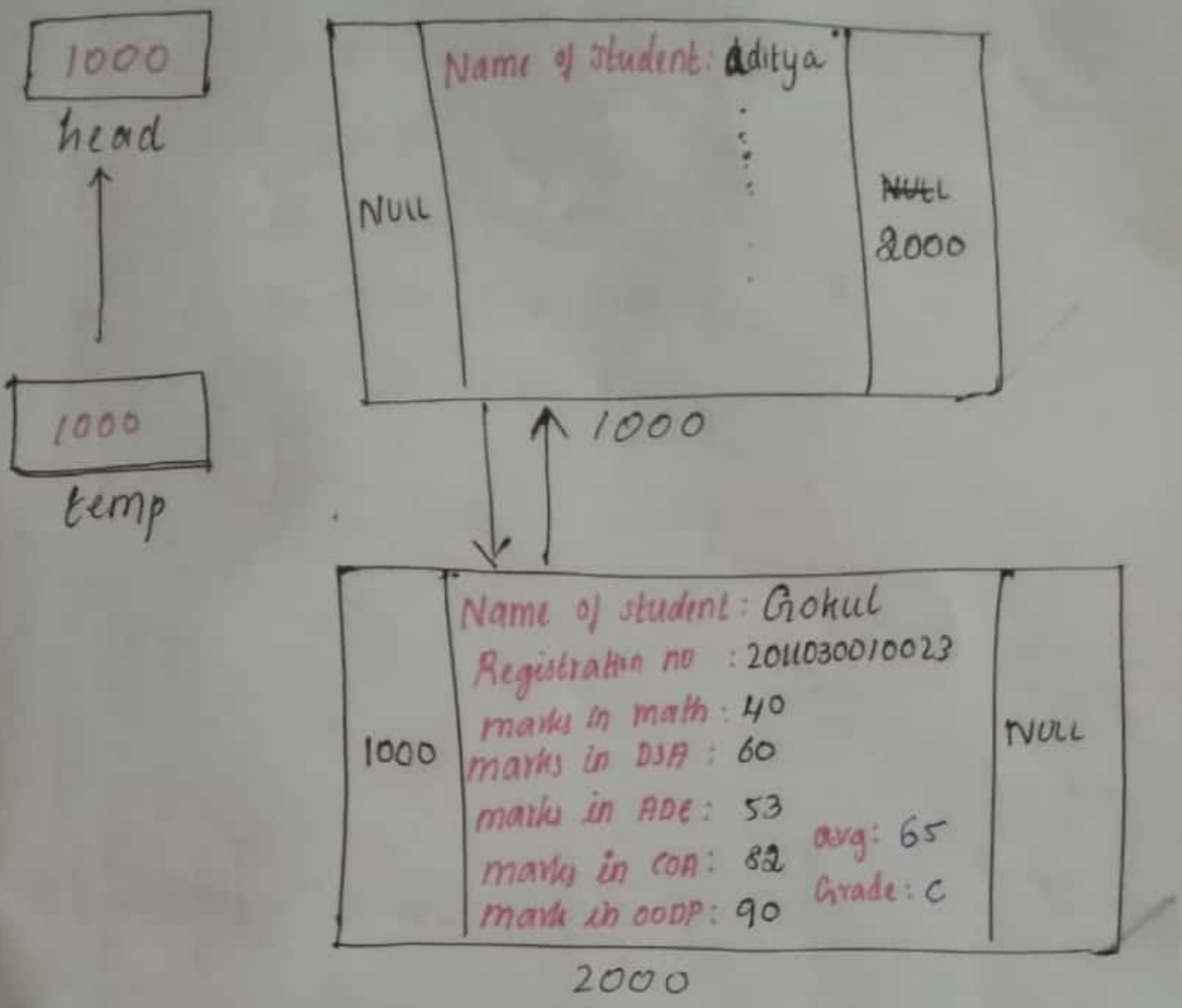
temp_next = 5000

new_node ->prev = 4000

new_node -> NULL



// Creating next node

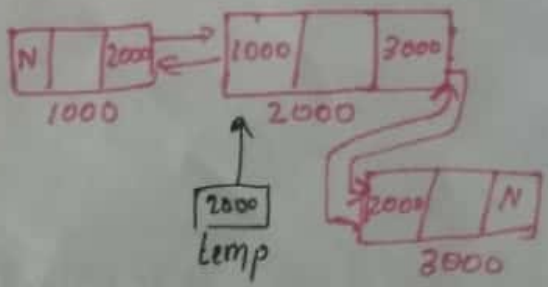


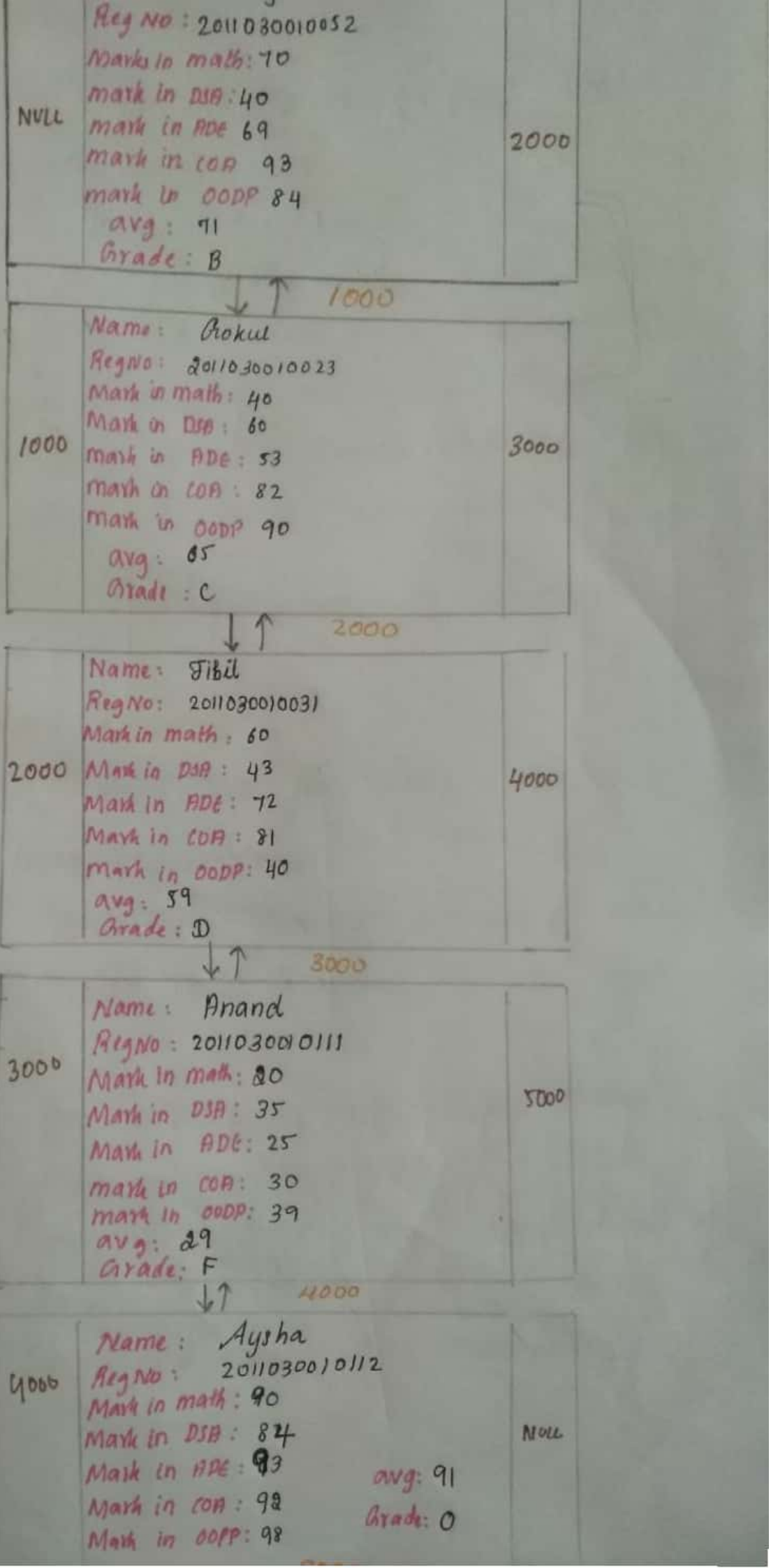
Now, New_node = 2000.
temp → Next = 2000
new_node → prev = 1000.
new_node → next = NULL

temp → avg is 65
temp → grade = 'C'

// creating next node

New_node = 3000
temp should be 2000
temp → next = 3000
new_node → prev = 2000
new_node → next = NULL.




Reg No : 20110300100 52
Marks in math: 70
mark in DSA : 40
mark in ADE 69
mark in COA 93
mark in OODP 84
avg : 71
Grade : B
NULL
2000
1000
Name : Gokul
RegNo : 20110300100 23
Mark in math : 40
Mark in DSA : 60
mark in ADE : 53
mark in COA : 82
mark in OODP 90
avg : 65
Grade : C
1000
3000
2000
Name : Tibil
RegNo : 20110300100 31
Mark in math : 60
Mark in DSA : 43
Mark in ADE : 72
Mark in COA : 81
mark in OODP : 40
avg : 59
Grade : D
2000
4000
3000
Name : Anand
RegNo : 20110300100 111
Mark in math : 20
Mark in DSA : 35
Mark in ADE : 25
mark in COA : 30
mark in OODP : 39
avg : 29
Grade : F
3000
5000
4000
Name : Aysha
RegNo : 20110300100 112
Mark in math : 90
Mark in DSA : 84
Mark in ADE : 93
Mark in COA : 92
Mark in OOPP : 98
avg : 91
Grade : O
4000
NULL

// Forward Travesing:

Print all data in each node

till temp -> next = NULL.

Initialy temp = 1000

1. It prints the aditya details, then the temp points 2000, then it prints Gokul details, temp points 3000, it prints Jibil details, temp points 4000, it prints anand details, temp points 5000, it prints aysha details.

then,

temp -> next = NULL, the condition fails.

// Backward Travesing.

Print all data in each node

till temp -> prev = NULL.

After the forward travesing, the current temp value is 5000, then it prints the details of aysha,

temp value 4000, => prints anand details

temp value 3000 => prints Jibil details

temp value 2000 => prints Gokul details

temp value 1000 => prints aditya details

then, temp -> prev = NULL, the condition fails..

6.RESULT

A program to create and traverse through(both forward and backward) a database of students' marks with the function to grade them has been created.