CSE2003 DATA STRUCTURES AND ALGORITHMS
DIGITAL ASSIGNMENT 1

INDUSTRY BASED APPLICATION :
HOSTEL COUNSELLING APPLICATION (EDUCATIONAL SECTOR)

CODE :

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#define MAXCHAR 1000
int absent[100],present[100],Qblock_1bed[10],Qblock_2bed[10][10],num=12;
float grp_topper=0;
//hash table is used to map the student details(key) to its
  slot/class(value).
//collision will occur since there are two slots, which is resolved through
  chaining.
struct hash *cse_branch=NULL;
struct hash *ece_branch=NULL;
struct hash *mech_branch=NULL;

struct node              //structure for student details
{
    int rollno;
    float cgpa;
    float ncgpa;
    int rank;
    char name[100];
    struct node *next;
}*current,*temp,*last=NULL,*p;

struct hash
{
    int count;
    float highest;
    struct node *head;
};

struct node *createnode(int rollno, char *name, float cgpa)
{
    struct node *newnode;
    newnode=malloc(sizeof(struct node));
    newnode->rollno=rollno;
    newnode->cgpa=cgpa;
    newnode->ncgpa=0;
    newnode->rank=0;
    strcpy(newnode->name,name);
    newnode->next=NULL;
    return newnode;
};
```

```
void insert(struct hash *branch, int rollno, char *name, float cgpa, int x)
{
    int index=x;
    struct node *newnode = createnode(rollno, name, cgpa);
    if((int)branch[index].head==0)
    {
        branch[index].head=newnode;
        branch[index].count=1;
        current=newnode;
    }
    else
    {
        current=branch[index].head;
        if(newnode->cgpa > current->cgpa)
        {
            newnode->next=current;
            branch[index].head=newnode;
            return;
        }
        else
        {
            while(current!=NULL)
            {
                if(newnode->cgpa <= current->cgpa)
                {
                    p=current;
                    current=current->next;
                }
                else
                    break;
            }
            newnode->next=p->next;
            p->next=newnode;
            branch[index].count++;
        }
    }
}

int search_details(struct hash *branch, int rno)
{
    int i,j;
    struct node *search;
    for(i=1;i<=2;i++)
    {
        search=branch[i].head;
        if(!search)
            return 0;
        while(search!=NULL)
        {
            if(search->rollno==rno)
            {
                printf("Roll No : %d\nName : %s\nCGPA : %.2f\n",search-
                                >rollno,search->name,search->cgpa);
                return 1;
            }
            search=search->next;
        }
    }
    return 0;
}
```

```
//calculate NCGPA of the students.
void NCGPA(struct hash *branch)
{
    struct node *class_topper;
    struct node *student;
    for(int i=0;i<=2;i++)
    {
        class_topper=branch[i].head;
        student=branch[i].head;
        while(student!=NULL)
        {
            student->ncgpa=(student->cgpa*grp_topper)/branch->highest;
            student=student->next;
        }
    }
}

void display(struct hash *branch)
{
    struct node *m;
    int i;
    for(i=1;i<=2;i++)
    {
        if(branch[i].count==0)
            continue;
        m = branch[i].head;
        printf("\n Students in Slot %d of branch:\n",i);
        printf("ROLL NO  \t  NAME             CGPA\n");
        while(m!=NULL)
        {
            printf("  %d\t\t%-15s\t%10f\n",m->rollno,m->name,m->cgpa);
            m=m->next;
        }
    }
}

void topper(struct hash *branch, char p[])
{
    struct node *z;
    branch->highest=branch[1].head->cgpa;
    z=branch[2].head;
    if(branch->highest < z->cgpa)
    {
        branch->highest=z->cgpa;
        z=branch[1].head;
    }
    printf("%s branch topper is :\nROLL NO: %d\nCGPA: %f\n",p,z->
                                        rollno,branch->highest);
    if(grp_topper<branch->highest)
    {
        grp_topper=branch->highest;
    }
}

void displaylist(struct node *disp)
{
    printf("    ROLL NO       NAME                  CGPA          NCGPA\n");
    printf("-------------------------------------------------------------------
\n");
    while(disp!=NULL)
```

```
    {
        printf("%8d\t%-15s%15f%15f\n",disp->rollno,disp->name,disp->
                                                        cgpa,disp->ncgpa);
        disp=disp->next;
    }
}
//using merge sort to sort the NCGPA list
struct node* merge(struct node* temp1, struct node* temp2)
{
    struct node* mergelist= NULL;
      mergelist=(struct node*)malloc(sizeof(struct node));

    if (temp1 == NULL)
        return(temp2);
    else if (temp2 == NULL)
        return(temp1);
    if ( temp1->ncgpa >= temp2 ->ncgpa)
    {
        mergelist=temp1;
      mergelist->next=merge(temp1->next, temp2);
    }
    else
    {
        mergelist=temp2;
        mergelist->next=merge(temp1,temp2->next);
    }
    return(mergelist);
}

struct node* ncgpalist(struct hash *hashtable)
{
    NCGPA(hashtable);
      return merge(hashtable[1].head, hashtable[2].head);
}

void mergeall (struct node* branch1, struct node* branch2, struct node*
branch3)
{
    printf("\n\n   FINAL NCGPA LIST: \n\n");
    last=merge(merge(branch1, branch2), branch3);
      displaylist(last);
}

int printlist()
{
      struct node *ptr=last;
      int x, rank=1;
      if (ptr==NULL)
      {
            printf("\n\n  SERVER ERROR !!");
            exit(0);
      }
      printf("\n\n\tRANK    ROLL NO     NCGPA\n");
      printf("-------------------------------------\n");
      while (ptr!=NULL)
      {
            ptr->rank=rank;
            printf("\n\t%d",ptr->rank);
            printf("\t  %d",ptr->rollno);
            printf("\t  %f",ptr->ncgpa);
            ptr=ptr->next;
```

```
            rank++;
      }
    return rank;
}
//students are given tokens according to ranks obtained
//queue data structure is used, based on FIFO,
void tokenlist()
{
      int i=1;
      struct node *current= NULL;
      printf("\n\nThe student rankers taking part in counselling are: \n");
    current=last;
      while (current!=NULL)
      {
            present[i++]=current->rank;
        current=current->next;
    }
      i=1;
      while(present[i]!=0)
      {
            printf(" %d\n",present[i++]);
      }
}

int findstudent(int arank)
{
      struct node *prev=NULL;
    struct node *current=NULL;
    struct node *temp=NULL;
    current=last;
      while (current!=NULL)
      {
            if (current->rank==arank)
            {
                  prev->next=current->next;
             temp=current;
                  current=current->next;
             free(temp);
                  return 1;
            }
            else
            {
                  prev=current;
                  current=current->next;
            }
      }
      return 0;
}

void registration(int totalstud)
{
      struct node* prev=NULL;
      struct node* current=NULL;
      current=last;
      int i=1, absentrank=0;
      printf("\nTotal students is: %d", totalstud-1);
      printf("\nEnter the rankers not attending counselling: ");
      while (i<=totalstud && absentrank!=-1)
      {
            scanf("%d", &absentrank);
            if (absentrank==-1)
```

```
                  i=totalstud+1;
            else
            {
                  if (findstudent(absentrank))
                        absent[i++]=absentrank;
                  else
                        printf("\n Roll No not found. Enter again or enter -1
to finish.\n");
            }
      }
      i=1;
      printf("\nSo now the ranks not attending counselling are: ");
      while (i<=100)
      {
            if (absent[i]!=0)
                  printf("%d ", absent[i++]);
            else if (absent[i]==0)
                  goto v;
      }
      v:  tokenlist();
}

int searchroommate(int rank)
{
      int i=1,found=0;
      while (absent[i]!=0)
      {
            if (absent[i]==rank)
            {
                  found=1;
                  absent[i]=-1;
                  break;
            }
            if ((absent[i]==0 || absent[i]==-1) && !(0<absent[i]<30))
                  found=-1;
            i++;
      }
      return found;
}

int chooseroom(int choice, int front)
{
      int i=0, allot=0, found=0, roommaterank;
      if (choice==1)
      {
            for (i=0; i<10; i++)
            {
                  if (Qblock_1bed[i]==0)
                  {
                        Qblock_1bed[i]=front;
                        printf("CONGRATULATIONS !! Student Rank %d has been
alloted 1 bed AC room in Q block.",present[front]);
                  allot=1;
                        break;
                  }
            }
            if (allot==0)
                  printf("\nSorry, Q Block is full.");
      }
    if (choice==2)
    {
```

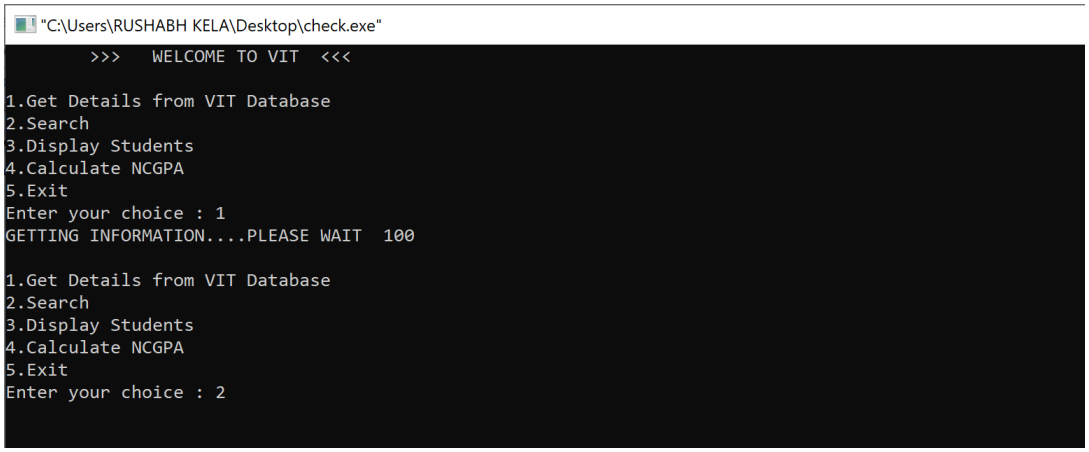
```
            z:
            printf("Enter NCGPA rank of roommate: ");
            scanf("%d", &roommaterank );
            found=searchroommate(roommaterank);
            if (found==0)
            {
                  printf("\nRoommate not found!!");
                  goto z;
            }
            if  (found==-1)
            {
                  printf("No roommate available ");
                  allot=0;
             return allot;
            }
            for (i=0; i<10; i++)
            {
                  if (Qblock_2bed[i][0]==0)
                  {
                        Qblock_2bed[i][0]=front;
                    Qblock_2bed[i][1]=roommaterank;
                        printf("CONGRATULATIONS !! Student Rank %d and %d
have been alloted 2 bed AC room in Q Block", present[front], roommaterank);
                        allot=1;
                  break;
                   }
            }
            if (allot==0)
                  printf("Sorry, Q Block is full");
      }
      return allot;
}

int main()
{
    int n,opt,rollno,counter=0,b,slot,total,flag=0;
    float cgpa;
    char name[100];
    FILE *fptr;
    char str[MAXCHAR],details[15][100];
    printf("\t>>>   WELCOME TO VIT  <<<\n");
    // TOTAL NUMBER OF STUDENTS IS 12
    cse_branch = (struct hash *)calloc(4, sizeof(struct hash));
    ece_branch = (struct hash *)calloc(4, sizeof(struct hash));
    mech_branch = (struct hash *)calloc(4, sizeof(struct hash));
    struct node *b1list=NULL, *b2list=NULL, *b3list=NULL;
    while(counter<=12)
    {
        printf("\n1.Get Details from VIT Database\n2.Search\n3.Display
Students\n4.Calculate NCGPA\n5.Exit\nEnter your choice : ");
        scanf("%d",&opt);
        switch(opt)
        {
        case 1:
            if(flag==1)
            {
                printf("Student Details Received.\n");
                break;
            }
            for (int i = 0; i <= 10000; i++)
            {
```

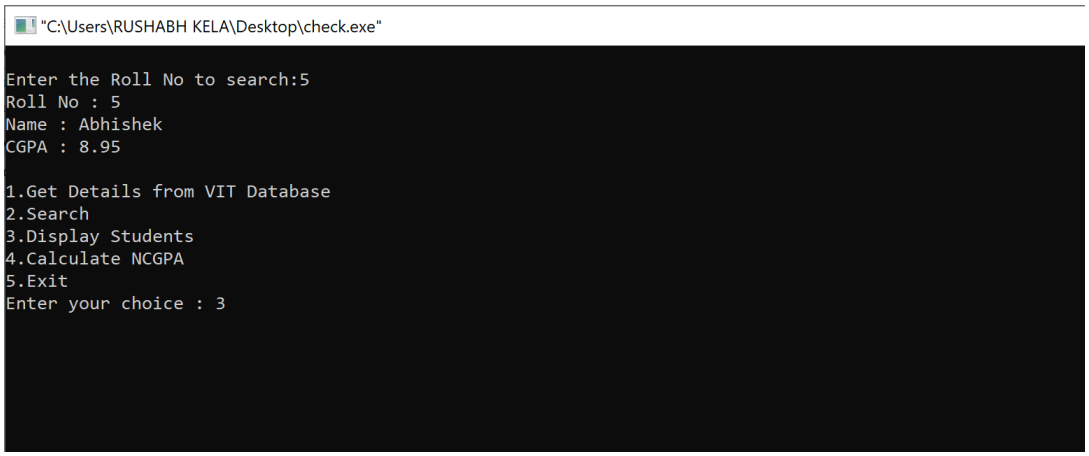
```
            printf("\rGETTING INFORMATION....PLEASE WAIT  %d", i/100);
        }
        printf("\n");
 //file handling is used, details of students stored as stored in a database
        fptr =
fopen("C:\\Users\\RUSHABHKELA\\Documents\\StuDetails.txt","r+");
        while (fgets(str, MAXCHAR, fptr) != NULL)
        {
            int j=0,ctr=0;
            for(int i=0;i<=(strlen(str));i++)
            {
                if(str[i]==' '||str[i]=='\0')
                {
                    details[ctr][j]='\0';
                    ctr++;
                    j=0;
                }
                else
                {
                    details[ctr][j]=str[i];
                    j++;
                }
            }
            rollno=atoi(details[0]);
            strcpy(name,details[1]);
            cgpa=atof(details[2]);
            b=atoi(details[3]);
            slot=atoi(details[4]);
            if(b==1)
                insert(cse_branch, rollno, name, cgpa, slot);
            else if(b==2)
                insert(ece_branch, rollno, name, cgpa, slot);
            else if (b==3)
                insert(mech_branch, rollno, name, cgpa, slot);
            counter++;
        }
        flag=1;
        break;
    case 2:
        system("cls");
        printf("\nEnter the Roll No to search:");
        scanf("%d", &rollno);
        b=search_details(cse_branch,rollno);
        if(b==0)
            b=search_details(ece_branch,rollno);
        if(b==0)
            b=search_details(mech_branch,rollno);
        if(b==0)
            printf("Student Not Found!\n");
        break;
    case 3:
        system("cls");
        printf("\tDISPLAYING ALL RECORDS ");
        printf("\n-------------------------------------------------");
        printf("\n\tCSE BRANCH");
        display(cse_branch);
        printf("\n-------------------------------------------------");
        printf("\n\tECE BRANCH");
        display(ece_branch);
        printf("\n-------------------------------------------------");
        printf("\n\tMECHANICAL BRANCH");
```

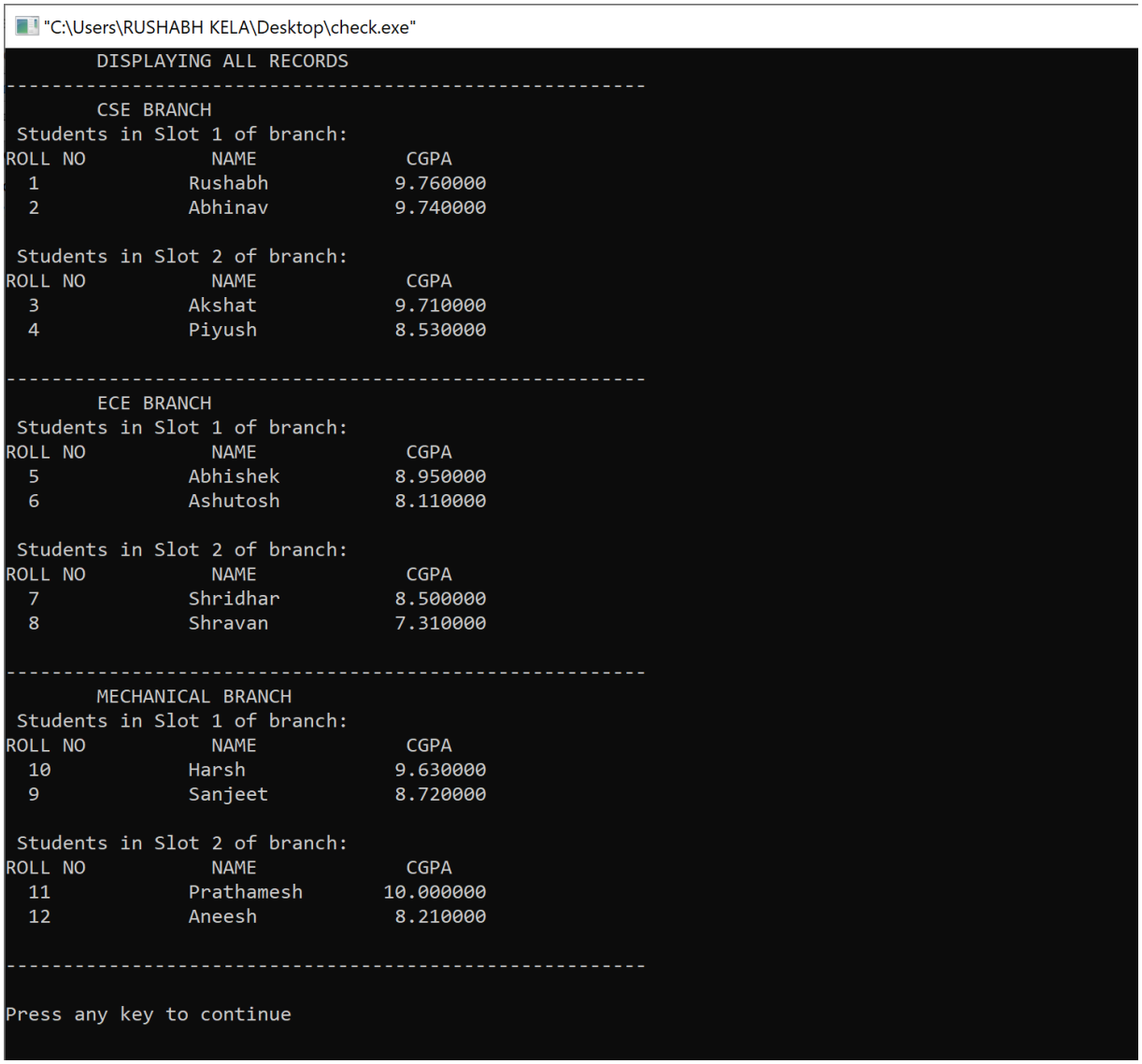
```
            display(mech_branch);
            printf("\n-----------------------------------------------------");
            printf("\n\nPress any key to continue ");
            getch();
            system("cls");
            break;
        case 4:
            counter=3*n+1;
            break;
        case 5:
            exit(0);
        default:
            printf("Enter correct option!!!\n");
            break;
        }
    }
    system("cls");
    printf("\n\tDISPLAYING BRANCH TOPPER DETAILS\n");
    topper(cse_branch,"CSE");
    topper(ece_branch,"ECE");
    topper(mech_branch,"MECHANICAL");
    printf("\n\nPress any key to continue: ");
    getch();
    system("cls");
    b1list=ncgpalist(cse_branch);
    b2list=ncgpalist(ece_branch);
    b3list=ncgpalist(mech_branch);

    printf("\nNCGPA list of CSE students:\n");
    displaylist(b1list);
    printf("\nNCGPA list of ECE students:\n");
    displaylist(b2list);
    printf("\nNCGPA list of MECHANICAL students:\n");
    displaylist(b3list);
    mergeall(b1list,b2list,b3list);
    system("pause");
    system("cls");
    printf("NCGPA RANK LIST:\n");
    total=printlist();
    printf("\nPROCEED TO ROOM COUNSELLING DAY? IF YES PRESS 1, ELSE
EXIT.\n");
    scanf("%d",&opt);
    if(opt!=1)
        exit(0);
    system("cls");
    printf(">>>>  WELCOME TO VIT HOSTEL ROOM COUNSELLING  <<<<\n");
    printf("\nProceed for registration:\n");
    registration(total);
    printf("\t\tROOMS AVAILABLE\n");
    printf("\tROOM TYPE\t\tTOTAL ROOMS\tCHOICE\n");
    printf("   Q BLOCK 1 BED AC\t\t\t10\t  1\n   Q BLOCK 2 BED AC\t\t\t10\t
2\n");
    int front=1;
    while (present[front]!=0)
    {
        int allot=0, choice;
        while (allot==0)
        {
            printf("\n\nStudent rank %d Choose your room type : ",
present[front]);
            scanf("%d", &choice);
```

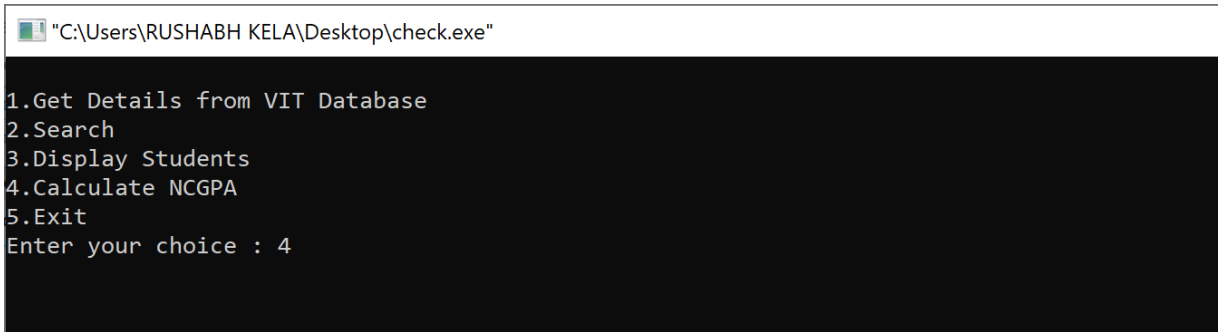
```
                allot=chooseroom(choice, front);
            if (allot==1)
                        front++;
            }
      }

    printf("\n\n    COUNSELLING PROCESS OVER !!\n");
    printf("A provisional hostel room allotment letter will be available in
VTOP under Hostels Menu. This allotment will be confirmed subject to payment
of balance Hostel Fee and entire Tuition fees for the academic year 2020-
21.\n");
    printf("\n----------------THE END----------------");
    return 0;
}
```

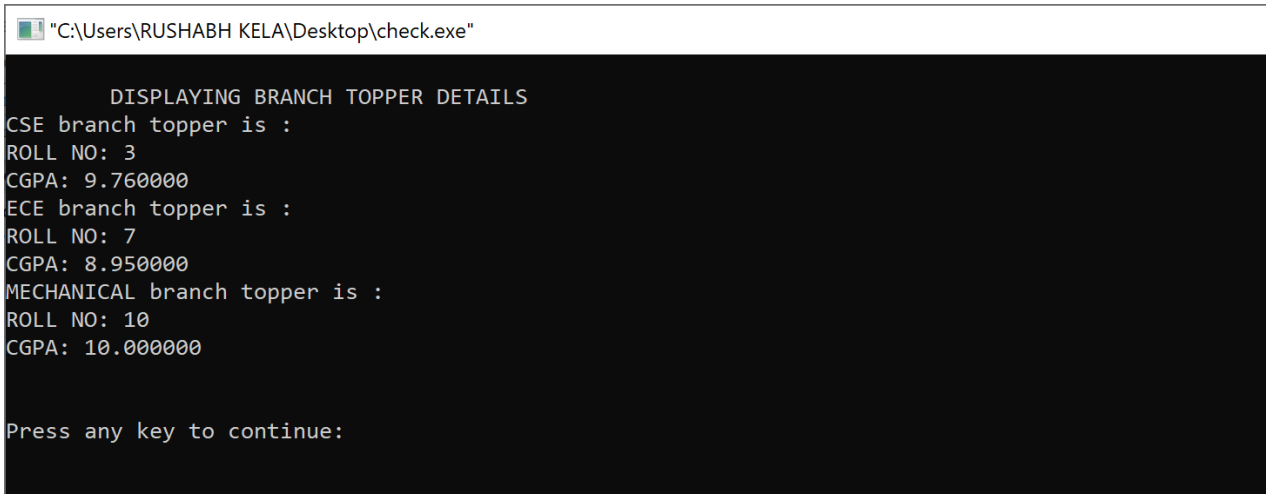
OUTPUT :

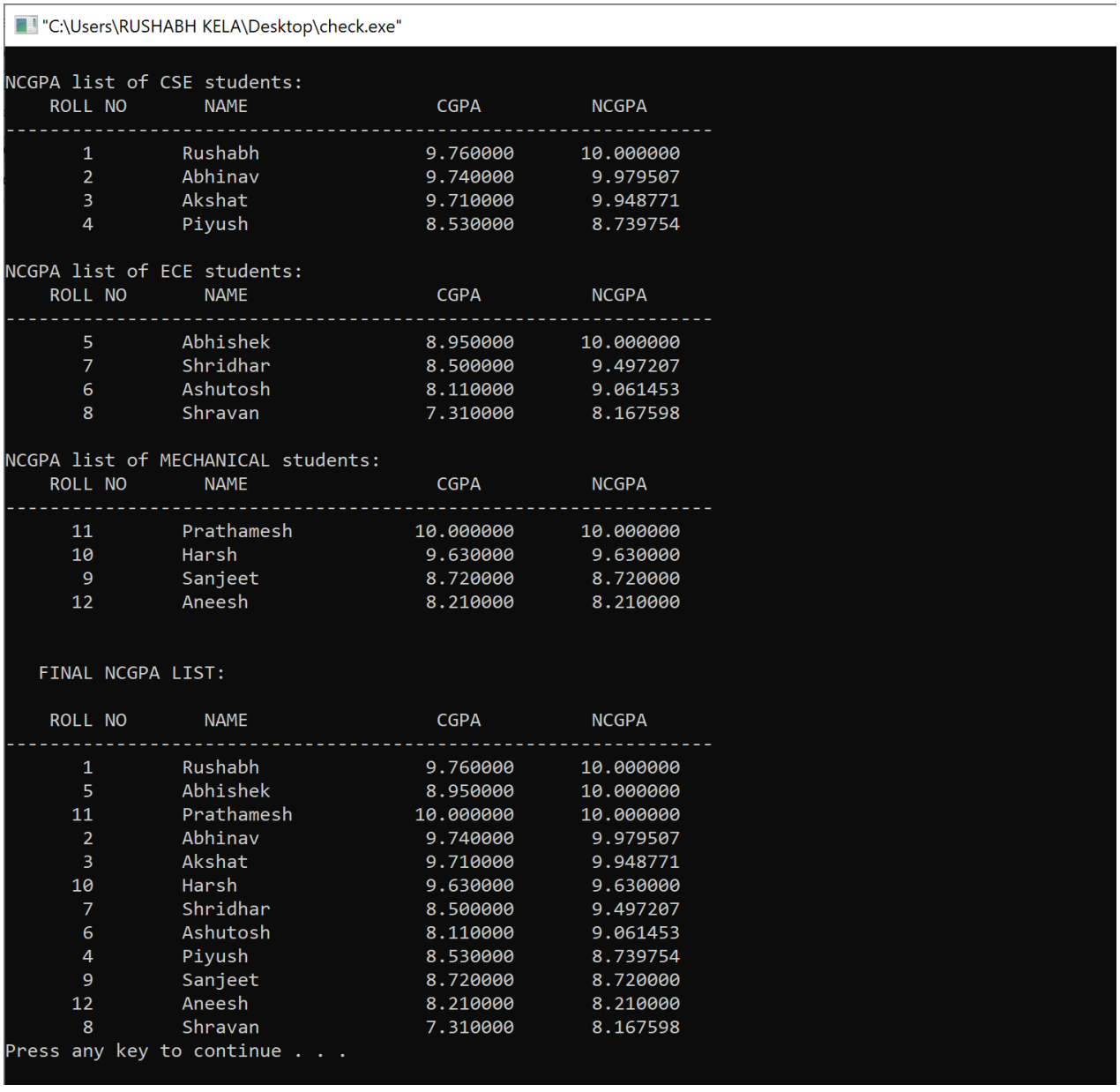





"C:\Users\RUSHABH KELA\Desktop\check.exe"
DISPLAYING ALL RECORDS
CSE BRANCH
Students in Slot 1 of branch:
ROLL NO NAME CGPA
1 Rushabh 9.760000
2 Abhinav 9.740000
Students in Slot 2 of branch:
ROLL NO NAME CGPA
3 Akshat 9.710000
4 Piyush 8.530000
ECE BRANCH
Students in Slot 1 of branch:
ROLL NO NAME CGPA
5 Abhishek 8.950000
6 Ashutosh 8.110000
Students in Slot 2 of branch:
ROLL NO NAME CGPA
7 Shridhar 8.500000
8 Shravan 7.310000
MECHANICAL BRANCH
Students in Slot 1 of branch:
ROLL NO NAME CGPA
10 Harsh 9.630000
9 Sanjeet 8.720000
Students in Slot 2 of branch:
ROLL NO NAME CGPA
11 Prathamesh 10.000000
12 Aneesh 8.210000
Press any key to continue


"C:\Users\RUSHABH KELA\Desktop\check.exe"
1.Get Details from VIT Database
2.Search
3.Display Students
4.Calculate NCGPA
5.Exit
Enter your choice : 4


"C:\Users\RUSHABH KELA\Desktop\check.exe"
DISPLAYING BRANCH TOPPER DETAILS
CSE branch topper is :
ROLL NO: 3
CGPA: 9.760000
ECE branch topper is :
ROLL NO: 7
CGPA: 8.950000
MECHANICAL branch topper is :
ROLL NO: 10
CGPA: 10.000000
Press any key to continue:

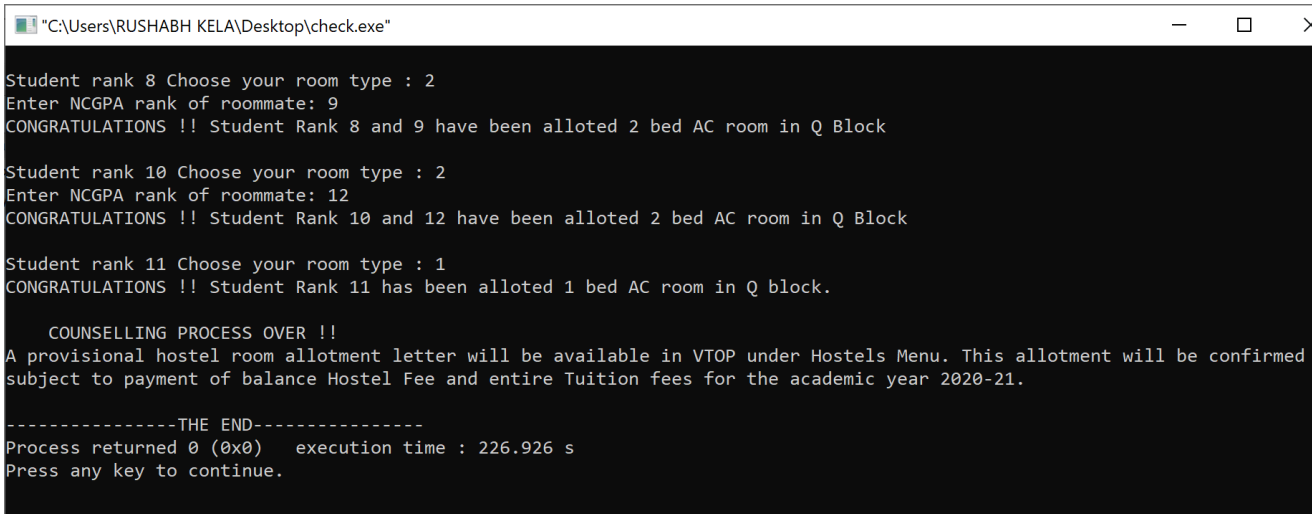

"C:\Users\RUSHABH KELA\Desktop\check.exe"
NCGPA list of CSE students:
ROLL NO NAME CGPA NCGPA
1 Rushabh 9.760000 10.000000
2 Abhinav 9.740000 9.979507
3 Akshat 9.710000 9.948771
4 Piyush 8.530000 8.739754
NCGPA list of ECE students:
ROLL NO NAME CGPA NCGPA
5 Abhishek 8.950000 10.000000
7 Shridhar 8.500000 9.497207
6 Ashutosh 8.110000 9.061453
8 Shravan 7.310000 8.167598
NCGPA list of MECHANICAL students:
ROLL NO NAME CGPA NCGPA
11 Prathamesh 10.000000 10.000000
10 Harsh 9.630000 9.630000
9 Sanjeet 8.720000 8.720000
12 Aneesh 8.210000 8.210000
FINAL NCGPA LIST:
ROLL NO NAME CGPA NCGPA
1 Rushabh 9.760000 10.000000
5 Abhishek 8.950000 10.000000
11 Prathamesh 10.000000 10.000000
2 Abhinav 9.740000 9.979507
3 Akshat 9.710000 9.948771
10 Harsh 9.630000 9.630000
7 Shridhar 8.500000 9.497207
6 Ashutosh 8.110000 9.061453
4 Piyush 8.530000 8.739754
9 Sanjeet 8.720000 8.720000
12 Aneesh 8.210000 8.210000
8 Shravan 7.310000 8.167598
Press any key to continue . . .

"C:\Users\RUSHABH KELA\Desktop\check.exe"

```
NCGPA RANK LIST:


       RANK    ROLL NO     NCGPA
-----------------------------------

       1         1       10.000000
       2         5       10.000000
       3         11      10.000000
       4         2       9.979507
       5         3       9.948771
       6         10      9.630000
       7         7       9.497207
       8         6       9.061453
       9         4       8.739754
       10        9       8.720000
       11        12      8.210000
       12        8       8.167598
PROCEED TO ROOM COUNSELLING DAY? IF YES PRESS 1, ELSE EXIT.
1
```

"C:\Users\RUSHABH KELA\Desktop\check.exe"

```
>>>>  WELCOME TO VIT HOSTEL ROOM COUNSELLING  <<<<

Proceed for registration:

Total students is: 12
Enter the rankers not attending counselling: 2 5 9 12 -1

So now the ranks not attending counselling are: 2 5 9 12

The student rankers taking part in counselling are:
 1
 3
 4
 6
 7
 8
 10
 11
               ROOMS AVAILABLE
       ROOM TYPE               TOTAL ROOMS     CHOICE
   Q BLOCK 1 BED AC                    10        1
   Q BLOCK 2 BED AC                    10        2


Student rank 1 Choose your room type : 2
Enter NCGPA rank of roommate: 2
CONGRATULATIONS !! Student Rank 1 and 2 have been alloted 2 bed AC room in Q Block

Student rank 3 Choose your room type : 2
Enter NCGPA rank of roommate: 5
CONGRATULATIONS !! Student Rank 3 and 5 have been alloted 2 bed AC room in Q Block

Student rank 4 Choose your room type : 1
CONGRATULATIONS !! Student Rank 4 has been alloted 1 bed AC room in Q block.

Student rank 6 Choose your room type : 1
CONGRATULATIONS !! Student Rank 6 has been alloted 1 bed AC room in Q block.

Student rank 7 Choose your room type : 1
CONGRATULATIONS !! Student Rank 7 has been alloted 1 bed AC room in Q block.
```



FILE USED AS STUDENT DETAILS DATA :