**Mini Project**

**On**

**IMSEC REGISTRATION HUB USING C**

**Submitted By:**

1.Ayush Verma (2001430100073)

2.Gungun Gupta (2001430100092)

3.Mehak Sangal (2001430100127)

4.Chehak Rajput (2001430100077)

**(2nd year section 2)**

**in**



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**IMS ENGINEERING COLLEGE, GHAZIABAD**



**Affiliated to Dr. A.P.J Abdul Kalam Technical University, Lucknow**

**2021-2022**

**Submitted To:**

Mr. Kishor Kunal Keshri

(Assistant Professor, CSE Dept.)



**IMS ENGINEERING COLLEGE, GHAZIABAD**

CERTIFICATE

# This is to certify that the project entitled “IMSEC Registration Hub” carried out by Gungun Gupta, Ayush Verma, Mehak Sangal, Chehak Rajput under my supervision at Department of DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING IMS ENGINEERING COLLEGE, GHAZIABAD. The work is original, as it has not been submitted earlier either in part or full for any purpose before.

**Mr. Kishor Kunal Keshri**

**(Assistant Professor, CSE Dept.)**

**Contents**

1.DECLARATION………………………………………………………………………………………………3

2. PREFACE…………………………………………………………………………..…………….………….. 4

3.REGISTRATION HUB ? ……………………………………………………….………………………….6

4. STRATEGY………………………………………………………………….………………….………...... 7

5. PROGRAM FOR IMSEC REGISTRATION HUB.…………………………………………..… 11

6. OUTPUTS……………………………………………………………………………………………………38

6. CONCLUSION………………………………………………………………………………….………….47

**DECLARATION**

We, hereby declare that the work presented in this dissertation entitled “IMSEC Registration Hub” has been done by us, and this dissertation embodies our own work.

Name of team members:

Gungun Gupta (2001430100092)

Ayush Verma (2001430100073)

Mehak Sangal (2001430100127)

Chehak Rajput (2001430100077)

Submitted to:

MR. KISHOR KUNAL KESHRI

(Assistant Professor, CSE Dept.)

Preface

Events create opportunities for people to connect with an area, spend time together, celebrate and experience the diversity of cultures and foster creativity and innovation. They allow a community to come alive and provide an opportunity for a destination to showcase its tourism experience and increase economic activity. Events contribute significantly to community building, lifestyle and leisure enhancement, cultural development, tourism promotion and increased visitation, volunteer participation, fundraising and economic development. Most importantly, events create a sense of fun and vibrancy, resulting in a strong sense of community connectivity, pride and a sense of place.

Event registration system is used to register all the activity related to event. In any event many students register simultaneously and it is very hard to manage these complexities. It is also important for event organizer that he has all the details of these participants so that he can contact them any time to plan an event at given time. To manage all the activity, we have developed this software.

By this program we can register the students in an event and sort them according to their year, branch, events they participated in, etc. We can even count the number of students participated in the event.

This program is generally made in the name of our respected organisation, so that we can successfully register any number of students and sort them respectively. Through this little attempt of ours, we want to showcase our set of skills.

IMSEC Registration Hub is a program that helps on to sort the participants of an event. It is both user and client friendly. One can check before registering, about the other participants also. It uses file handling concept to store all the registrations. One can register more than once i.e., in continuation or even dis continuously.

To manage all the activity, we have developed this software.

Registration hub

The major drivers of this strategy are the cultural, social and environmental benefits of community events. A clear vision is needed to provide direction to activity in the future.

Every event must have a clearly stated overall aim; otherwise, the event should not happen. Events demand a lot of concentrated effort and commitment. This commitment can only come out of a genuine belief among all participants that the aims are worthwhile and that they will be beneficial in the long term. As well as an overall purpose any specific event must have its own set of objectives, these must be clear and be set down in a way which will allow you to judge the success of the event after completion.

Creativity is also an attitude, the ability to accept change and newness, a willingness to play with ideas and possibilities, a flexibility of outlook, the habit of enjoying the good, while looking for ways to improve it, we are socialized into accepting only a small number of permissible or normal things Creative in event management helps companies by opening up new opportunities for problem solving and growth that more conventional methods would not allow for. New perspectives on problems from a creative approach can lead to new and perhaps previously unheard-of solutions.

It provides a roadmap to guide actions and is continually modified as conditions change and new opportunities or threats emerge. While it should be fundamentally targeted at guiding regular operations, it should also be adaptable to account for new opportunities and challenges.

Strategy

IMSEC REGISTRATION HUB works in the following steps:



* We have created functions to perform each and every specific task. Functions like, reg(), show(), reg\_no(),etc. these functions help us to perform each and every task differently and efficiently.
* Structures are made to store the names of all students, roll numbers of the participant, branch, year, and the event they participate in and count the number of participants that are registered in this event.
* In the main block, we will print the instructions whether to make a registration, see the details of the students or exit out of the program. We run a switch case in this block to make choices and execute the following block which is asked for. We then enter a case 4 in which we thank the user for using our program.



* Then we pass on to the function reg(), in this function we make the user enter the first name, last name, roll number, branch and the event in which the user wants to participate. Then at last we increase the value of i by 1. This is the way the users ends making a registration in the program.
* After the function reg() we pass the year() function. The year() function checks the participant’s year and compares it with 1,2,3 and 4. Therefore, then makes a file year wise and stores the name and the entry corresponding to it in the same file. This function uses file handling.
* The function trait(), checks the participant’s branch and compares it with cse, cs, it, mech, ec, biotech and mba . If any other branch is added it will be added in a separate file. Therefore, then makes a file branch wise and stores the name and the entry corresponding to it in the same file. This function uses file handling. Now using the above two functions we can sort year and branch wise.
* The function event(), checks the participant’s event in which he/she may have participated and compares it with the give choice of the events mainly singing, dancing, poetry and anchoring .Therefore, then makes a file event wise and stores the name and the entry corresponding to it in the same file. This function uses file handling. Now using the above two functions we can sort year, event wise and branch wise.
* The next function count(), counts the total number of registration and stores in a variable then we increment the value of that variable by 1. Then it enters the value of the total count to a file. This function basically tells the total number of registrations done through the entire program.
* The function reg\_no() stores the number of registrations. It uses file handling. It opens the file in which it stores the number of registration and prints it. If no data is found it will print the message respectively. This function is just an addition to the previous function count.
* We made the next function named show(), first of all we check by year, this function shows the students detail sorted by year, branch and event respectively. This takes the user input according to its needs and then access the corresponding block. It prints the name and the entry corresponding to it in the same file. If there is no such entry by users choice it will print no data found or a message related to the error.
* We made the next function named show(), first of all we check by branch, this function shows the students detail sorted by year, branch and event respectively. This takes the user input according to its needs and then access the corresponding block. It prints the name and the entry corresponding to it in the same file. If there is no such entry by users choice it will print no data found or a message related to the error.
* We made the next function named show(), first of all we check by event, this function shows the students detail sorted by year, branch and event respectively. This takes the user input according to its needs and then access the corresponding block. It prints the name and the entry corresponding to it in the same file. If there is no such entry by users choice it will print no data found or a message related to the error.

PROGRAM FOR IMSEC Registration HUB

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<conio.h>

int i=0,j=0,k=0,l=0,m=0;

struct stu

{

int roll, year;

char branch[10],event[10];

char f\_name[10], l\_name[10];

};

struct new

{

int cnt;

};

FILE \*ptr;

void note(struct stu a);

void reg();

void show();

void year(struct stu a);

void trait(struct stu b);

void eventing(struct stu c);

void reg\_no();

void count();

int main()

{

int choice;

while(1)

{

system("cls");

printf("##### Welcome To My Program #####\n\n");

printf("\*\*\* Enter Student Details \*\*\*\n");

printf("1. Enter student data\n");

printf("2. Display Student data\n");

printf("3. Display no of registration \n");

printf("4. End Program\n");

printf("# Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1:

reg();

break;

case 2:

show();

break;

case 3:

reg\_no();

break;

case 4:

printf("\n\n\t\t### Thank You For using This Program ###\n\n");

printf("\t\t This Program is Made by \n");

printf("\t\t1. Ayush Verma(2001430100073)\n");

printf("\t\t2. Gungun Gupta(2001430100092)\n");

printf("\t\t3. mehak sangal(2001430100127)\n");

printf("\t\t4. Chehak Rajput(2001430100077)\n");

exit(0);

}

}

return 0;

}

void reg()

{

int q;

struct stu s;

system("cls");

printf("<----: Enter %d Student Detail :---->\n",i+1);

printf("Enter Student First Name : ");

fflush(stdin);

scanf("%s",&s.f\_name);

printf("Enter Student Last Name : ");

fflush(stdin);

scanf("%s",&s.l\_name);

printf("Enter student's branch-->(Write in Word)\n(1.cse || 2.cs || 3.it || 4.biotech || 5.mech || 6.ec || 7.mba ): ");

fflush(stdin);

scanf("%s",&s.branch);

printf("Enter Student's Year (1-4) : ");

scanf("%d",&s.year);

printf("Enter Student's Roll Number (Last 3 Digits) : ");

scanf("%d",&s.roll);

printf("Enter event's name in which you want to participate-->(Write in Word)\n( 1.Dancing || 2.Singing || 3.Poetry || 4.Anchoring ): ");

fflush(stdin);

scanf("%s",&s.event);

i++;

year(s);

trait(s);

eventing(s);

count();

}

void year(struct stu a)

{

if(a.year == 1)

{

ptr = fopen("data\_1.txt","a+");

}

else if(a.year == 2)

{

ptr = fopen("data\_2.txt","a+");

}

else if(a.year == 3)

{

ptr = fopen("data\_3.txt","a+");

}

else

{

ptr = fopen("data\_4.txt","a+");

}

fprintf(ptr,"%s %s\t%s\t%d\t%d\t%s\n",a.f\_name,a.l\_name,a.branch,a.year,a.roll,a.event);

fclose(ptr);

}

void trait(struct stu b)

{

if((strcmp("cse",b.branch) == 0))

{

ptr = fopen("branch\_1.txt","a+");

}

else if((strcmp("cs",b.branch) == 0))

{

ptr = fopen("branch\_2.txt","a+");

}

else if((strcmp("it",b.branch) == 0))

{

ptr = fopen("branch\_3.txt","a+");

}

else if((strcmp("biotech",b.branch) == 0))

{

ptr = fopen("branch\_4.txt","a+");

}

else if((strcmp("mech",b.branch) == 0))

{

ptr = fopen("branch\_5.txt","a+");

}

else if((strcmp("ec",b.branch) == 0))

{

ptr = fopen("branch\_6.txt","a+");

}

else

{

ptr = fopen("branch\_7.txt","a+");

}

fprintf(ptr,"%s %s\t%s\t%d\t%d\t%s\n",b.f\_name,b.l\_name,b.branch,b.year,b.roll,b.event);

fclose(ptr);

}

void eventing(struct stu c)

{

if((strcmp("dancing",c.event) == 0))

{

ptr = fopen("event\_1.txt","a+");

}

else if((strcmp("singing",c.event) == 0))

{

ptr = fopen("event\_2.txt","a+");

}

else if((strcmp("anchoring",c.event) == 0))

{

ptr = fopen("event\_3.txt","a+");

}

else

{

ptr = fopen("event\_4.txt","a+");

}

fprintf(ptr,"%s %s\t%s\t%d\t%d\t%s\n",c.f\_name,c.l\_name,c.branch,c.year,c.roll,c.event);

fclose(ptr);

}

void count()

{

struct new n;

ptr = fopen("count.txt","w");

n.cnt=m+1;

fprintf(ptr,"%d",n.cnt);

m++;

fclose(ptr);

}

void reg\_no()

{

struct new n;

ptr = fopen("count.txt","r");

if(ptr == NULL)

{

printf("data not found");

getch();

}

else

{

fscanf(ptr,"%d",&n.cnt);

printf("NO OF REGISTRATION ARE : %d",n.cnt);

getch();

fclose(ptr);

}

}

void show()

{

int tip,ch;

struct stu sh;

system("cls");

while(1)

{

system("cls");

printf("1. get student detail by Year :\n");

printf("2. get student detail by branch :\n");

printf("3. get student detail by event: \n");

printf("4. exit\n");

printf("enter choice :--->");

scanf("%d",&ch);

if(ch==1)

{

system("cls");

printf("which Year student you want to search : ");

scanf("%d",&tip);

if(tip == 1)

{

ptr = fopen("data\_1.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if(tip == 2)

{

ptr = fopen("data\_2.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if(tip == 3)

{

ptr = fopen("data\_3.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if(tip == 4)

{

ptr = fopen("data\_4.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else

{

printf("### INVALID YEAR ###\n");

getch();

show();

}

}

else if(ch==2)

{

system("cls");

char p[20];

printf(" 1.cse\n 2.cs\n 3.it\n 4.biotech\n 5.mech\n 6.ec\n 7.mba\n which branch's student you want to search(1-7): ");

fflush(stdin);

gets(p);

if((strcmp(p,"1")==0))

{

ptr = fopen("branch\_1.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"2")==0))

{

ptr = fopen("branch\_2.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"3")==0))

{

ptr = fopen("branch\_3.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"4")==0))

{

ptr = fopen("branch\_4.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"5")==0))

{

ptr = fopen("branch\_5.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"6")==0))

{

ptr = fopen("branch\_6.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(p,"7")==0))

{

ptr = fopen("branch\_7.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else

{

printf("### INVALID BRANCH ###\n");

getch();

show();

}

}

else if(ch==3)

{

system("cls");

char z[20];

printf("1.dancing\n 2.singing\n 3.anchoring\n 4.poetry\n which event's student you want to search(1-4): ");

fflush(stdin);

gets(z);

if((strcmp(z,"1")==0))

{

ptr = fopen("event\_1.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(z,"2")==0))

{

ptr = fopen("event\_2.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(z,"3")==0))

{

ptr = fopen("event\_3.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else if((strcmp(z,"4")==0))

{

ptr = fopen("event\_4.txt","r");

if( ptr == NULL)

{

printf("####### DATA NOT FOUND ######");

getch();

}

else

{

printf("NAME \t\t\tBRANCH\t\tYEAR\t\tROLL\_NO\t\tEVENT \n");

while(fscanf(ptr,"%s%s%s%d%d%s",sh.f\_name,sh.l\_name,sh.branch,&sh.year,&sh.roll,sh.event) !=EOF)

{

printf("%s %s\t\t%s\t\t%d\t\t%d\t\t%s\n",sh.f\_name,sh.l\_name,sh.branch,sh.year,sh.roll,sh.event);

}

getch();

}

}

else

{

printf("### INVALID EVENT ###");

getch();

main();

}

}

else

{

main();

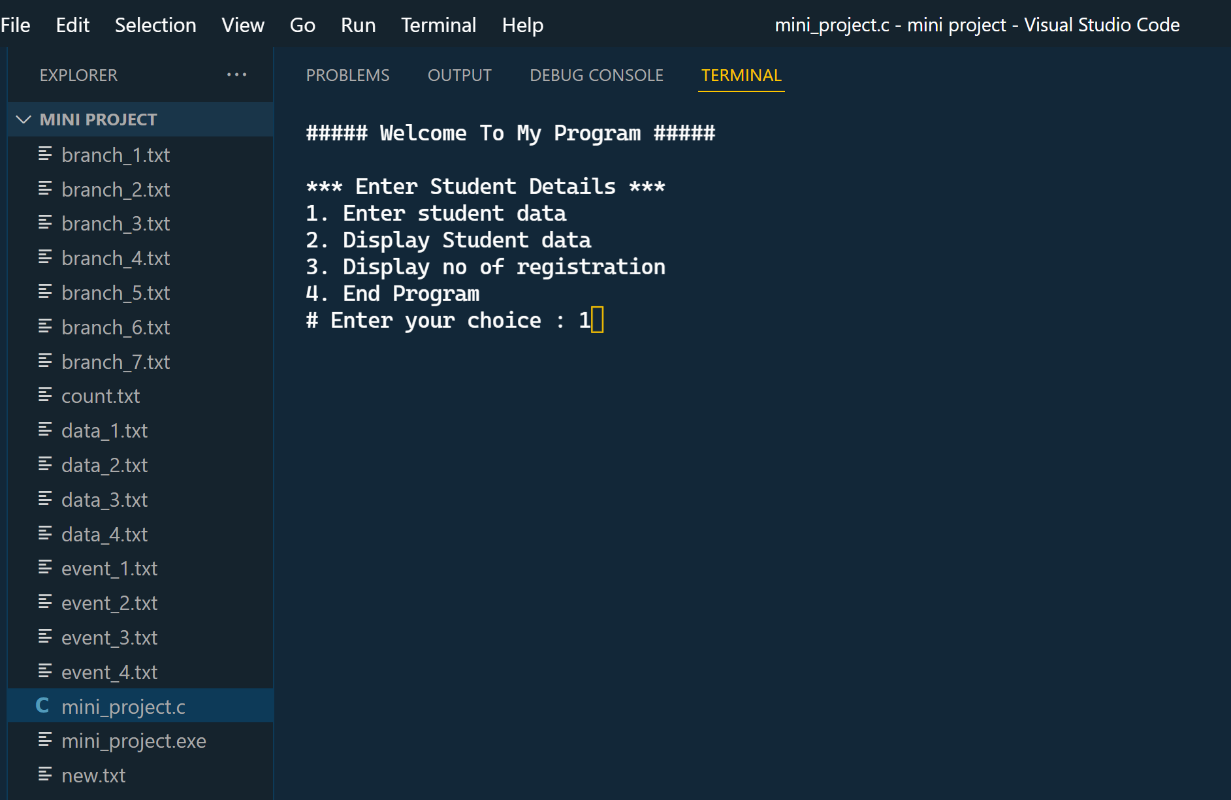
}

}

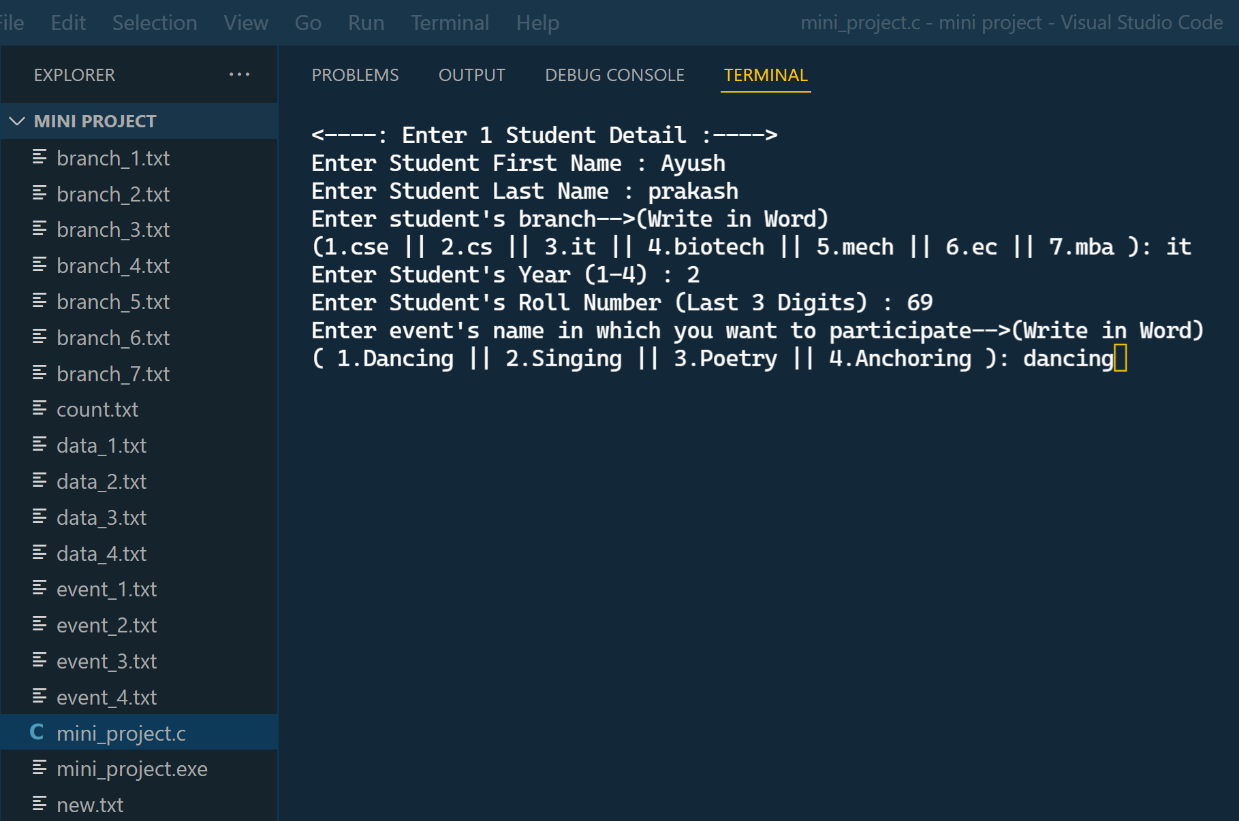
}

Output

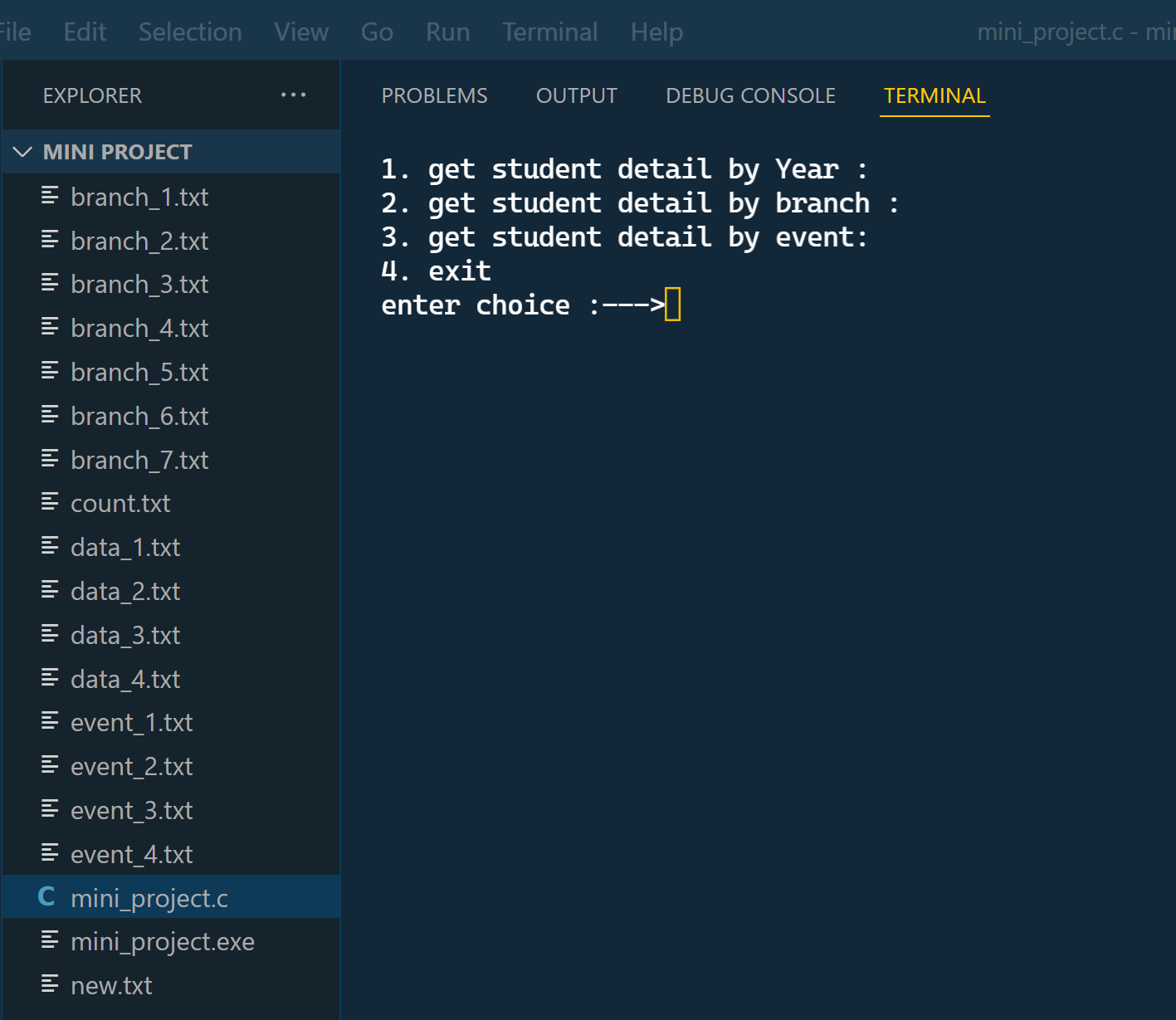
* WELCOME TO IMSEC REGISTRATION HUB



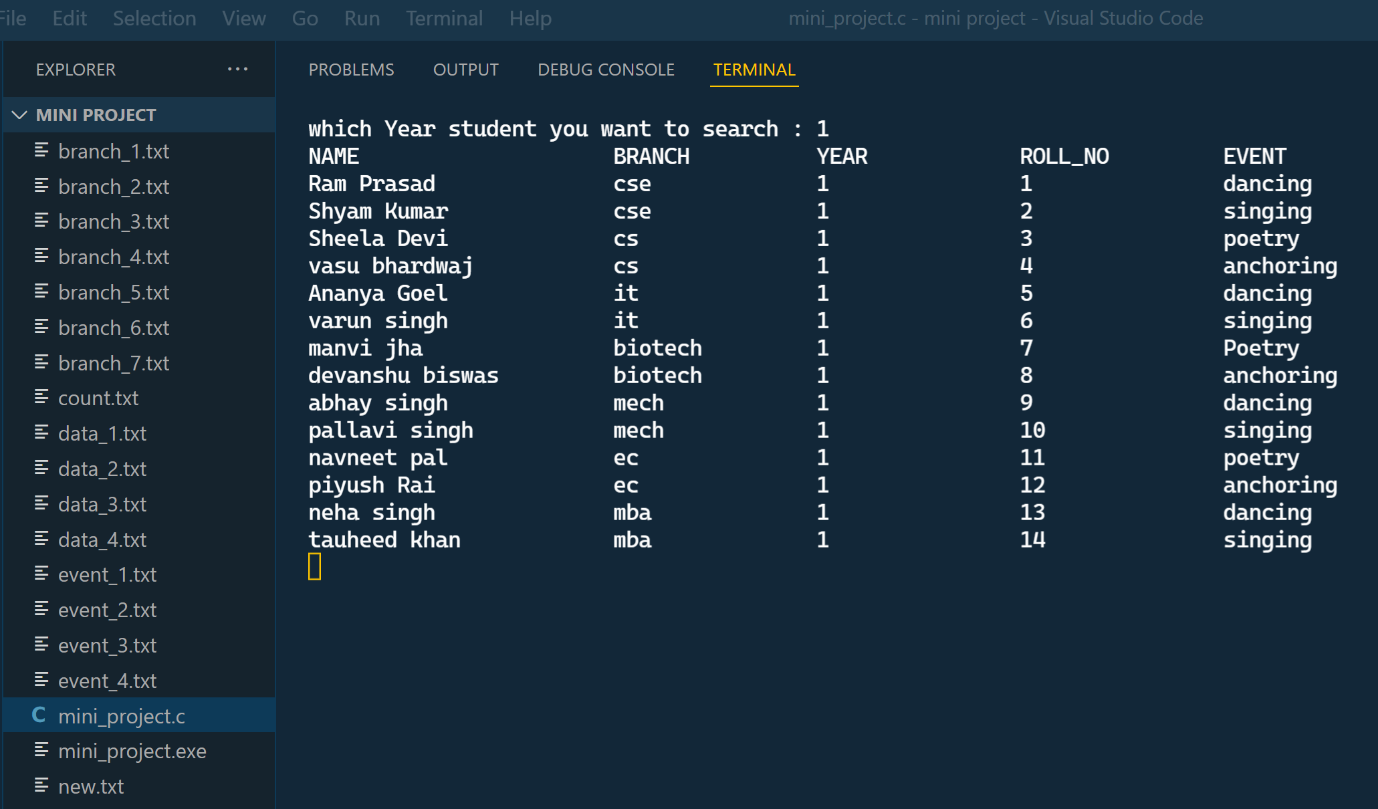
* LET’S REGISTER SOME STUDENTS



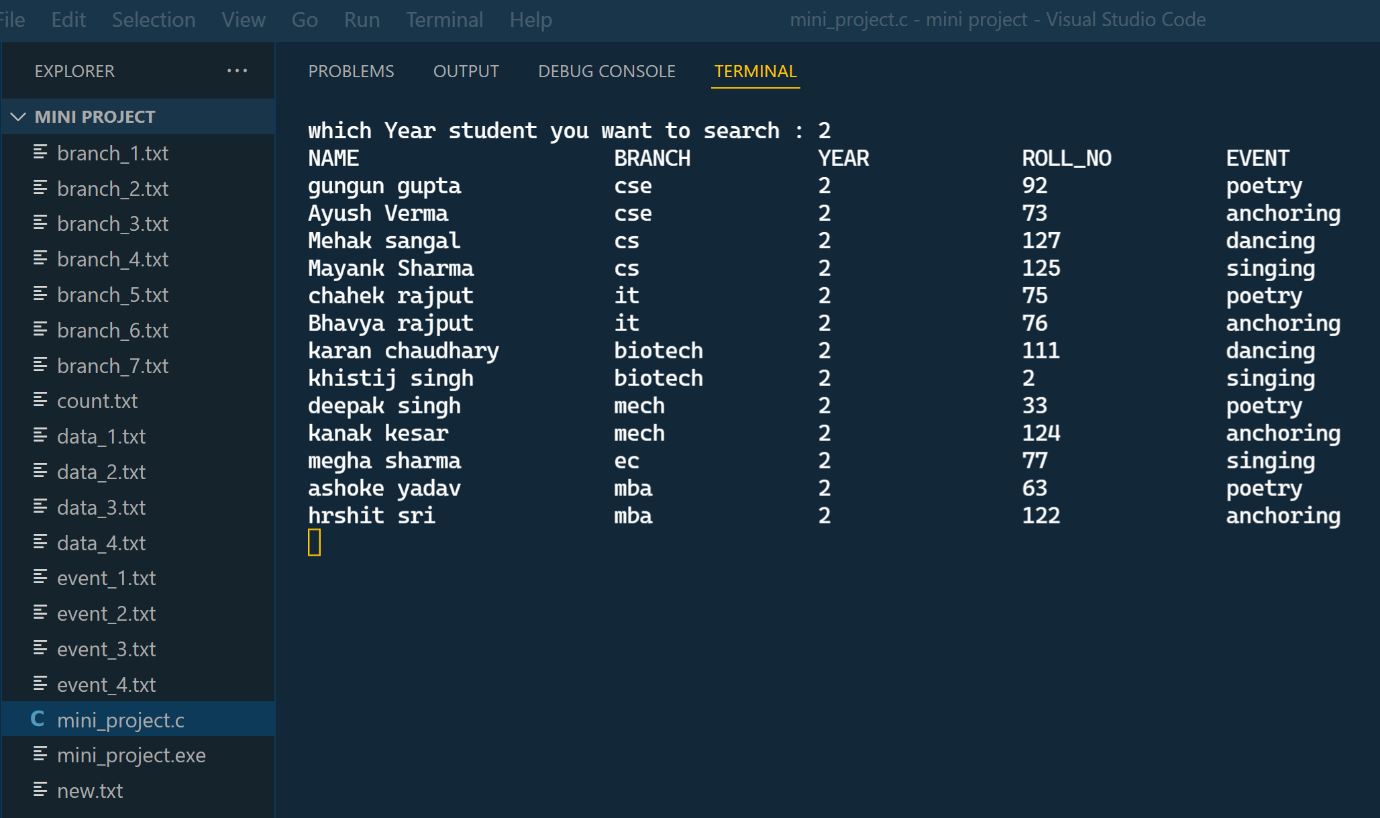
* NOW LET’S SORT OUR PARTICIPANTS



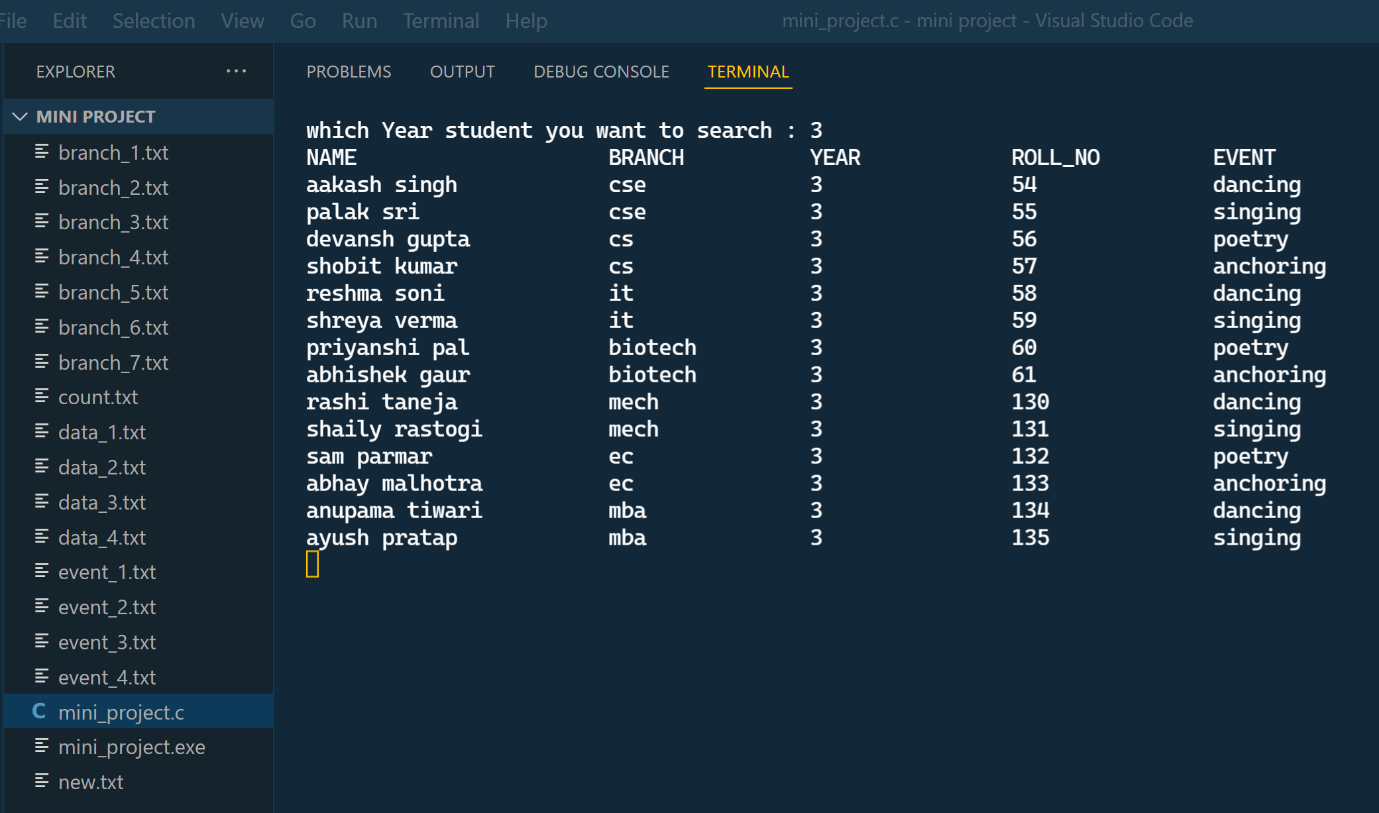
* AFTER ALL REGISTRATIONS LET’S SEARCH FOR THE STUDENTS OF YEAR 1.



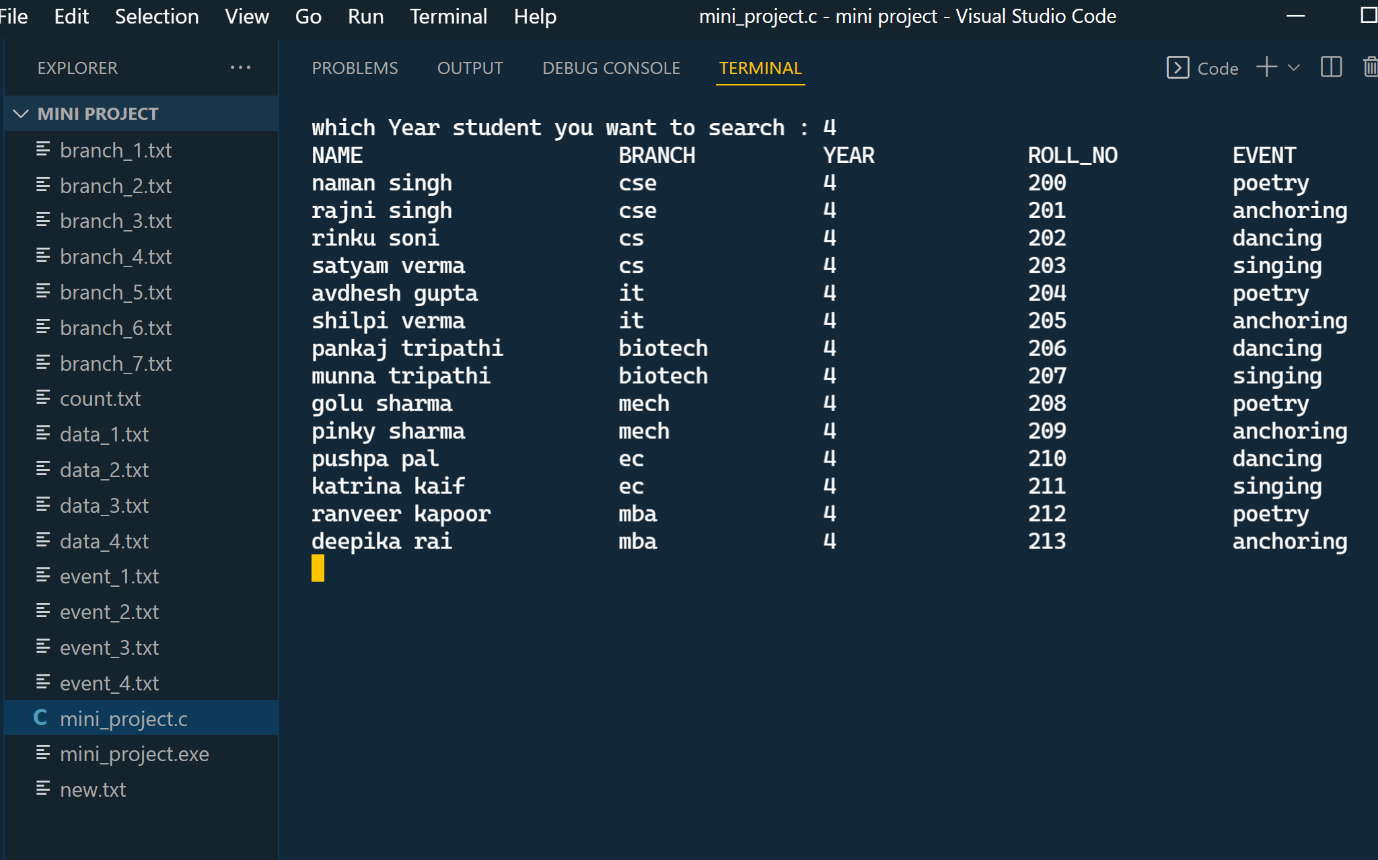
* SIMILARLY, WHEN WE ENTER 2ND YEAR



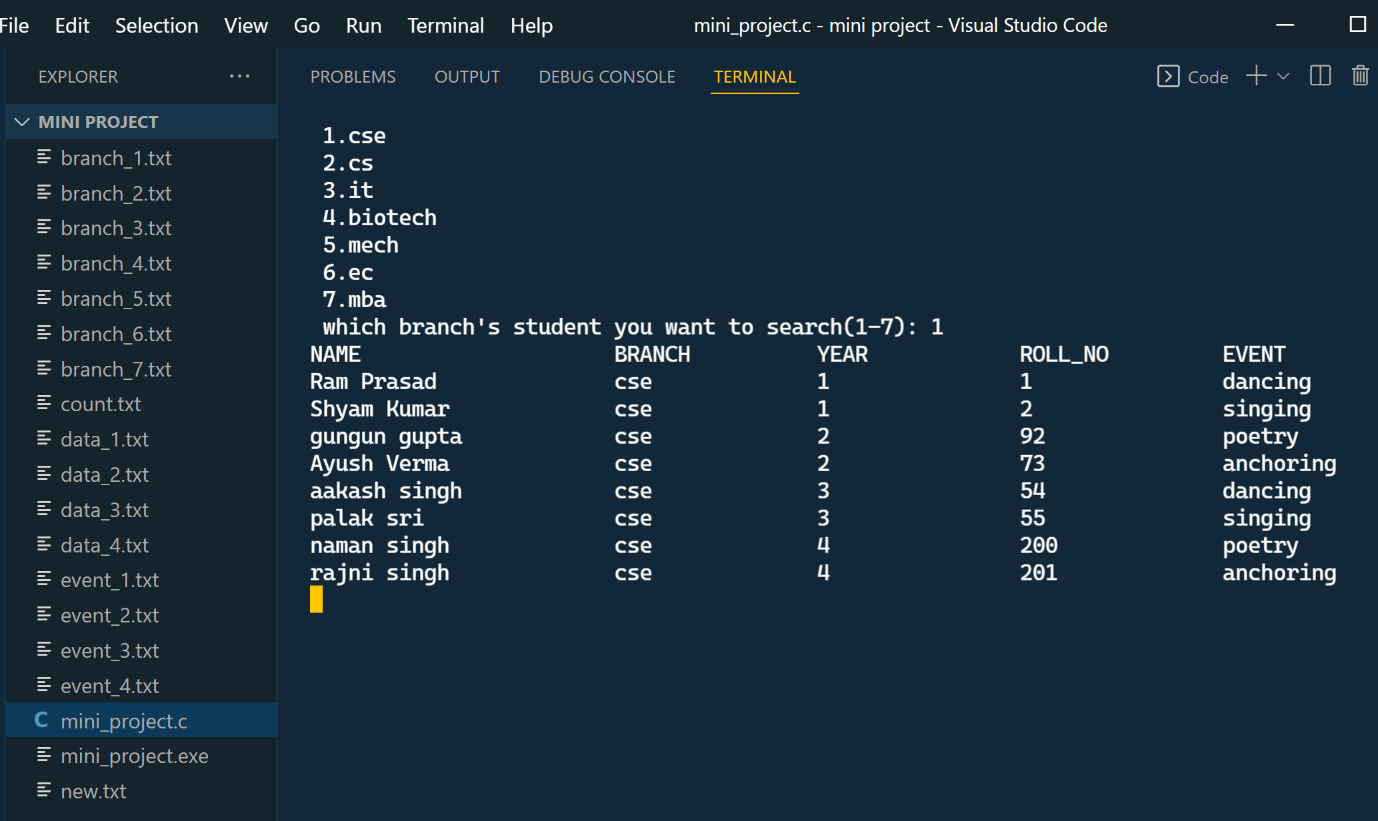
* LET’S TRY FOR 3RD YEAR STUDENT



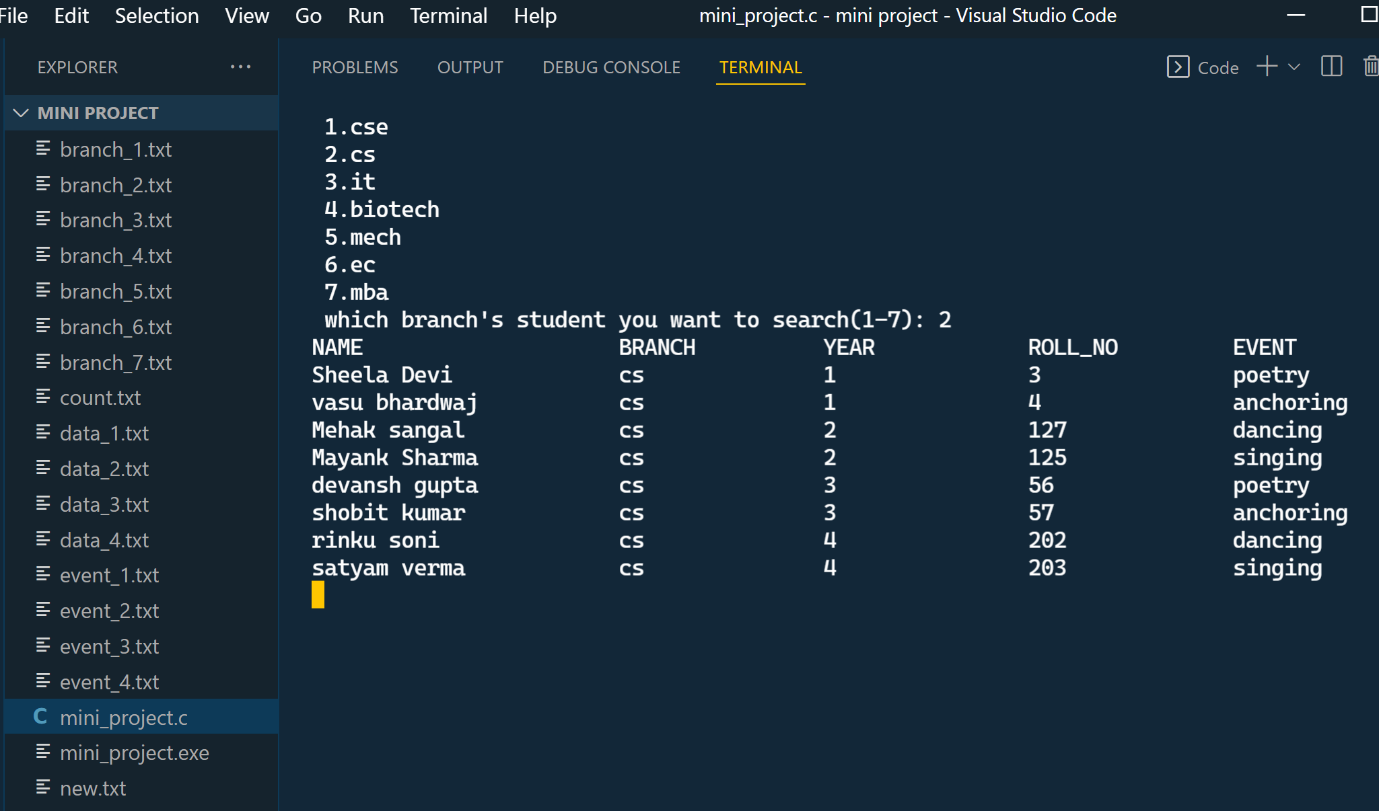
* LASTLY 4TH YEAR STUDENT



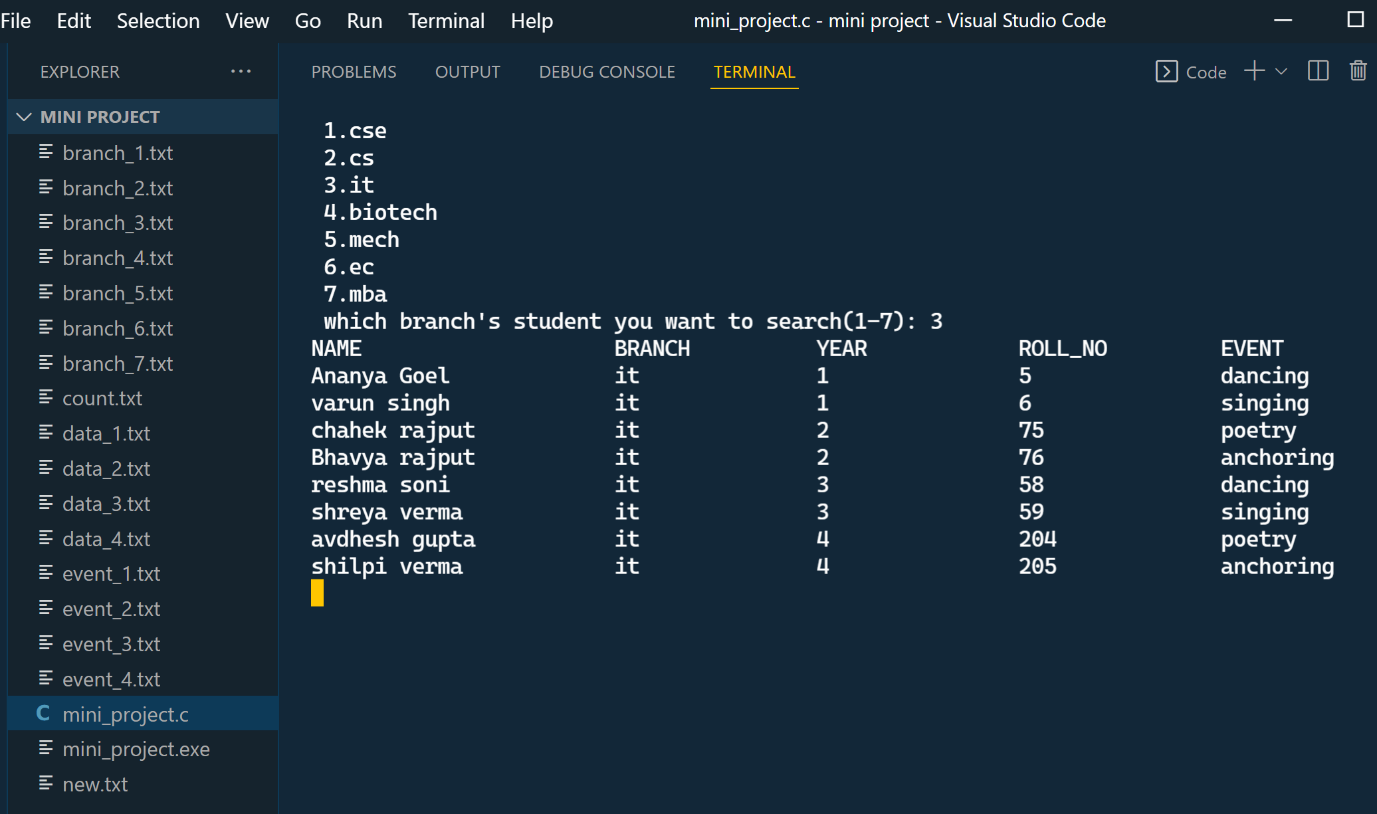
* NOW LET’S SEARCH BRANCH WISE. LET’S TRY CSE FIRST



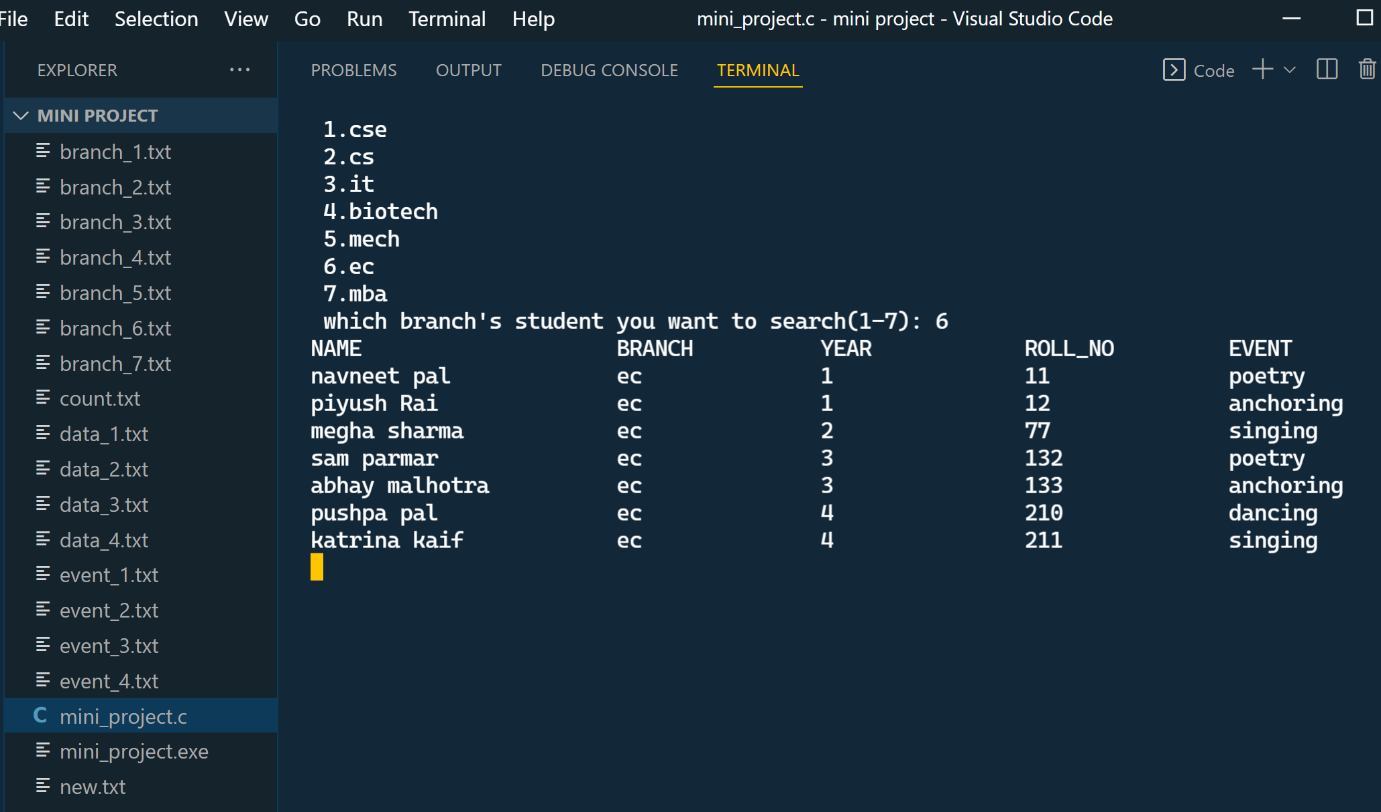
* NEXT WE SEARCH FOR CS BRANCH



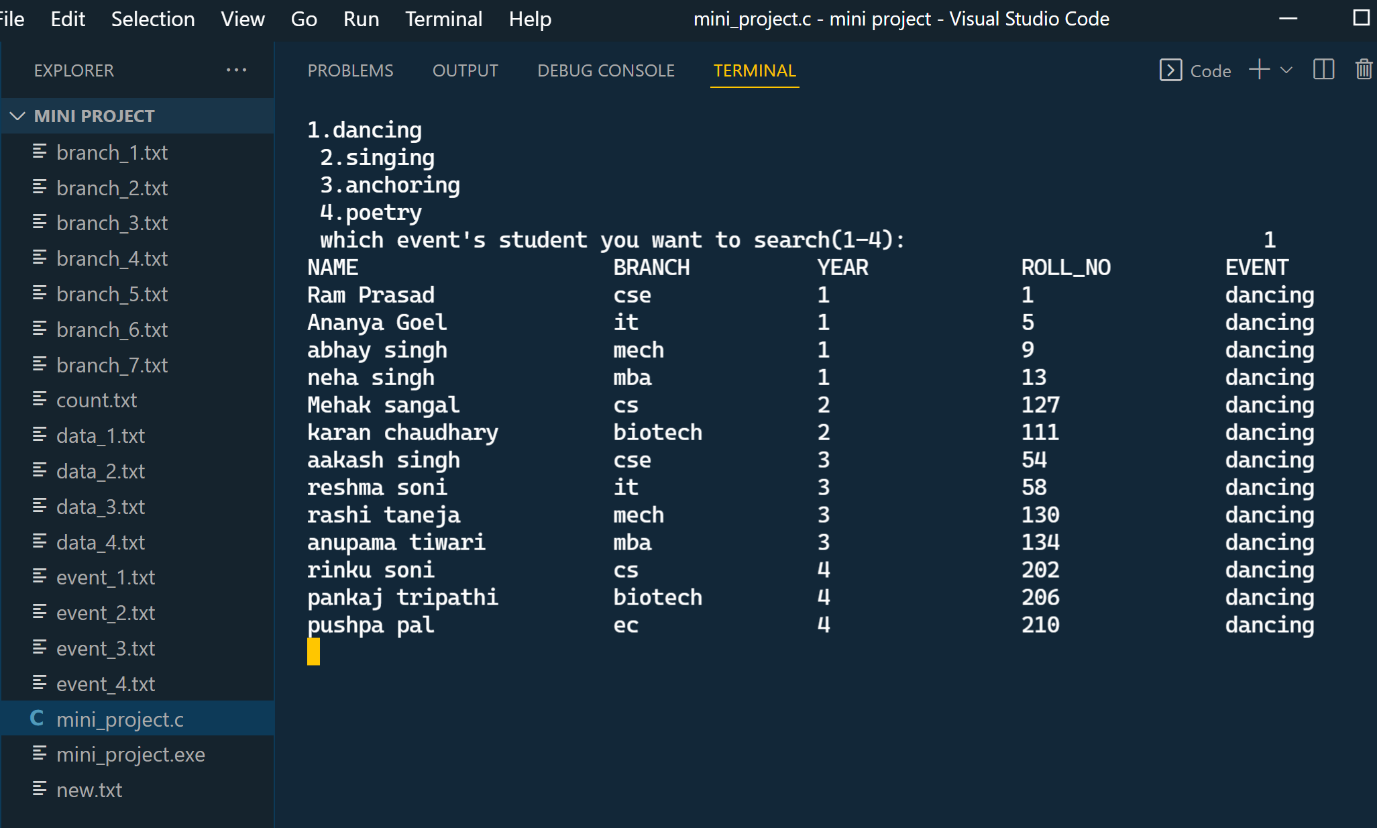
* NEXT, WE SEARCH FOR IT BRANCH

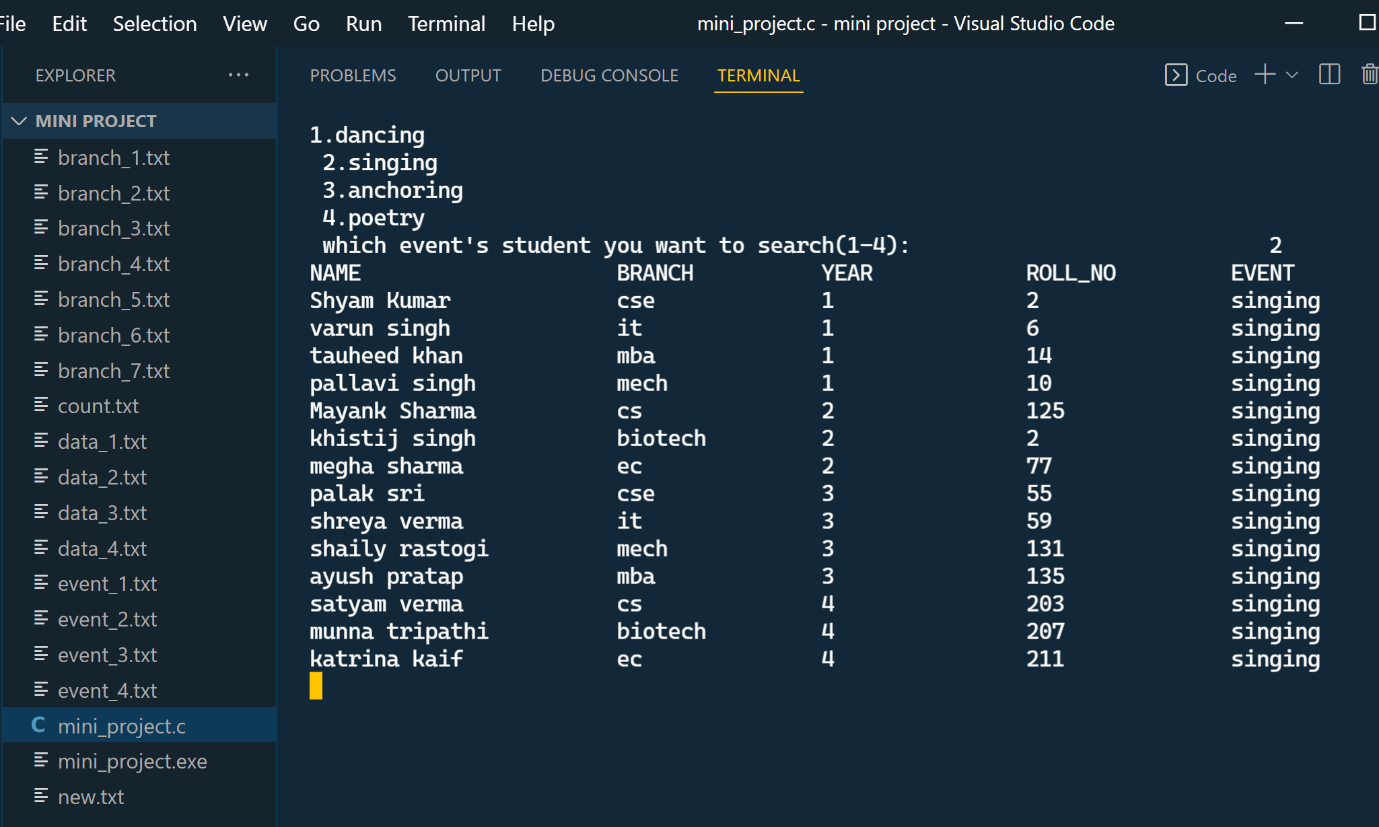


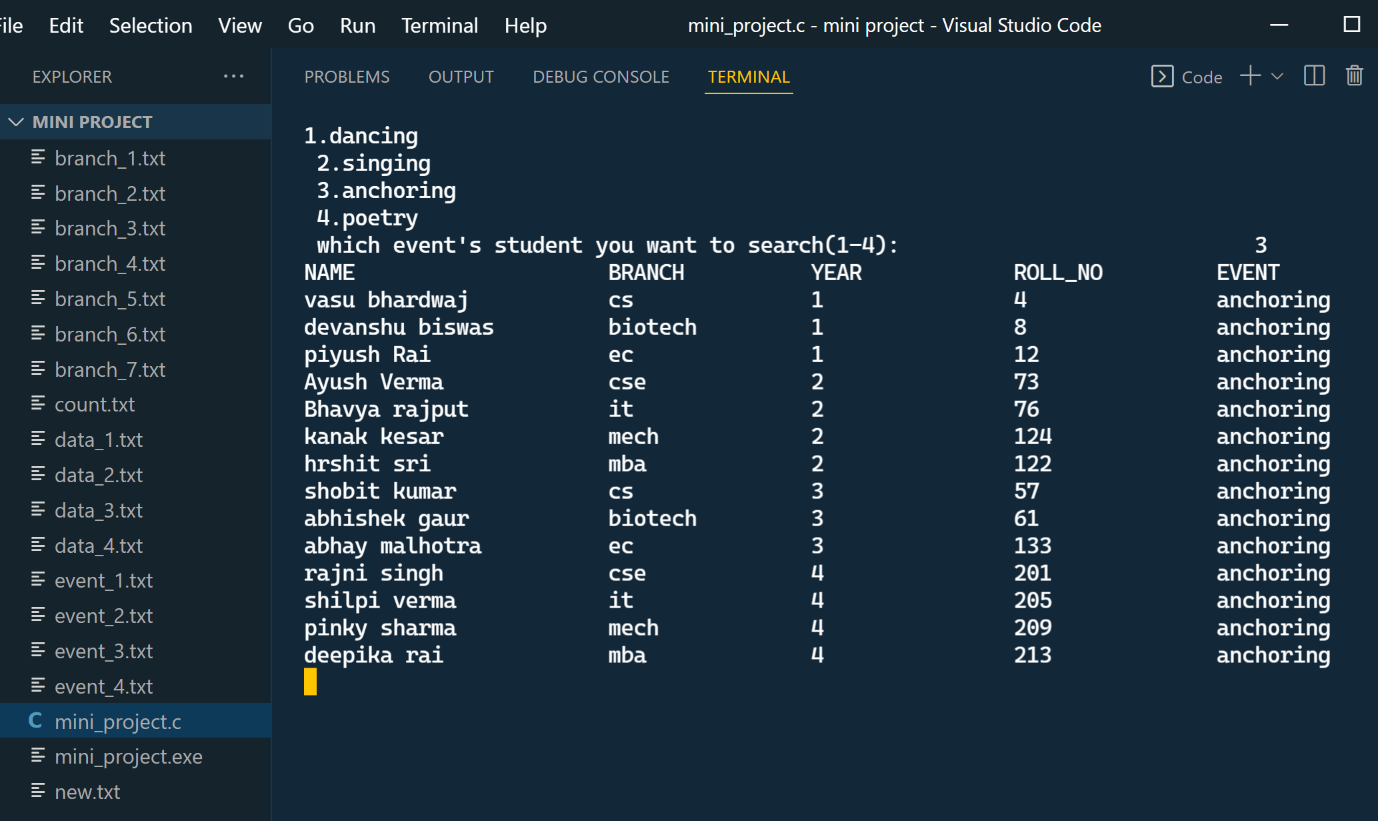
* LETS CHECK FOR EC ACCOUNT



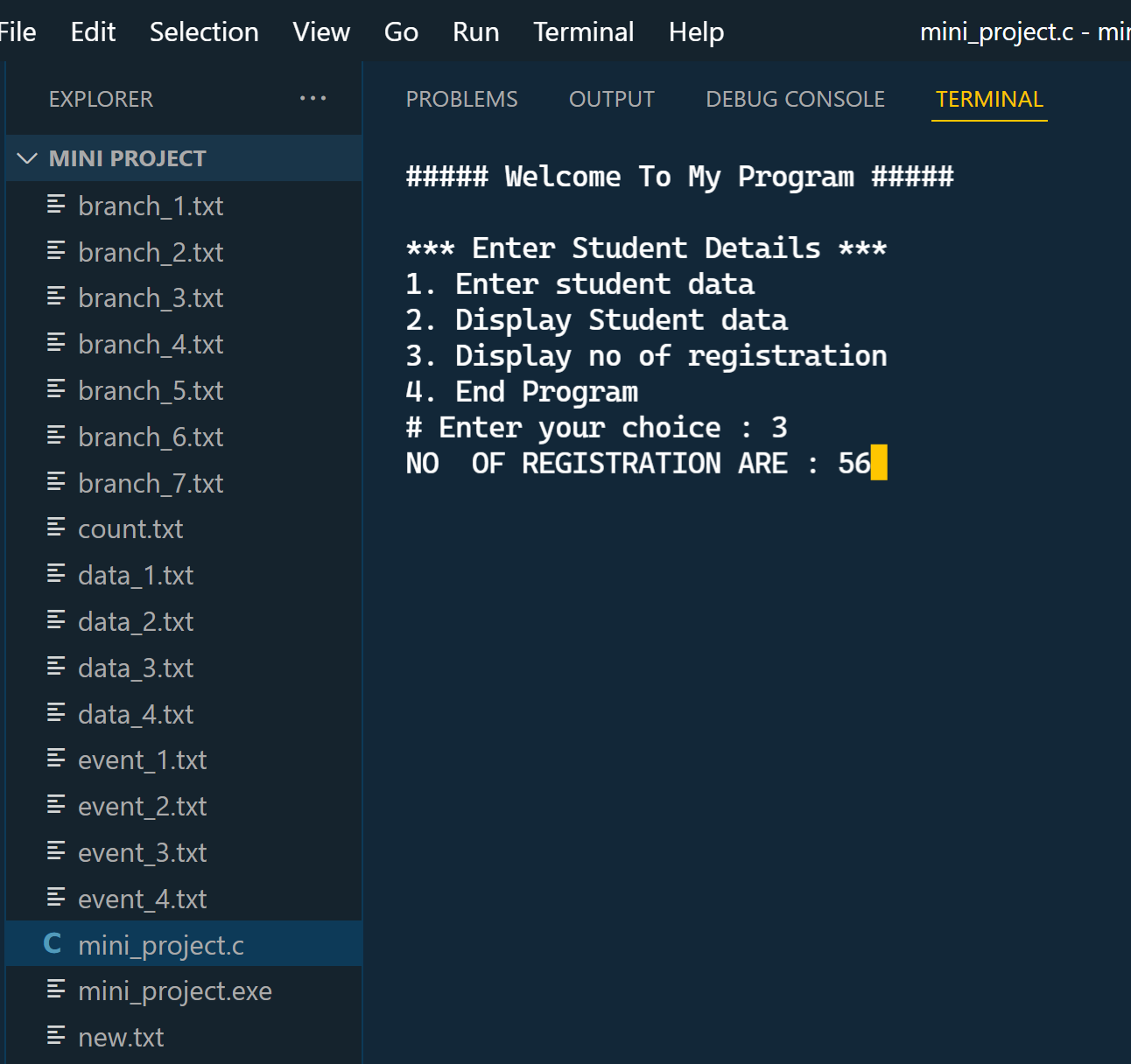
* LET’S TRY SORTING BY EVENTS (DANCING)



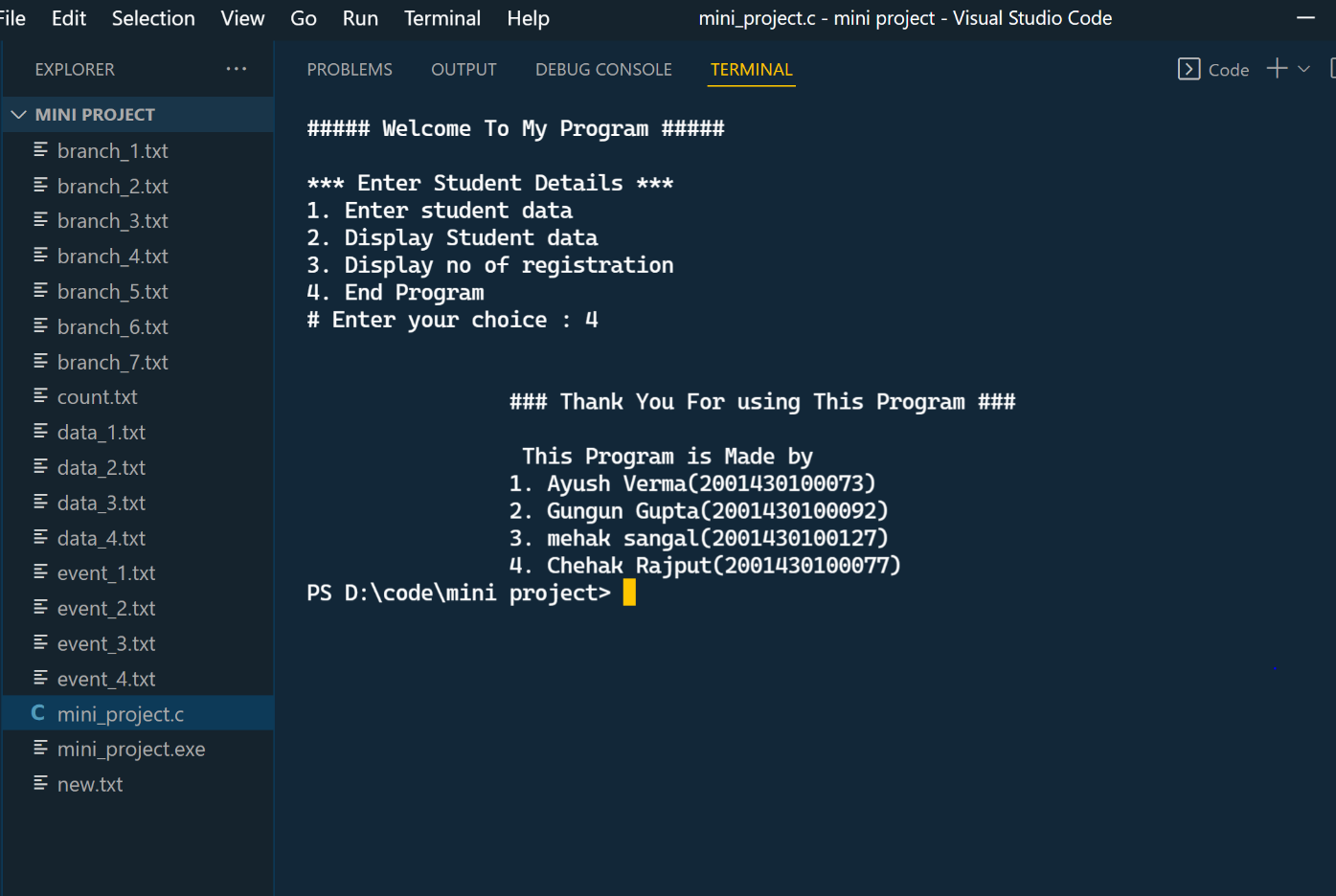
* NOW, WE SEARCH FOR THE PARICIPATION IN SINGING
* AT LAST LET’S SEARCH FOR ANCHORING PARTICIPANTS



* TO COUNT THE NUMBER OF REGISTRATIONS JUST PRESS 3 IN THE MENU.



* FINALLY, PRESS 4 IN THE MENU IF YOU WANT TO EXIT.



CONCLUSION

We successfully made our program to register and sort the participants using C programming language. We learnt how to sort and organize data in an organized way. We learned a better way to use structures and implementing them. C language is now clearer. We got a better vision towards it now. Learning C language was indeed fun!

We used many string functions as our program included string comparison and then its extraction. It all happened because our organization and our generous teachers gave us the responsibility to do so.

The program is successfully running and output is also correct. IMSEC registration hub is hence completed under proper guidance and surplus efforts.