STUDENT MERIT LIST

PROJECT MEMBERS

S.No. Roll No. Name Branch

1. 23P81A6746 S. Sowmya CSD

2. 23P81A6743 P. Jahnavi CSD

3. 23P81A6747 S. Akshitha CSD

4. 23P81A6701 A. Bharghav CSD

5. 23P81A6736 M. Karthikeya CSD

6. 23P81A6737 Sri Charan CSD

REQUIREMENT SPECIFICATION:

Student marks: Save marks of maximum 10 students along with important details in a binary file

Requirements:

1. Subject-wise merit list
2. List of all students
3. For a given student give his details along with marks and merit position

SOFTWARE REQUIREMENT:

Windows os, DEV C++

PROCEDURAL PROGRAMMING:

1. The structure used is struct student which contains of three elements namely
2. rollno – It’s datatype is integer, this is used to take the roll number of the given student
3. name – It is a character array of size 30 which is used to get the name of the student
4. marks – It is a array of integer datatype and size 5 which is used to get the marks of 5 subjects of the student
5. A variable ‘s1’ of struct student datatype is also declared
6. There are 8 user defined functions used in the program
7. main() – main function is used to print the menu and call different functions to perform some instructions.
8. int update\_std\_details() – This user-defined function used to add more records to the binary file
9. int compare(int) – This user-defined function is used to check whether the given roll number already exists or not.
10. void read\_std() – This user-defined function helps us to display the student details
11. void std\_list() – To display the student list and get the information of given student
12. void merit\_position(int) – To give the merit position of given the student
13. void merit\_list() – To display the list of subjects
14. void sorting(int) – To sort the marks and roll numbers of the students and display the sorted list of student details of respective subject
15. Two user-defined Header files namely meritlist1.h and meritlist2.h are used in the program
16. meritlist1.h – This header file consists of 4 user defined functions namely in

int update\_std\_details();

read\_std();

std\_list();

merit\_list();

1. meritlist2.h – This header file consists of 3 user-defined functions namely

int compare(int);

merit\_position(int);

sorting(int);

ALGORITHMS

grp\_project.c FILE:

ALGORITHM (int main() function):

1. START
2. Declare the variables ch,i,j,a of integer datatype
3. Declare a file pointer fp
4. Open Std\_merit\_list.bin named binary file in reading mode
5. Place the pointer at the initial character of the binary file
6. If fp = NULL go to 7 else 9
7. Print “Cannot open the file”
8. Go to 42
9. Close the file
10. Print “MENU”
11. Print “1. Update student details”
12. Print “2. Display student details”
13. Print “3. Show student list”
14. Print “4. Merit list”
15. Print “5. Exit”
16. Print “Enter your choice”
17. Read the choice ‘ch’ from the user
18. Clear the screen
19. If ch=1 go to 20 else 24
20. Go to update\_std\_details() function in the meritlist1.h header file
21. Assign the returned integer value from the function update\_std\_details() to a
22. If a=1 go to 20 else 23
23. Go to 41
24. If ch = 2 go to 25 else 28
25. Print “STUDENT DETAILS”
26. Go to read\_std() function in the meritlist1.h header file
27. Go to 41
28. If ch = 3 go to 29 else 31
29. Go to std\_list() function in the meritlist1().h header file
30. Go to 41
31. If ch = 4 go to 32 else 34
32. Go to merit\_list() function in meritlist1().h header file
33. Go to 41
34. If ch = 5 go to 35 else 37
35. Print “EXITING”
36. Go to 42
37. Print “Invalid input. Please, Re-enter!”
38. Print “Press any key to exit!”
39. Get any character from the user
40. Clear the screen
41. Go to 10
42. STOP

meritlist1.h FILE:

ALGORITHM (int update\_std\_details()):

1. START
2. Declare i,k,m variables of integer datatypes and initialise i=0
3. Declare a file pointer ‘ptr’
4. Open the file ‘std\_merit\_list.bin’ in read mode
5. Assign the address of first character of the file to ‘ptr’
6. Start reading the structure at which the ptr pointer is pointing using fread() function
7. If the fread() function returns 1 go to 8 else 11
8. i = i+1
9. Move the pointer to next structure in binary fine
10. Go to 6
11. Close the file
12. If i = 10 go to 13 else 15
13. Print “You have reached maximum limit!”
14. Return 2 to the main function and return control to main()
15. Print “Enter details of student”
16. Print “Roll no.”
17. Read the input ‘k’ from the user
18. Call int compare(int) function passing k as actual parameter from meritlist2.h
19. Assign the returned integer value of int compare(int) function into i variable
20. If i != 0 go to 16 else 21
21. Store the k value into rollno element of s1 variable of struct student type defined in meritlist2.h
22. Get the name of student from user
23. Store the name into name element of s1 variable of struct student type defined in meritlist2.h
24. Print “Enter subject-wise marks”
25. Set i=0
26. If i<5 go to 27 else 49
27. If i=0 go to 28 else 30
28. Print “Engineering Mathematics”
29. Go to 41
30. If i=1 go to 31 else 33
31. Print “Basic electrical engineering mathematics”
32. Go to 41
33. If i=2 go to 34 else 36
34. Print “Engineering chemistry”
35. Go to 41
36. If i=3 go to 37 else 39
37. Print “Programming for Problem Solving”
38. Go to 41
39. If i=4 go to 40
40. Print “Engineering Graphics”
41. Read marks ‘m’ from the user
42. If m>=0 and m<=100 go to 43 else 45
43. Assign the m to marks[i] element of s1 variable of struct student type
44. Go to 47
45. Print “Invalid marks! Please, re-renter the marks”
46. i = i – 1
47. i = i+1
48. Go to 26
49. Open the Std\_merit\_list.bin file in read and write mode
50. Place the pointer at the first character of the file
51. Seek the end of the file and point the pointer at end character of file using fseek() function
52. Write the entire structure data stored in s1 into the file
53. Close the file
54. Print “Do you want to create another record? (yes=1/no=0)”
55. Read the user choice and store it into variable i
56. Clear the screen
57. Return i to main() function and return control to main()

ALGORITHM (void read\_std()):

1. START
2. Declare a variable i of integer datatype
3. Open Std\_merit\_list.bin file using read mode
4. Assign the address of first character of binary to the pointer “ptr1”
5. Start reading the structure at which the ptr1 pointer is pointing using fread() function
6. If the fread() function returns 1 go to 7 else 29
7. Print the value at rollno element of structure
8. Print the string at name element of structure
9. Set i=0
10. If i<5 go to 11 else 27
11. If i=0 go to 12 else 14
12. Print the value at marks[i] element of structure
13. Go to 25
14. If i=1 go to 15 else 17
15. Print the value at marks[i] element of structure
16. Go to 25
17. If i=2 go to 18 else 20
18. Print the value at marks[i] element of structure
19. Go to 25
20. If i=3 go to 21 else 23
21. Print the value at marks[i] element of structure
22. Go to 25
23. If i=4 go to 24
24. Print the value at marks[i] element of structure
25. i = i + 1
26. Go to 10
27. Advance the pointer to next structure in binary fine
28. Go to 5
29. Close the file
30. Print “Press any key to exit”
31. Get a character from the user
32. Clear the screen
33. Return control to main function

ALGORITHM (void std\_list())

1. START
2. Declare op,ch variables of integer datatype
3. Declare a file pointer ‘ptr1’
4. Declare i and k variables of integer datatype and initialize i =1
5. Open the file Std\_merit\_list.bin in reading mode
6. Assign the address of first character of binary to the pointer “ptr1”
7. Print “List of students”
8. Start reading the structure at which the ptr1 pointer is pointing using fread() function
9. If the fread() function returns 1 go to 10 else 15
10. Print ‘i’ value and value of rollno in that structure
11. Print name string in the same structure
12. i = i + 1
13. Advance the pointer to next structure in binary fine
14. Go to 8
15. Print “Enter the s.no. of the student whose details you wanna access”
16. Read the user choice and store it into op variable
17. Clear the screen
18. If op>(i-1) go to 19 else 24
19. Print “Error! Please try again!”
20. Print “Press any key to exit”
21. Get a character from the user
22. Clear the screen
23. Go to 4
24. Rewind the pointer to the initial character of the binary file
25. Seek the structure at (op-1) position of the binary file and place the pointer at the starting position of the structure
26. Start reading the structure at which the ptr1 pointer is pointing using fread() function
27. Print “Student details”
28. Print values at rollno, name, marks[0], marks[1],marks[2],marks[3],marks[4] in the structure
29. Assign the value of rollno to the variable k
30. Close the file
31. Call the merit\_position(int) function passing k as the parameter
32. Print “Do you want to access another student data?(yes=1/no=0)”
33. Read the choice from user and assign that value into ch variable
34. Clear the screen
35. If ch = 0 go to 35 else 37
36. Return control to main() function
37. Go to 4

ALGORITHM (void merit\_list()):

1. START
2. Declare op,ch variables of integer datatype
3. Print “List of subjects”
4. Print “1. Engineering Mathematics”
5. Print “2. Basic Electrical Engineering”
6. Print “3. Engineering chemistry”
7. Print “4. Programming for problem solving”
8. Print “5. Engineering graphics”
9. Print “Select the subject to view merit list: ”
10. Read the choice from the user and store it in op variable
11. Clear the screen
12. If op>0 and op<6 go to else
13. If op=1
14. Print “Mathematics merit list”
15. Go to
16. If op=2
17. Print “BEE merit list”
18. Go to
19. If op=3
20. Print “Chemistry merit list”
21. Go to
22. If op=4
23. Print “PPS merit list”
24. Go to
25. Print “EG merit list”
26. Call the sorting() function passing op as a parameter
27. Go to
28. Clear the screen
29. Print “The choice is not included in the menu”
30. Print “Do you want to view merit list of another subject?(yes=1/no=0)”
31. Read the choice from user and assign the given value into ch variable
32. Clear the screen
33. If ch = 0 go to else
34. Return the control to main() function
35. Go to 3

meritlist2.h FILE:

ALGORITHM (int compare(int)):

1. START
2. Take the parameter from update\_std\_details() and copy the value to c variable of int
3. Declare flag,j,i variables of integer datatype and initialize flag and j to 0
4. Declare an array temp of size 10 and integer datatype
5. Declare a file pointer ‘ptr’
6. Open the file Std\_merit\_list.bin in reading mode
7. Assign the address of first character of binary file to ptr
8. Start reading the structure at which the ptr pointer is pointing using fread() function
9. If the fread() function returns 1 go to 10 else 14
10. Assign the rollno in the structure to temp[j]
11. j = j+1
12. Advance the pointer to next structure
13. Go to 9
14. Set i=0
15. If i<j go to 16 else 21
16. If temp[i] = c go to 17 else 19
17. Flag = 1
18. Go to 21
19. i = i+1
20. Go to
21. Close the file
22. If flag = 1 go to 23 else 25
23. Print “Entry of the roll number already exists. Please, Re-enter!”
24. Return 1 to update\_std\_details() and return control to main()
25. Return 0 to update\_std\_details() and return control to main()

ALGORITHM (void merit\_position(int)):

1. START
2. Declare i, j, k, temp variables of integer datatype and initialize j=0 and store the copy of parameter from std\_list() into variable c.
3. Declare id, total arrays of integer datatype of size 10
4. Declare a file pointer ‘fp’
5. Open the file Std\_merit\_list.bin in reading mode
6. Assign the address of first character of binary file to fp
7. Start reading the structure at which the fs pointer is pointing using fread() function
8. If the fread() function returns 1 go to 9 else 20
9. Set temp = 0
10. Assign the value of rollno in the structure to id[j]
11. Set i=0
12. If i<5 go to 13 else 16
13. temp = temp + marks[i] (value of marks at the given location of structure)
14. i = i+1
15. Go to 12
16. Assign temp to total[j]
17. j = j+1
18. Advance the pointer to next structure location
19. Go to 7
20. Set i =0
21. If i<(j-1) go to 22 else 35
22. Set k = i+1
23. If k<j go to 24 else 33
24. If total[i] < total[k] go to 25 else 31
25. temp = total[i]
26. total[i] = total[k]
27. total[k] = temp
28. temp = id[i]
29. Id[i] = id[k]
30. Id[k] = temp
31. k = k+1
32. Go to 23
33. i = i+1
34. Go to 21
35. Set i=0
36. If i<j go to 37 else 41
37. If id[i] = c
38. Go to 41
39. i = i+1
40. Go to 36
41. Close the file
42. Print “Overall merit position of student (i+1)”
43. Return control to std\_list() function

ALGORITHM (void sorting(int)):

1. START
2. Take the parameter from merit\_list() function and copy the value into variable op of int datatype
3. Declare i, j, k, a, temp variables of integer datatype and initialize i=0
4. Declare a file pointer ‘fs’
5. Declare arr, id arrays of integer datatype and size 10
6. Declare a pointer array \*std\_name of char datatype and size 1
7. Open Std\_merit\_list.bin file in read mode
8. Assign the address of first character of the binary file to the pointer fs
9. Start reading the structure at which the fs pointer is pointing using fread() function
10. If the fread() function returns 1 go to 11 else 16
11. Assign value at rollno in the structure to id[i]
12. Assign value at marks[op-1] to arr[i]
13. i = i+1
14. Advance the pointer to next structure location
15. Go to 9
16. Set j=0
17. If j<i-1 go to 18 else 31
18. Set k = j+1
19. If k<i go to 20 else 29
20. If arr[j] < arr[k] go to 21 else 27
21. temp = arr[j];
22. arr[j] = arr[k];
23. arr[k] = temp;
24. temp = id[j];
25. id[j] = id[k];
26. id[k] = temp;
27. K = k+1
28. Go to 19
29. j = j+1
30. Go to 17
31. Declare a pointer ‘t’ of char datatype and allocated 30 bytes of memory for it using memory allocation
32. Set j=0
33. If j<i go to 34 else 47
34. Rewind the pointer to point at the first character of the binary file
35. Start reading the structure at which the fs pointer is pointing using fread() function
36. If the fread() function returns 1 go to 37 else 43
37. Assign rollno value in the structure to ‘a’
38. If id[j] = a go to 39 else 41
39. Assign the base address of name string of structure to pointer ‘t’
40. Go to 43
41. Advance the file pointer to next structure
42. Go to 35
43. Store the address in pointer into pointer array std\_name[1]
44. Print ‘j+1’, ‘std\_name[1]’, id[j], arr[j]
45. j = j+1
46. Go to 33
47. Close the file
48. Return the control to merit\_list()

CODE

1. grp\_project.c FILE
2. meritlist1.h FILE
3. meritlist2.h FILE

grp\_project.c FILE:

#include<stdio.h>

#include<windows.h>

#include"meritlist1.h"

int main()

{

int ch,i,j,a;

FILE \*fp;

fp = fopen("Std\_merit\_list.bin","rb");

if(fp==NULL)

{

printf("Cannot open the file!");

return 1;

}

fclose(fp);

do{

printf("\n\n\n\n\n%-61s"," ");

for(i=0;i<30;i++)

printf("-");

printf("\n%-74sMENU\n"," ");

printf("%-61s"," ");

for(i=0;i<30;i++)

printf("-");

printf("\n\n%-62s1.Update student details"," ");

printf("\n\n%-62s2.Display all student details"," ");

printf("\n\n%-62s3.Show student list"," ");

printf("\n\n%-62s4.Merit list"," ");

printf("\n\n%-62s5.Exit"," ");

printf("\n\n%-62sEnter your choice: "," ");

scanf("%d",&ch);

system("cls");

switch(ch)

{

case 1: do{

a=update\_std\_details();

}while(a==1);

break;

case 2: printf("\n\n\n\n");

printf("%-45s"," ");

for(i=0;i<50;i++)

printf("-");

printf("\n%-62sSTUDENT DETAILS: \n"," ");

printf("%-45s"," ");

for(i=0;i<50;i++)

printf("-");

printf("\n");

read\_std();

break;

case 3: std\_list();

break;

case 4: merit\_list();

break;

case 5: printf("EXITING.........");

return 1;

default: printf("\n\n\n%-45sInvalid input. Please, Re-enter!\n"," ");

printf("\n\n%-45sPress any key to exit!"," ");

getch();

system("cls");

}

}while(1);

return 0;

}

meritlist1.h HEADER FILE:

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include"meritlist2.h"

int update\_std\_details()

{

int i=0,m,k,count=0;

FILE \*ptr;

ptr = fopen("Std\_merit\_list.bin","rb");

while(fread(&s1,sizeof(s1),1,ptr)==1)

{

i++;

}

fclose(ptr);

if(i>=10)

{

printf("\n%-62sYou have reached maximum limit!\n"," ");

return 2;

}

printf("\n%-30sEnter the details of student:\n"," ");

do{

printf("\n%-30sRoll no: "," ");

scanf("%d",&k);

i = compare(k);

}while(i!=0);

s1.rollno = k;

printf("\n%-30sName: "," ");

fflush(stdin);

gets(s1.name);

printf("\n%-30sEnter subject-wise marks:\n"," ");

for(i=0;i<5;i++)

{

if(i==0)

printf("\n%-30sEngineering Mathematics: "," ");

else if(i==1)

printf("\n%-30sBasic Electrical Engineering: "," ");

else if(i==2)

printf("\n%-30sEngineering Chemistry: "," ");

else if(i==3)

printf("\n%-30sProgramming for Problem Solving: "," ");

else if(i==4)

printf("\n%-30sEngineering Graphics: "," ");

scanf("%d",&m);

if(m>=0&&m<=100)

s1.marks[i] = m;

else

{

printf("\n%-30sInvalid marks! Please, re-enter the marks\n"," ");

i--;

}

}

ptr = fopen("Std\_merit\_list.bin","rb+");

fseek(ptr,sizeof(s1)\*count,SEEK\_END);

fwrite(&s1,sizeof(s1),1,ptr);

fclose(ptr);

printf("\n%-30sRecord has been successfully added....."," ");

printf("\n\n%-30sDo you want to create another record?(yes=1,no=0) "," ");

scanf("%d",&i);

system("cls");

return i;

}

void read\_std()

{

int i;

FILE \*ptr1;

ptr1 = fopen("Std\_merit\_list.bin","rb");

printf("\n");

printf("\n\t\tRollno\t\tName\t\t\tMaths\t\tBEE\t\tChem\t\tPPS\t\tEG\n");

printf("\t\t\t\t\t\t\tMarks\t\tMarks\t\tMarks\t\tMarks\t\tMarks");

printf("\n\n");

while(fread(&s1,sizeof(s1),1,ptr1)==1)

{

printf("\t\t%d\t\t",s1.rollno);

printf("%s\t\t",s1.name);

for(i=0;i<5;i++)

{

if(i==0)

printf("%d\t\t",s1.marks[i]);

else if(i==1)

printf("%d\t\t",s1.marks[i]);

else if(i==2)

printf("%d\t\t",s1.marks[i]);

else if(i==3)

printf("%d\t\t",s1.marks[i]);

else if(i==4)

printf("%d\n",s1.marks[i]);

}

printf("\n");

}

fclose(ptr1);

printf("\n%-62sPress any key to exit!"," ");

getch();

system("cls");

}

void std\_list()

{

int op,ch;

FILE \*ptr1;

do{

int i=1,k,j;

ptr1 = fopen("Std\_merit\_list.bin","rb");

printf("\n\n%-32s"," ");

for(j=0;j<20;j++)

printf("-");

printf("\n%-34sList of students:\n"," ");

printf("%-32s"," ");

for(j=0;j<20;j++)

printf("-");

printf("\n\n%-32ss.no\tRollno\tName\n"," ");

while(fread(&s1,sizeof(s1),1,ptr1)==1)

{

printf("\n%-32s%d\t%d\t"," ",i,s1.rollno);

puts(s1.name);

i++;

}

printf("\n\n%-20sEnter the s.no. of the student whose details you wanna access: "," ");

scanf("%d",&op);

system("cls");

if(op>(i-1))

{

printf("\n\n%-40sError! Please try again!\n\n%-40sPress any key to exit!"," "," ");

getch();

system("cls");

continue;

}

rewind(ptr1);

fseek(ptr1,sizeof(s1)\*(op-1),SEEK\_SET);

fread(&s1,sizeof(s1),1,ptr1);

printf("\n%-32s"," ");

for(j=0;j<20;j++)

printf("-");

printf("\n%-34sStudent details:\n"," ");

printf("%-32s"," ");

for(j=0;j<20;j++)

printf("-");

printf("\n\n%-30sRoll no of the student: %d\n"," ",s1.rollno);

printf("\n\n%-30sName of the student: %s\n"," ",s1.name);

printf("\n\n%-30sEngineering Mathematics: %d\n"," ",s1.marks[0]);

printf("\n\n%-30sBasic Electrical Engineering: %d\n"," ",s1.marks[1]);

printf("\n\n%-30sEngineering Chemistry: %d\n"," ",s1.marks[2]);

printf("\n\n%-30sProgramming for Problem Solving: %d\n"," ",s1.marks[3]);

printf("\n\n%-30sEngineering Graphics: %d\n"," ",s1.marks[4]);

k = s1.rollno;

fclose(ptr1);

merit\_position(k);

printf("\n\n%-20sDo you want to access another student data?(yes=1/no=0) "," ");

scanf("%d",&ch);

system("cls");

if(ch==0)

break;

}while(1);

}

void merit\_list()

{

int op,ch,j;

while(1)

{

printf("\nList of subjects:\n");

printf("\n\n1.Engineering Mathematics");

printf("\n\n2.Basic Electrical Engineering");

printf("\n\n3.Engineering chemistry");

printf("\n\n4.Programming for problem solving");

printf("\n\n5.Engineering graphics\n\n");

printf("Select the subject to view merit list: ");

scanf("%d",&op);

system("cls");

if(op>0&&op<6)

{

if(op==1)

{

printf("\nMaths merit list:\n");

printf("\n\nMerit\t\tName\t\t\tRollno\t\tMath\n");

printf("Position\t\t\t\t\t\tMarks\n\n");

}

else if(op==2)

{

printf("\nBEE merit list:\n");

printf("\n\nMerit\t\tName\t\t\tRollno\t\tBEE\n");

printf("Position\t\t\t\t\t\tMarks\n\n");

}

else if(op==3)

{

printf("\nChemistry merit list:\n");

printf("\n\nMerit\t\tName\t\t\tRollno\t\tChem\n");

printf("Position\t\t\t\t\t\tMarks\n\n");

}

else if(op==4)

{

printf("\nPPS merit list:\n");

printf("\n\nMerit\t\tName\t\t\tRollno\t\tPPS\n");

printf("Position\t\t\t\t\t\tMarks\n\n"," ");

}

else

{

printf("\nEG merit list:\n");

printf("\n\nMerit\t\tName\t\t\tRollno\t\tEG\n");

printf("Position\t\t\t\t\t\tMarks\n\n");

}

sorting(op);

}

else

{

system("cls");

printf("\n\nThe choice is not included in the menu!\n\n");

}

printf("\n\nDo you want to view merit list of another subject?(yes=1/no=0) ");

scanf("%d",&ch);

system("cls");

if(ch==0)

break;

}

}

meritlist2.h HEADER FILE:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct student

{

int rollno;

char name[30];

int marks[5];

}s1;

int compare(int c)

{

int temp[10];

int flag=0,i;

int j=0;

FILE \*ptr1;

ptr1 = fopen("Std\_merit\_list.bin","rb");

while(fread(&s1,sizeof(s1),1,ptr1)==1)

{

temp[j]=s1.rollno;

j++;

}

for(i=0;i<j;i++)

{

if(temp[i] == c)

{

flag = 1;

break;

}

}

fclose(ptr1);

if(flag)

{

printf("\n%-30sEntry of this roll number already exists. Please, Re-enter!\n"," ");

return 1;

}

else

return 0;

}

void merit\_position(int c)

{

int i,j=0,k,temp,id[10],total[10];

FILE \*fp = fopen("Std\_merit\_list.bin","rb");

while(fread(&s1,sizeof(s1),1,fp)==1)

{

temp=0;

id[j] = s1.rollno;

for(i=0;i<5;i++)

temp = temp+s1.marks[i];

total[j] = temp;

j++;

}

for(i=0;i<(j-1);i++)

{

for(k=i+1;k<j;k++)

{

if(total[i]<total[k])

{

temp = total[i];

total[i] = total[k];

total[k] = temp;

temp = id[i];

id[i] = id[k];

id[k] = temp;

}

}

}

for(i=0;i<j;i++)

{

if(id[i] == c)

break;

}

fclose(fp);

printf("\n\n%-30sOverall merit position of student: %d\n"," ",(i+1));

}

void sorting(int op)

{

int i=0,j,k,a,temp,arr[10],id[10];

char \*std\_name[1];

FILE \*fs = fopen("Std\_merit\_list.bin","rb");

while(fread(&s1,sizeof(s1),1,fs)==1)

{

id[i] = s1.rollno;

arr[i] = s1.marks[(op-1)];

i++;

}

for(j=0;j<(i-1);j++)

{

for(k=j+1;k<i;k++)

{

if(arr[j]<arr[k])

{

temp = arr[j];

arr[j] = arr[k];

arr[k] = temp;

temp = id[j];

id[j] = id[k];

id[k] = temp;

}

}

}

char \*t = (char \*)malloc(30\*sizeof(char));

for(j=0;j<i;j++)

{

rewind(fs);

while(fread(&s1,sizeof(s1),1,fs)==1)

{

a = s1.rollno;

if(id[j] == a)

{

t = s1.name;

break;

}

}

std\_name[1] = t;

printf("\n%d\t\t%s\t\t%d\t\t%d\n",j+1,std\_name[1],id[j],arr[j]);

}

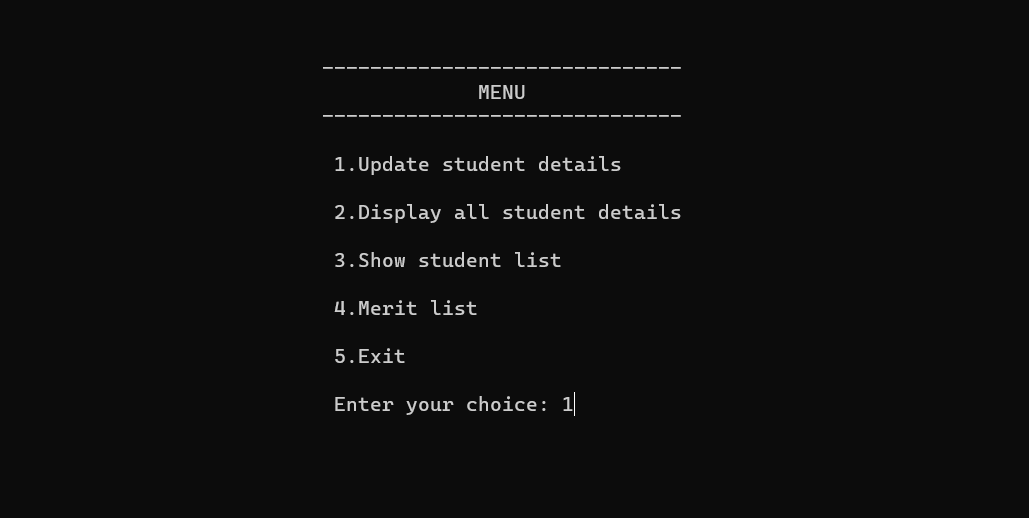
printf("\n");

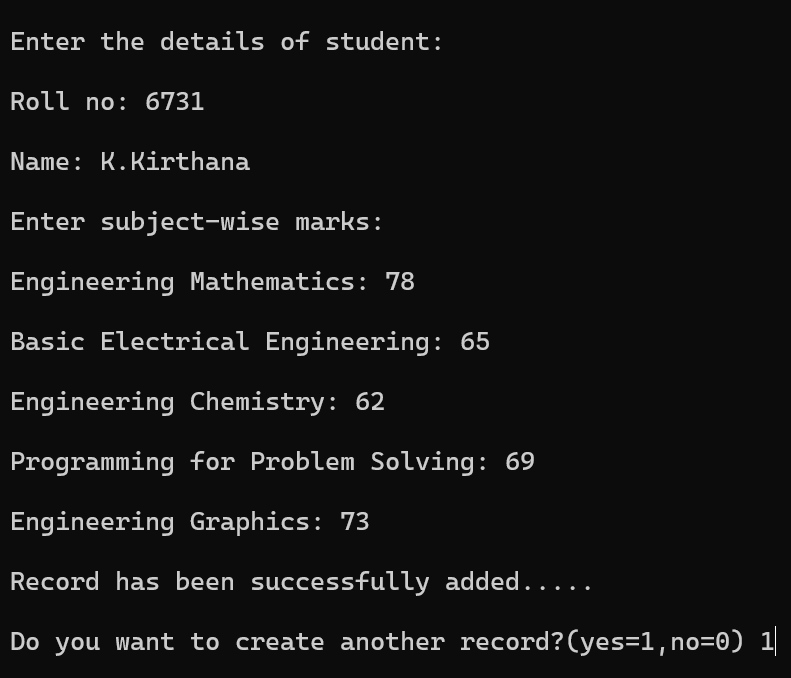
fclose(fs);

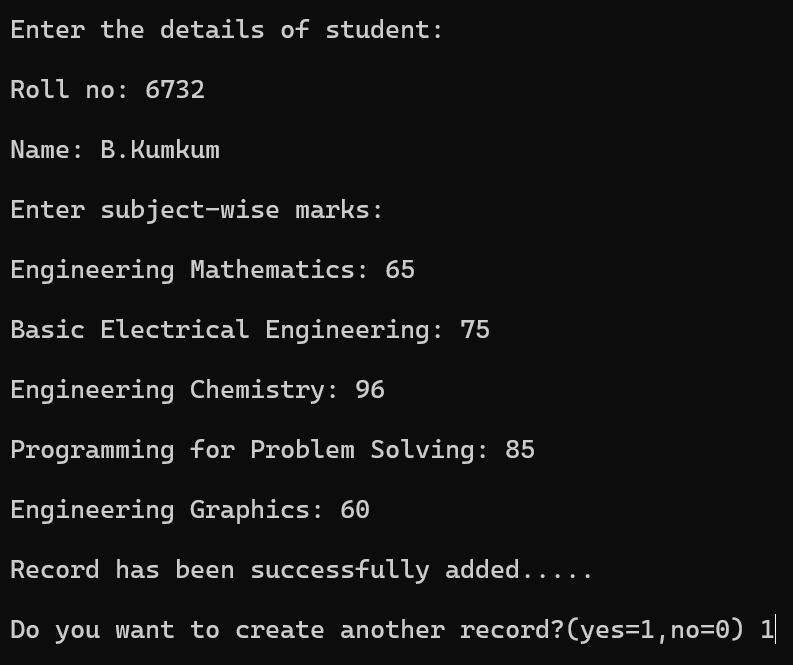
}

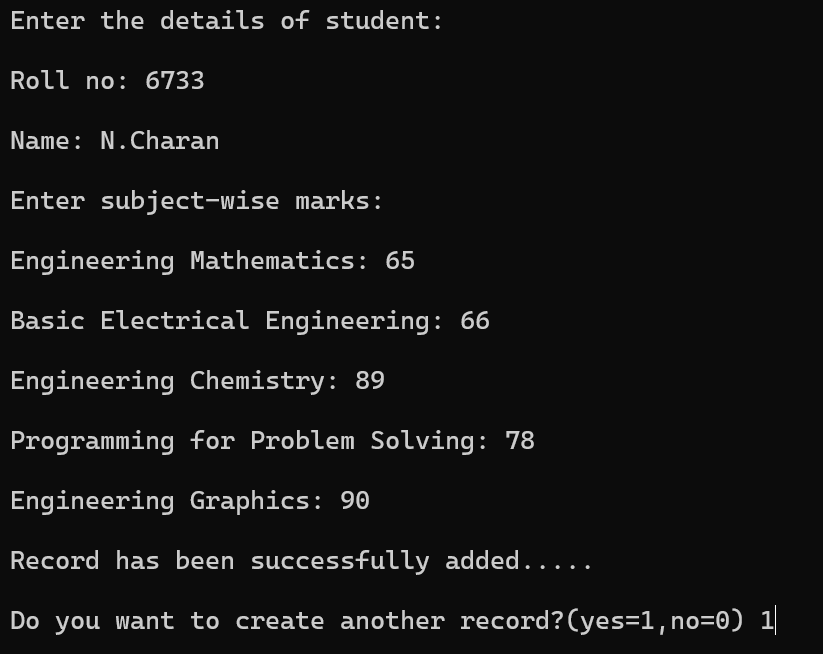
INPUT

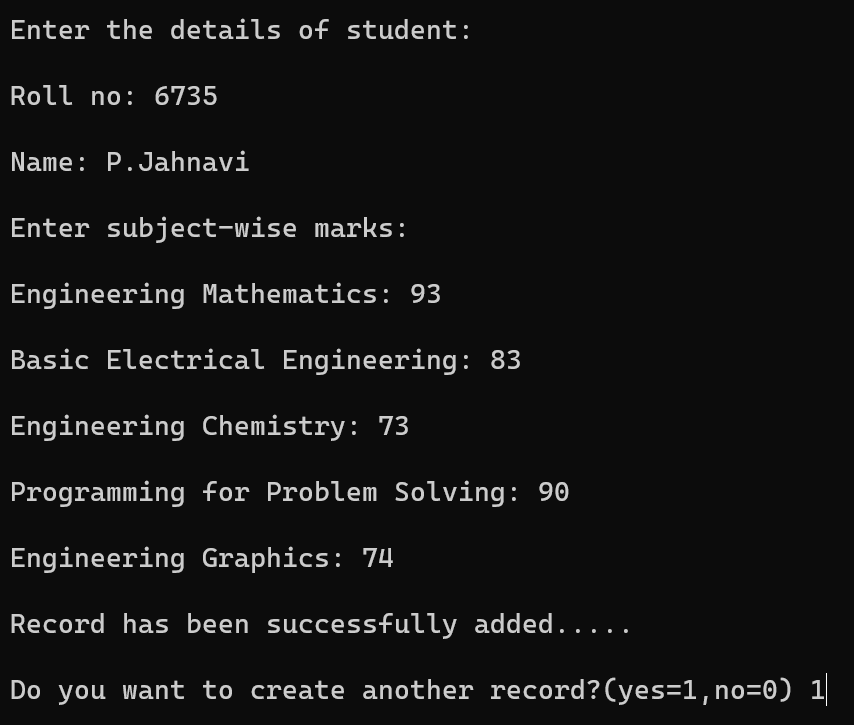
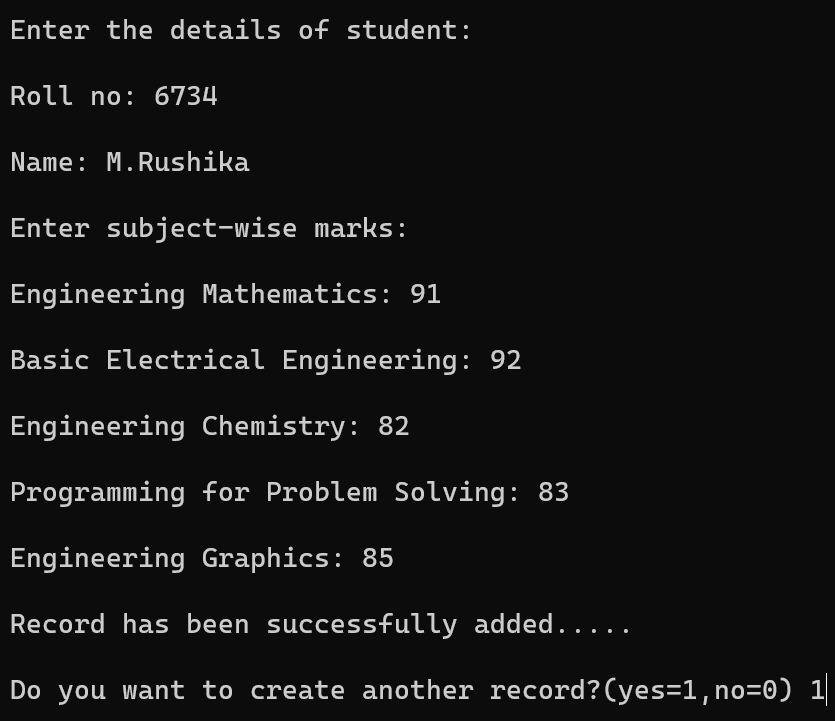
Updating the student details

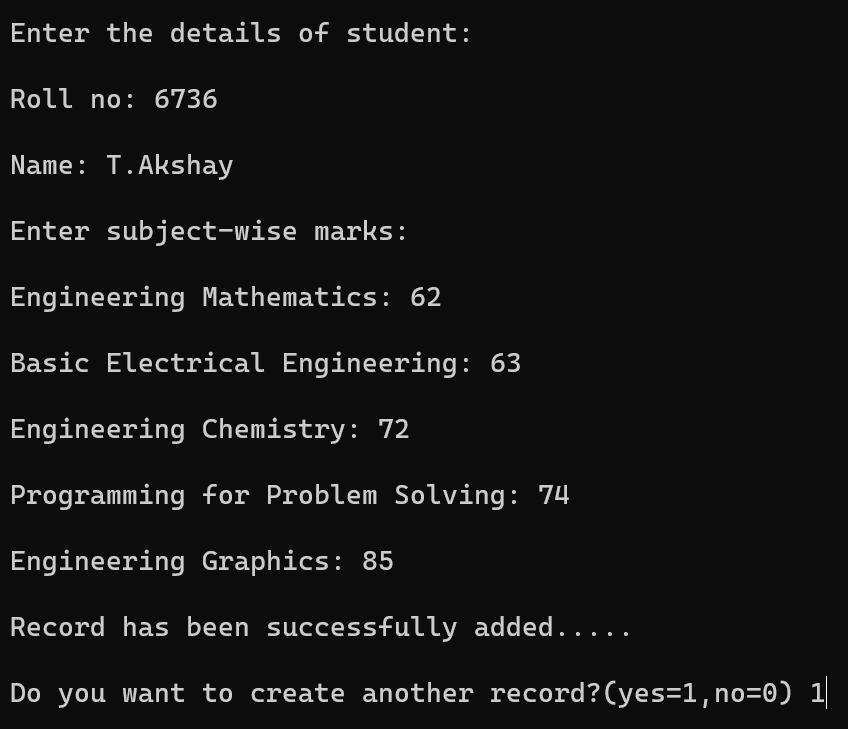


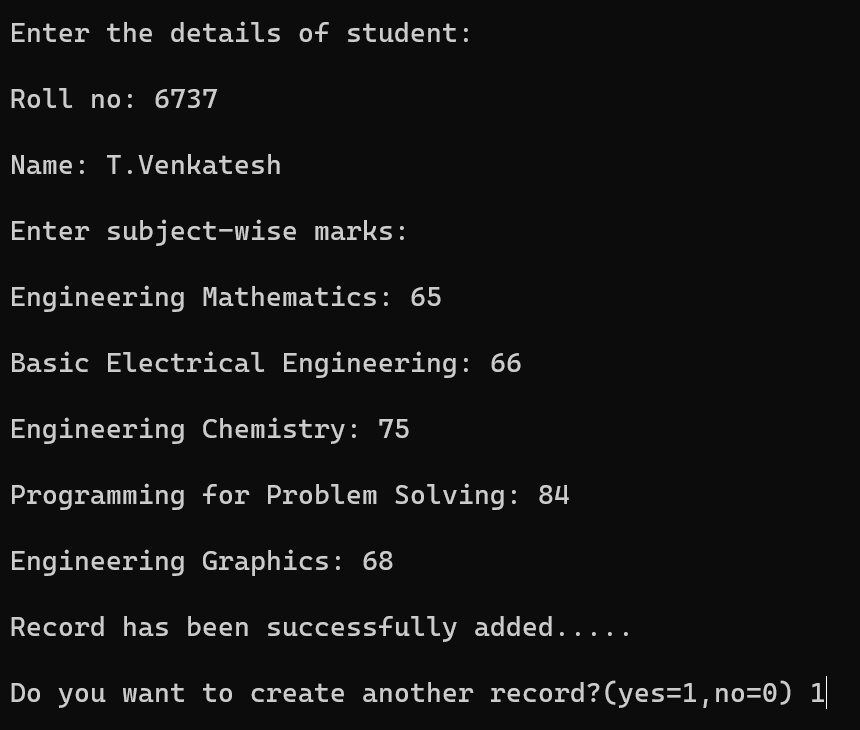


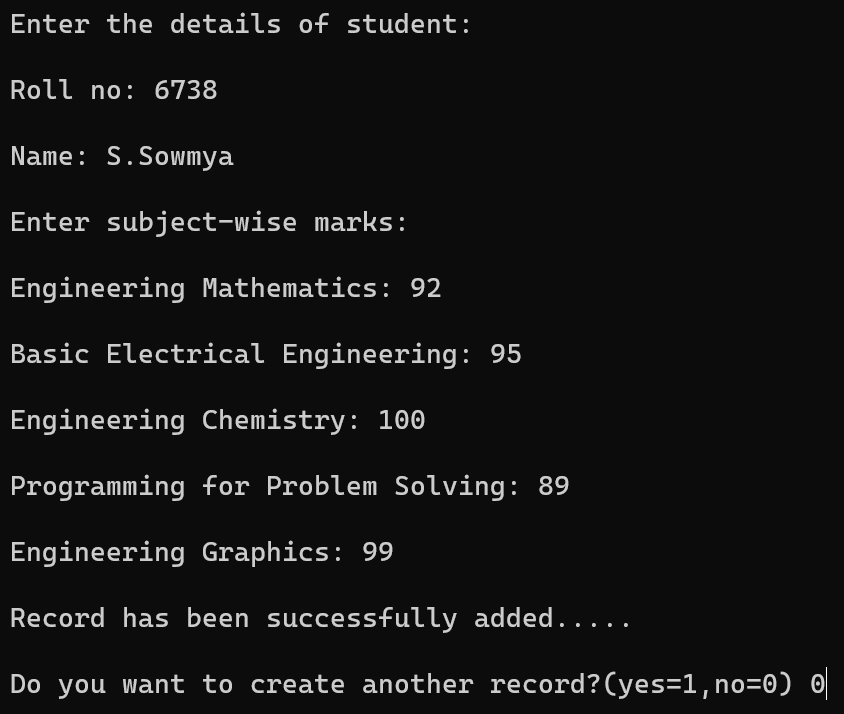






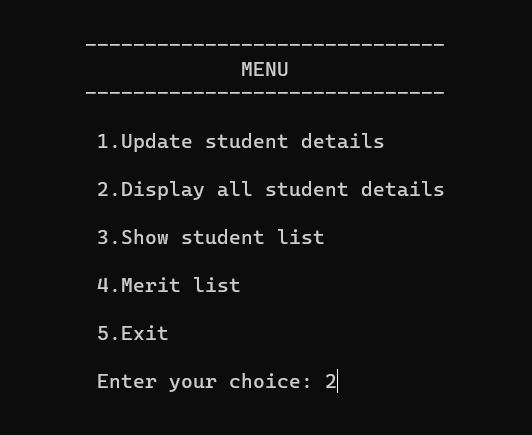


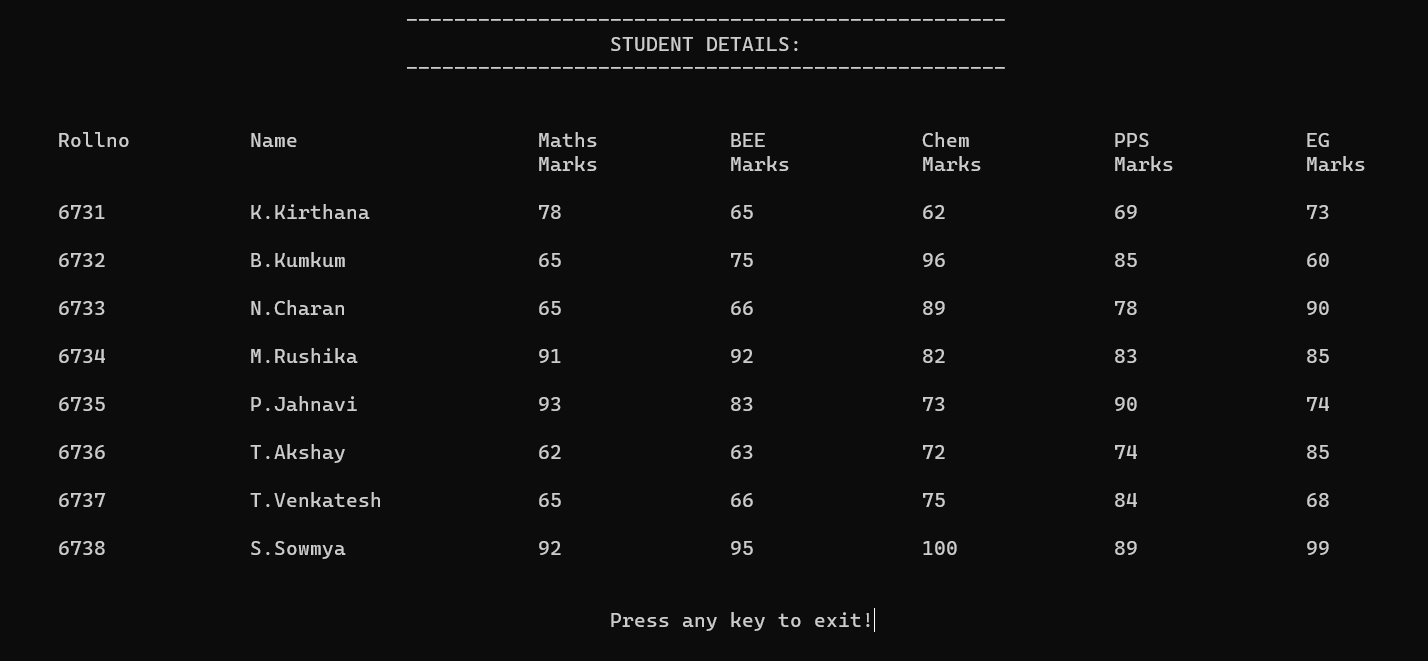




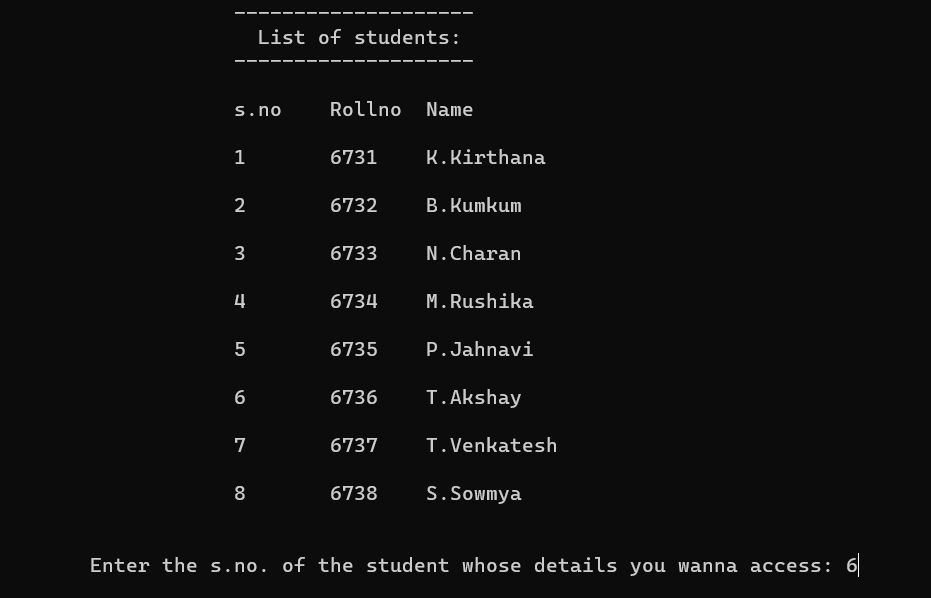
OUTPUT

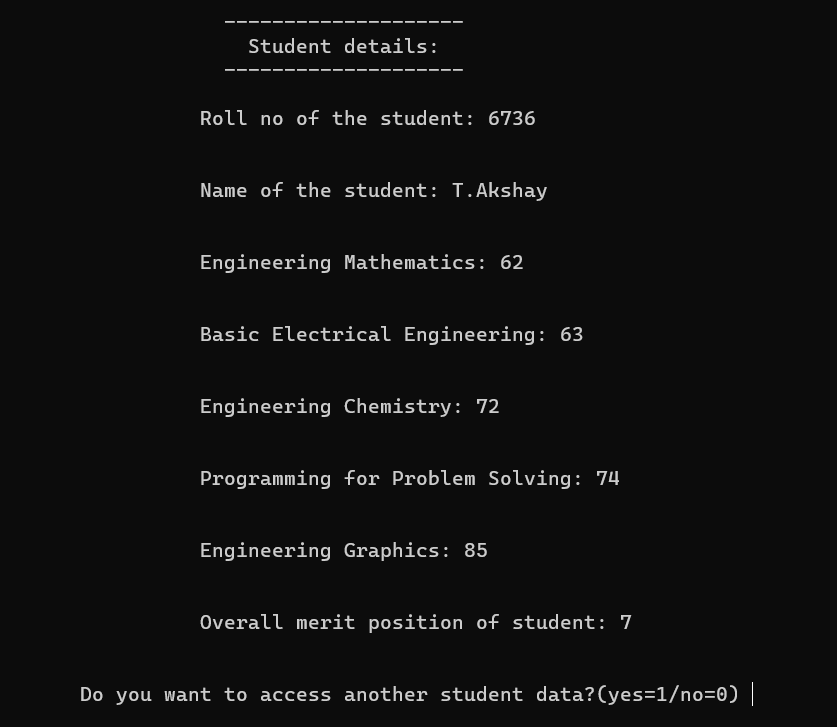
Displaying the student details



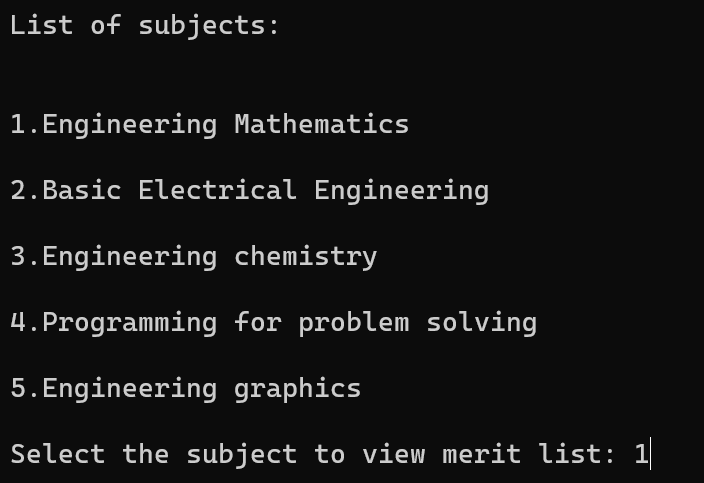


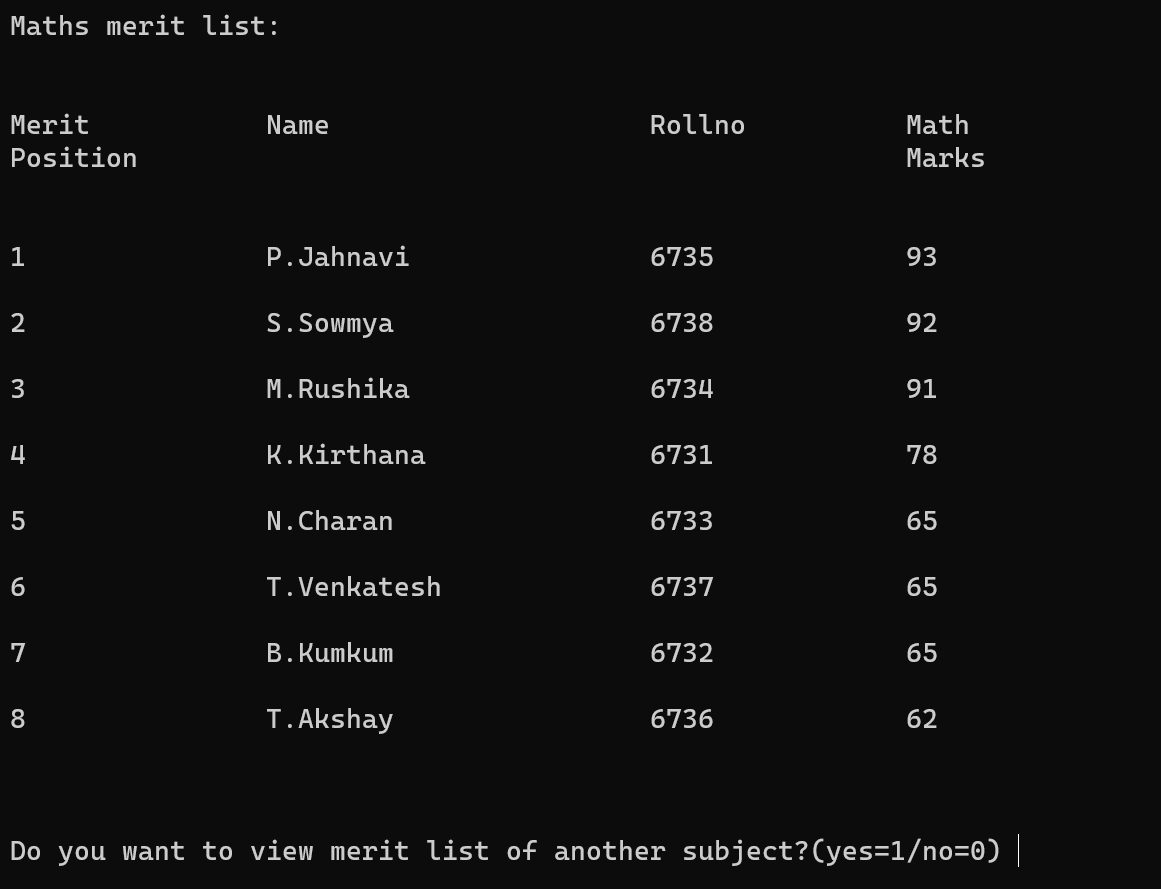
Showing the student list:



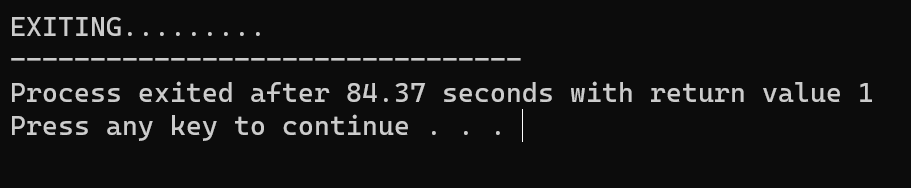


Merit list





EXITING



BINARY FILE

(Std\_merit\_list.bin FILE)

