**Task 1:**

#include<iostream>

#include<string>

using namespace std;

struct CourseGrade

{

string Name;

int Id;

int\* tests;

float Average;

char Grade;

};

void\* average(int Total\_Students, int Total\_tests, CourseGrade\* students);

void\* allocategrade(int Total\_Students, CourseGrade\* students);

int main()

{

int Total\_Students, Total\_tests;

cout << "Enter number of Students: ";

cin >> Total\_Students;

cout << "Enter number of Test Scores you want to add: ";

cin >> Total\_tests;

CourseGrade\* students = new CourseGrade[Total\_Students];

for (int i = 0; i < Total\_Students; i++)

{

students[i].tests = new int[Total\_tests];

}

cout << "\t\t\t\t\*Enter Data of Students\*" << endl;

for (int i = 0; i < Total\_Students; i++)

{

cout << endl << endl;

cout << "Enter Name of student " << i + 1 << ": ";

cin >> students[i].Name;

cout << "Enter Student ID: ";

cin >> students[i].Id;

cout << "\t\t\t\t\t\t Enter Test Scores" << endl;

for (int j = 0; j < Total\_tests; j++)

{

cout << "Enter Test Score of test " << j + 1 << " :";

cin >> students[i].tests[j];

}

}

average(Total\_Students, Total\_tests, students);

allocategrade(Total\_Students, students);

cout << " \t\t\t\tSTUDENTS DATA" << endl;

cout << "----------------------------------------------------------------------" << endl;

cout << "NAME\t\tID\t\t Average \t\tGrade" << endl;

cout << "----------------------------------------------------------------------" << endl;

for (int i = 0; i < Total\_Students; i++)

{

cout << students[i].Name << "\t\t";

cout << students[i].Id << "\t\t";

cout << students[i].Average << "\t\t";

cout << students[i].Grade << endl;

}

}

void\* average(int Total\_Students, int Total\_tests, CourseGrade\* students)

{

int average, sum;

for (int i = 0; i < Total\_Students; i++)

{

sum = 0;

for (int j = 0; j < Total\_tests; j++)

{

sum = sum + students[i].tests[j];

}

students[i].Average = (sum / Total\_tests);

}

}

void\* allocategrade(int Total\_Students, CourseGrade\* students)

{

int average;

for (int i = 0; i < Total\_Students; i++)

{

if (students[i].Average > 90)

{

students[i].Grade = 'A';

}

else if (students[i].Average > 80 && students[i].Average <= 90)

{

students[i].Grade = 'B';

}

else if (students[i].Average > 70 && students[i].Average <= 80)

{

students[i].Grade = 'C';

}

else if (students[i].Average > 60 && students[i].Average <= 70)

{

students[i].Grade = 'D';

}

else

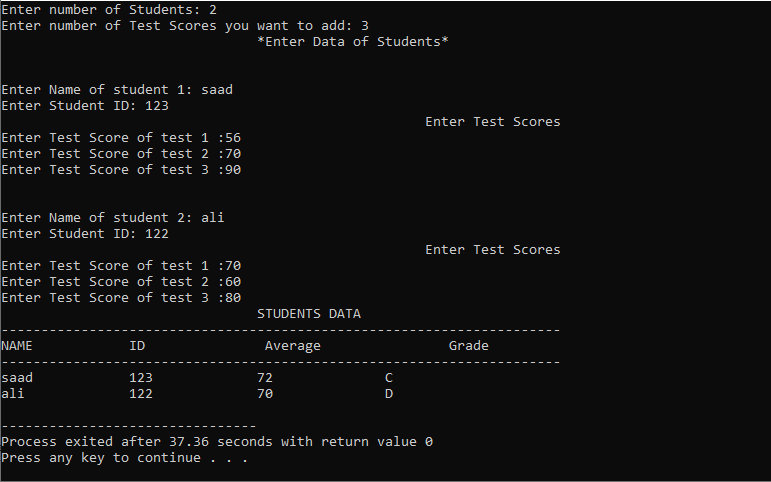
{

students[i].Grade = 'F';

}

}

}



**TASK 2:**

#include<iostream>

using namespace std;

struct date

{

int day;

int month;

int year;

};

struct student

{

int student\_id;

string student\_name;

date dateOfBirth;

char gender;

int semester;

string department;

float gpa;

string extracurricularSkill;

};

void display(student\* st, int s, int k)

{

cout << "Id " << "Name " << "Gender " << "Semester " << "GPA " << "Department " << " DOB " << "Extra" << endl;

for (int i = 0; i < k; i++)

{

if (st[i].student\_id != -1)

{

cout << st[i].student\_id << " " << st[i].student\_name << " " << st[i].gender << " " << st[i].semester;

cout << " " << st[i].gpa << " " << st[i].department << " " << st[i].dateOfBirth.day << "/";

cout << st[i].dateOfBirth.month << "/" << st[i].dateOfBirth.year << " " << st[i].extracurricularSkill << endl;

}

}

}

void addStudent(student\* st, int s, int& i)

{

cout << "Enter student id" << endl;

cin >> st[i].student\_id;

cout << "Enter student name" << endl;

cin >> st[i].student\_name;

cout << "Enter student DOB" << endl;

cout << "day: ";

cin >> st[i].dateOfBirth.day;

cout << " month: ";

cin >> st[i].dateOfBirth.month;

cout << " Year: ";

cin >> st[i].dateOfBirth.year;

cout << "Enter student department" << endl;

cin >> st[i].department;

cout << "Enter student gender" << endl;

cin >> st[i].gender;

cout << "Enter student GPA" << endl;

cin >> st[i].gpa;

cout << "Enter student semester" << endl;

cin >> st[i].semester;

cout << "Enter student extra curricular activity" << endl;

cout << "1. Society Member" << endl;

cout << "2. Sports Man" << endl;

cout << "3. Internet Geek" << endl;

cout << "4. none" << endl;

cin >> st[i].extracurricularSkill;

}

bool update(student\* st, int size)

{

int id, c;

cout << "Enter id of student to update data" << endl;

cin >> id;

for (int i = 0; i < size; i++)

{

if (id == st[i].student\_id)

{

cout << "Enter student name" << endl;

cin >> st[i].student\_name;

cout << "Enter student DOB" << endl;

cout << "day: ";

cin >> st[i].dateOfBirth.day;

cout << " month: ";

cin >> st[i].dateOfBirth.month;

cout << " Year: ";

cin >> st[i].dateOfBirth.year;

cout << "Enter student department" << endl;

cin >> st[i].department;

cout << "Enter student gender" << endl;

cin >> st[i].gender;

cout << "Enter student GPA" << endl;

cin >> st[i].gpa;

cout << "Enter student semester" << endl;

cin >> st[i].semester;

cout << "Enter student extra curricular activity" << endl;

cout << "1. Society Member" << endl;

cout << "2. Sports Man" << endl;

cout << "3. Internet Geek" << endl;

cout << "4. none" << endl;

cin >> st[i].extracurricularSkill;

return true;

}

}

return false;

}

bool del(student\* st, int s)

{

int id;

cout << "Enter id to remove student data" << endl;

cin >> id;

for (int i = 0; i < s; i++)

{

if (id == st[i].student\_id)

{

st[i].student\_id = -1;

return true;

}

}

return false;

}

void searchBy(student\* st, int s)

{

int c, sem;

string name;

cout << "How you want to search" << endl;

cout << "1. By department" << endl;

cout << "2. By Semester" << endl;

cin >> c;

switch (c)

{

case 1:

cout << "Enter department name" << endl;

cin >> name;

cout << "Id " << "Name " << " Gender " << "Semester " << "GPA " << "Department " << "DOB " << "Extra" << endl;

for (int i = 0; i < s; i++)

{

if (name == st[i].department)

{

cout << st[i].student\_id << " " << st[i].student\_name << " " << st[i].gender << " " << st[i].semester;

cout << " " << st[i].gpa << " " << st[i].department << " " << st[i].dateOfBirth.day << "/";

cout << st[i].dateOfBirth.month << "/" << st[i].dateOfBirth.year << " " << st[i].extracurricularSkill << endl;

}

}

break;

case 2:

cout << "Enter semester number" << endl;

cin >> sem;

cout << "Id " << "Name " << " Gender " << "Semester " << "GPA " << "Department " << "DOB " << "Extra" << endl;

for (int i = 0; i < s; i++)

{

if (sem == st[i].semester)

{

cout << st[i].student\_id << " " << st[i].student\_name << " " << st[i].gender << " " << st[i].semester;

cout << " " << st[i].gpa << " " << st[i].department << " " << st[i].dateOfBirth.day << "/";

cout << st[i].dateOfBirth.month << "/" << st[i].dateOfBirth.year << " " << st[i].extracurricularSkill << endl;

}

}

break;

default:

cout << "Invalid option" << endl;

}//end of switch

}

bool searchN(student\* st, int s)

{

string name;

cout << "Enter student name" << endl;

cin >> name;

for (int i = 0; i < s; i++)

{

if (name == st[i].student\_name)

{

cout << "Id " << "Name " << "Gender " << "Semester " << "GPA " << "Department " << " DOB " << "Extra" << endl;

cout << st[i].student\_id << " " << st[i].student\_name << " " << st[i].gender << " " << st[i].semester;

cout << " " << st[i].gpa << " " << st[i].department << " " << st[i].dateOfBirth.day << "/";

cout << st[i].dateOfBirth.month << "/" << st[i].dateOfBirth.year << " " << st[i].extracurricularSkill << endl;

return true;

}

}

return false;

}

void seprate(student\*& n, student\*& v, student\*& d, student\* st, int s)

{

int nC = 0, vC = 0, dC = 0;

int c1 = 0, c2 = 0, c3 = 0;

for (int i = 0; i < s; i++)

{

if (st[i].gpa >= 3.5)

{

nC++;

}

if (st[i].gpa >= 3)

{

vC++;

}

if (st[i].gpa < 2)

{

dC++;

}

}

n = new student[nC];

v = new student[vC];

d = new student[dC];

for (int i = 0; i < s; i++)

{

if (st[i].gpa >= 3.5)

{

n[c1].dateOfBirth.day = st[i].dateOfBirth.day;

n[c1].dateOfBirth.month = st[i].dateOfBirth.month;

n[c1].dateOfBirth.year = st[i].dateOfBirth.year;

n[c1].department = st[i].department;

n[c1].extracurricularSkill = st[i].extracurricularSkill;

n[c1].gender = st[i].gender;

n[c1].gpa = st[i].gpa;

n[c1].semester = st[i].semester;

n[c1].student\_id = st[i].student\_id;

n[c1].student\_name = st[i].student\_name;

c1++;

st[i].student\_id = -1;

}

if (st[i].gpa >= 3)

{

v[c2].dateOfBirth.day = st[i].dateOfBirth.day;

v[c2].dateOfBirth.month = st[i].dateOfBirth.month;

v[c2].dateOfBirth.year = st[i].dateOfBirth.year;

v[c2].department = st[i].department;

v[c2].extracurricularSkill = st[i].extracurricularSkill;

v[c2].gender = st[i].gender;

v[c2].gpa = st[i].gpa;

v[c2].semester = st[i].semester;

v[c2].student\_id = st[i].student\_id;

v[c2].student\_name = st[i].student\_name;

c2++;

st[i].student\_id = -1;

}

if (st[i].gpa < 2)

{

d[c3].dateOfBirth.day = st[i].dateOfBirth.day;

d[c3].dateOfBirth.month = st[i].dateOfBirth.month;

d[c3].dateOfBirth.year = st[i].dateOfBirth.year;

d[c3].department = st[i].department;

d[c3].extracurricularSkill = st[i].extracurricularSkill;

d[c3].gender = st[i].gender;

d[c3].gpa = st[i].gpa;

d[c3].semester = st[i].semester;

d[c3].student\_id = st[i].student\_id;

d[c3].student\_name = st[i].student\_name;

c3++;

st[i].student\_id = -1;

}

}

}

int main()

{

int size = 50;

int i = 0;

student\* nerd = NULL, \* vibrant = NULL, \* dumb = NULL;

student\* st = new student[size];

int n;

a: cout << "Chose an option" << endl;

cout << "1. Add student" << endl;

cout << "2. Display" << endl;

cout << "3. Update student info" << endl;

cout << "4. Search student by name" << endl;

cout << "5. Search by semester/department" << endl;

cout << "6. Delete student data" << endl;

cout << "7. Seprate on basis of skill" << endl;

cout << "8. To exit" << endl;

cin >> n;

if (n == 1)

{

addStudent(st, size, i);

i++;

goto a;

}

if (n == 2)

{

display(st, size, i);

goto a;

}

if (n == 3)

{

if (update(st, size))

{

cout << "Database updated sucessfully" << endl;

}

else

{

cout << "ID you entered doesn't exist" << endl;

}

goto a;

}

if (n == 4)

{

if (!searchN(st, size))

{

cout << "Name you entered doesn't exist" << endl;

}

goto a;

}

if (n == 5)

{

searchBy(st, size);

goto a;

}

if (n == 6)

{

if (del(st, size))

{

cout << "Student data deleted sucessflly" << endl;

}

else

{

cout << "ID entered doesn't exist" << endl;

}

goto a;

}

if (n == 7)

{

seprate(nerd, vibrant, dumb, st, size);

goto a;

}

if (n == 8)

{

exit;

}

if (n > 8 || n < 1)

{

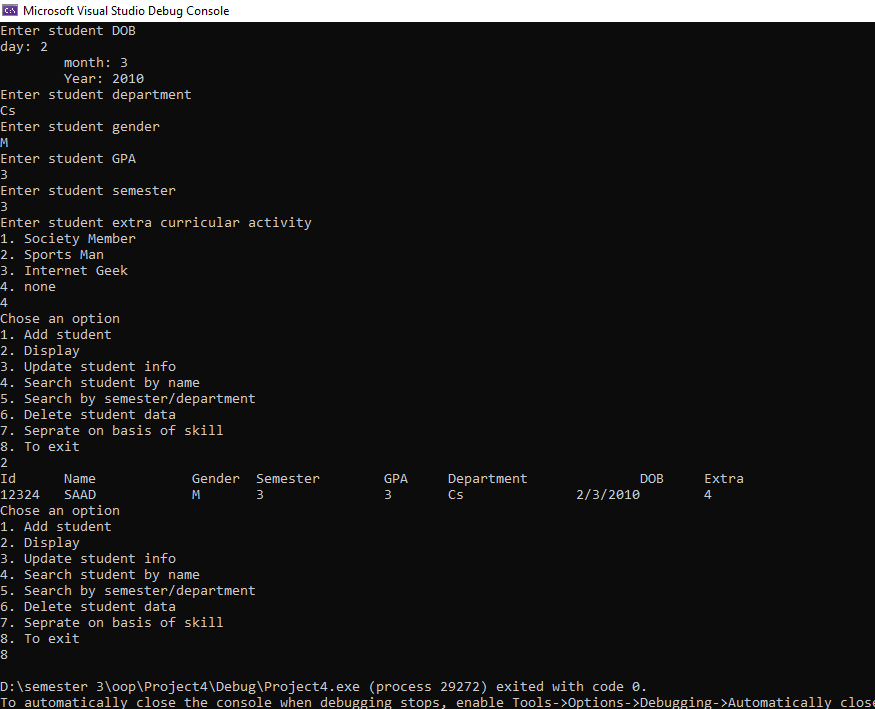
cout << "Wrong input\n";

goto a;

}

system("pause");

}



**Task 3:**

#include<iostream>

#include<cstring>

using namespace std;

int main() {

string str = "DSC is a Student Community";

int StrLength = str.length();

int startIndex = 0;

int endIndex = 0;

int minLength = StrLength;

int maxLength = 0;

int currentLength;

string smallest, largest;

while (endIndex <= StrLength) {

if (str[endIndex] != '\0' && str[endIndex] != ' ')

endIndex++;

else {

currentLength = endIndex - startIndex;

if (currentLength < minLength) {

smallest = str.substr(startIndex, currentLength);

minLength = currentLength;

}

if (currentLength > maxLength) {

largest = str.substr(startIndex, currentLength);

maxLength = currentLength;

}

endIndex++;

startIndex = endIndex;

}

}

cout << "input :DSC is a Student Community" << endl;

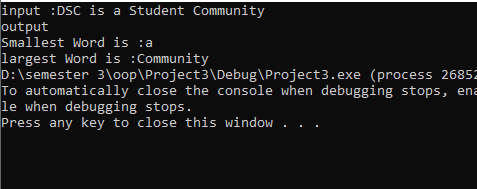
cout << "output" << endl;

cout << "Smallest Word is :" << smallest << endl;

cout << "largest Word is :" << largest;

system("pause");

}



**TASK 4:**

#include<iostream>

using namespace std;

void Mul(int Array1[][3], int Array2[][3], int Array3[][3], int R1, int R2, int C1, int C2)

{

static int i = 0, j = 0, k = 0;

if (i >= R1)

{

return;

}

if (j < C2)

{

if (k < C1)

{

Array3[i][j] += Array1[i][k] \* Array2[k][j];

k++;

Mul(Array1, Array2, Array3, R1, R2, C1, C2);

}

k = 0;

j++;

Mul(Array1, Array2, Array3, R1, R2, C1, C2);

}

j = 0;

i++;

Mul(Array1, Array2, Array3, R1, R2, C1, C2);

}

void Display(int Array1[][3], int Array2[][3], int R1, int R2, int C1, int C2)

{

int Array3[3][3] = { 0 };

if (R1 != C2)

{

cout << "Not Possible!" << endl;

}

Mul(Array1, Array2, Array3, R1, R2, C1, C2);

cout << "The Matrix After Multiplication Is " << endl;

for (int i = 0; i < R1; i++)

{

for (int j = 0; j < C2; j++)

{

cout << Array3[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

int main()

{

int R1, C1, R2, C2;

int Array1[3][3];

int Array2[3][3];

cout << "Enter Row 1: ";

cin >> R1;

cout << "Enter Row 2: ";

cin >> R2;

cout << "Enter Coloumn 1: ";

cin >> C1;

cout << "Enter Coloumn 2: ";

cin >> C2;

cout << "Enter Elements of Array 1 " << endl;

for (int i = 0; i < R1; i++)

{

for (int j = 0; j < C2; j++)

{

cin >> Array1[i][j];

}

}

cout << "Enter Elements of Array 2 " << endl;

for (int i = 0; i < R2; i++)

{

for (int j = 0; j < C1; j++)

{

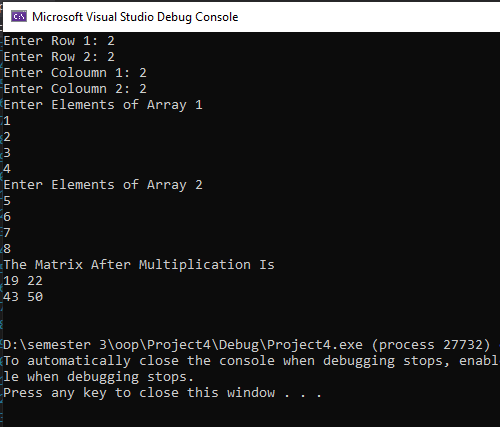
cin >> Array2[i][j];

}

}

Display(Array1, Array2, R1, R2, C1, C2);

}



**Task 5**

**(a)Selction sort:**

#include<iostream>

using namespace std;

const int size = 10;

void selection\_sort(int\* ptr)

{

const int size = 10;

int array[size], array2[size];

for (int count = 0; count < size; count++)

{

for (int counter = count + 1; counter < size; counter++)

{

if (\*(ptr + count) > \* (ptr + counter))

{

\*(ptr + count) = \*(ptr + count) + \*(ptr + counter);

\*(ptr + counter) = \*(ptr + count) - \*(ptr + counter);

\*(ptr + count) = \*(ptr + count) - \*(ptr + counter);

}

}

}

for (int count = 0; count < size; count++)

{

array2[count] = \*(ptr + count);

}

}

int main()

{

const int size = 10;

int array[size], array2[size];

int\* ptr;

cout << "Plese enter the elements := ";

for (int input = 0; input < size; input++)

{

cin >> array[input];

}

ptr = array;

selection\_sort(ptr);

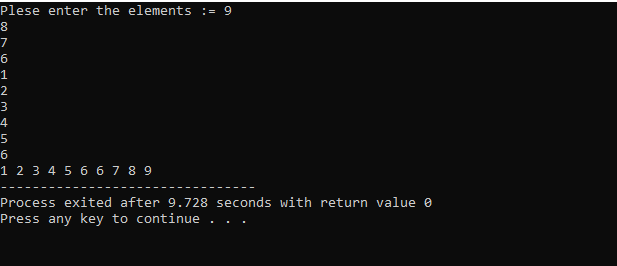
for (int input = 0; input < size; input++)

{

cout << array2[input] << " ";

}

}



**(b)Bubble sort:**

#include <stdio.h>

#include<iostream>

using namespace std;

int main()

{

const int size = 10;

int choice;

int array[size];

cout << "Enter the elements in the array:";

for (int i = 0; i < size; i++)

{

cin >> array[i];

}

int temp;

for (int i = 1; i < size; i++)

{

for (int j = 0; j < size - i; j++)

{

if (\*(array + j) > \* (array + j + 1))

{

temp = \*(array + j);

\*(array + j) = \*(array + j + 1);

\*(array + j + 1) = temp;

}

}

} cout << "The array after bubble sorting:" << endl;

for (int i = 0; i < size; i++)

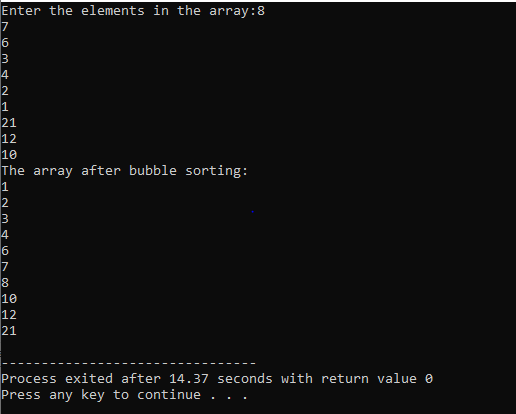
{

cout << array[i] << endl;

}

return 0;

}



**(c)Insertion sort:**

#include <stdio.h>

#include<iostream>

using namespace std;

int main()

{

const int size = 10;

int choice;

int array[size];

cout << "Enter the elements in the array:";

for (int i = 0; i < size; i++)

{

cin >> array[i];

}

int key, j;

for (int i = 1; i < size; i++)

{

key = array[i];

j = i;

while (j > 0 && array[j - 1] > key)

{

array[j] = array[j - 1];

j--;

}

array[j] = key;

}

cout << "The array after insertion sorting is :" << endl;

for (int i = 0; i < size; i++)

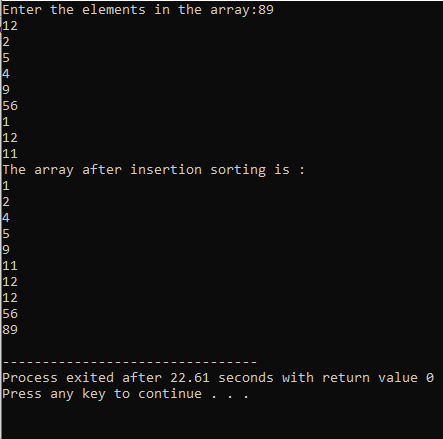
{

cout << array[i] << endl;

}

return 0;

}



**Question no 6:**

**(Multiplicaion of matrices)**

#include<iostream>

using namespace std;

int\*\* ma()

{

int\*\* array1;

array1 = new int\* [3];

for (int i = 0; i < 3; i++)

{

array1[i] = new int[3];

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << "Enter the values of array" << "[" << i << "]" << "[" << j << "] :";

cin >> array1[i][j];

}

cout << endl;

}

return array1;

}

int main()

{

int sum;

int product1[3][3] = { 0 };

int\*\* ptr = NULL;

ptr = ma();

int\*\* ptr2 = NULL;

ptr2 = ma();

int product[3][3];

cout << "The values of a matrix A " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr[i][j];

cout << "\t";

}

cout << endl;

}

cout << "The values of a matrix B " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr2[i][j];

cout << "\t";

}

cout << endl;

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

product1[i][j] += ptr[i][j] \* ptr2[j][k];

}

}

cout << endl;

}

cout << "The matrix after multiplication " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << product1[i][j];

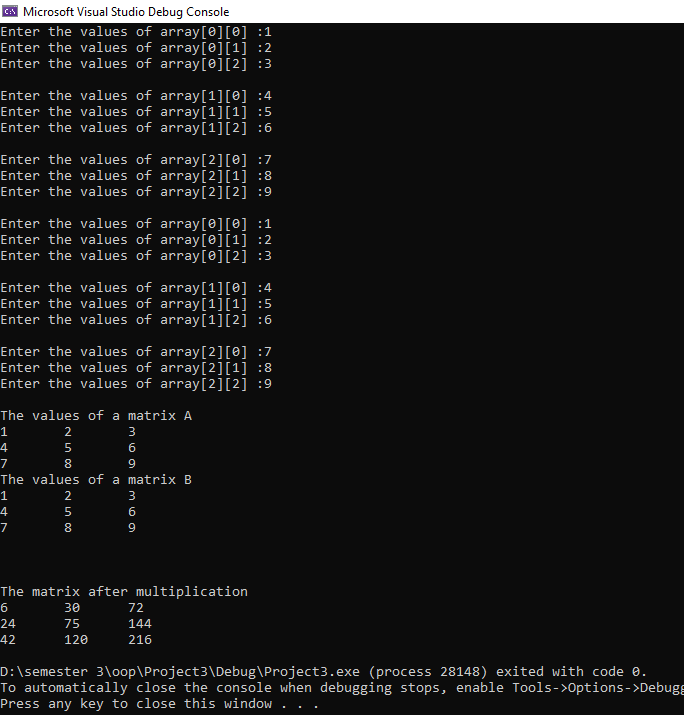
cout << "\t";

}

cout << endl;

}

}



**(Addition of matrices)**

#include<iostream>

using namespace std;

int\*\* ma()

{

int\*\* array1;

array1 = new int\* [3];

for (int i = 0; i < 3; i++)

{

array1[i] = new int[3];

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << "Enter the values of array" << "[" << i << "]" << "[" << j << "] :";

cin >> array1[i][j];

}

cout << endl;

}

return array1;

}

int main()

{

int sum;

int product1[3][3] = { 0 };

int\*\* ptr = NULL;

ptr = ma();

int\*\* ptr2 = NULL;

ptr2 = ma();

int product[3][3];

cout << "The values of a matrix A " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr[i][j];

cout << "\t";

}

cout << endl;

}

cout << "The values of a matrix B " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr2[i][j];

cout << "\t";

}

cout << endl;

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

product1[i][j] = ptr[i][j] + ptr2[i][j];

}

}

cout << endl;

}

cout << "The matrix after multiplication " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << product1[i][j];

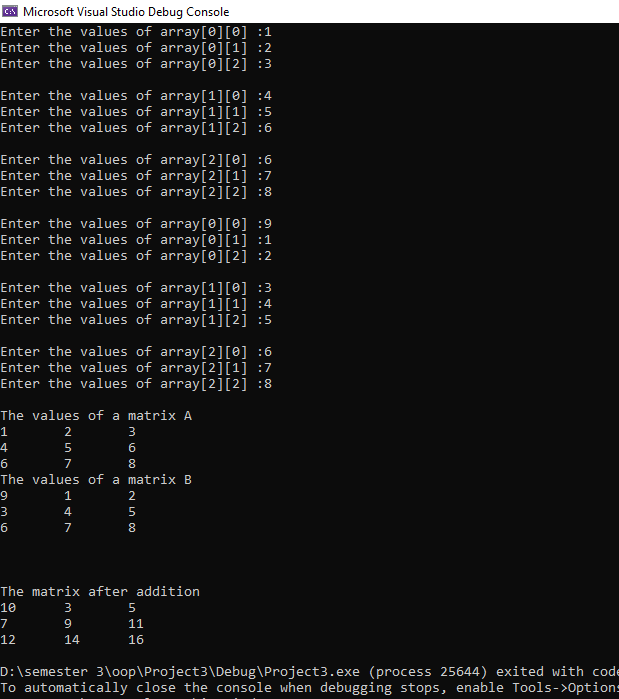
cout << "\t";

}

cout << endl;

}

}



**(Subtraction of matrices)**

#include<iostream>

using namespace std;

int\*\* ma()

{

int\*\* array1;

array1 = new int\* [3];

for (int i = 0; i < 3; i++)

{

array1[i] = new int[3];

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << "Enter the values of array" << "[" << i << "]" << "[" << j << "] :";

cin >> array1[i][j];

}

cout << endl;

}

return array1;

}

int main()

{

int sum;

int product1[3][3] = { 0 };

int\*\* ptr = NULL;

ptr = ma();

int\*\* ptr2 = NULL;

ptr2 = ma();

int product[3][3];

cout << "The values of a matrix A " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr[i][j];

cout << "\t";

}

cout << endl;

}

cout << "The values of a matrix B " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << ptr2[i][j];

cout << "\t";

}

cout << endl;

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

product1[i][j] = ptr[i][j] - ptr2[i][j];

}

}

cout << endl;

}

cout << "The matrix after multiplication " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << product1[i][j];

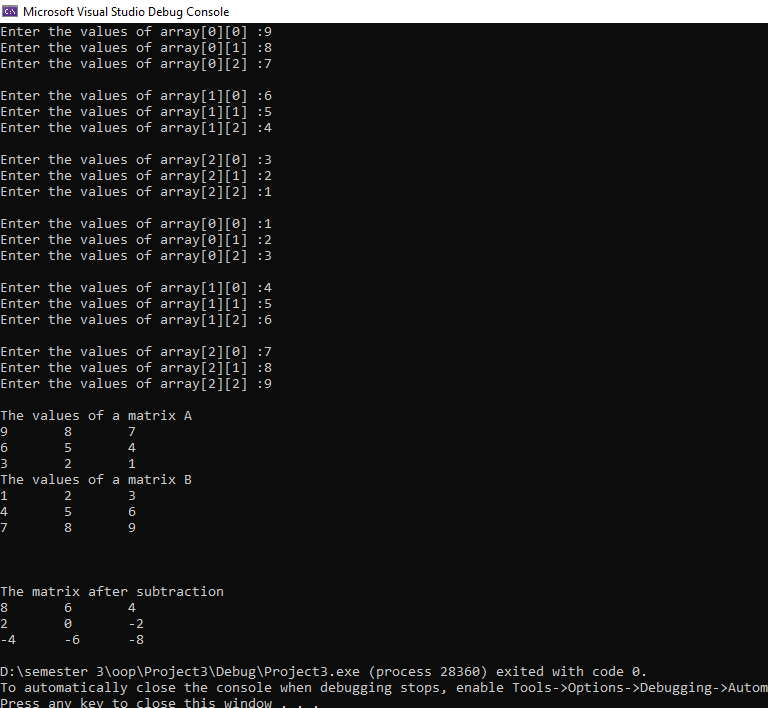
cout << "\t";

}

cout << endl;

}

}



**(Transpose of matrices)**

#include<iostream>

using namespace std;

int main()

{

int sum;

int product1[3][3] = { 0 };

int\*\* array1;

array1 = new int\* [3];

for (int i = 0; i < 3; i++)

{

array1[i] = new int[3];

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << "Enter the values of array" << "[" << i << "]" << "[" << j << "] :";

cin >> array1[i][j];

}

cout << endl;

}

int result[3][3];

cout << "The values of a matrix A " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << array1[i][j];

cout << "\t";

}

cout << endl;

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

result[i][j] = array1[j][i];

}

}

cout << endl;

}

cout << "The matrix after transpose " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << result[i][j];

cout << "\t";

}

cout << endl;

}

}

