**PF Assignment #03**

**Question #01**

#include<stdio.h>

void matriceTWO() {

int i, j;

int matrice[2][2];

printf("Enter the elements of 2x2 matrice:\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

scanf("%d", & matrice[i][j]);

}

}

printf("\n");

int max = matrice[0][1];

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", matrice[i][j]);

if (max < matrice[i][j]) {

max = matrice[i][j];

}

}

printf("\n");

}

printf("Output: %d\n\n", max);

}

void matriceFOUR() {

int i, j;

int matrice[4][4];

int sub1[2][2], sub2[2][2], sub3[2][2], sub4[2][2], sub5[2][2], sub6[2][2], sub7[2][2], sub8[2][2], sub9[2][2], sub10[2][2], sub11[2][2], sub12[2][2], sub13[2][2], sub14[2][2], sub15[2][2], sub16[2][2];

printf("Enter the elements of 4x4 matrice:\n");

for (i=0; i<4; i++) {

for (j=0; j<4; j++) {

scanf("%d", & matrice[i][j]);

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub1[i][j] = matrice[i][j];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub2[i][j] = matrice[i][j + 2];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub3[i][j] = matrice[i + 2][j];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub4[i][j] = matrice[i + 2][j + 2];

}

}

int max1 = sub1[0][0], max2 = sub2[0][0], max3 = sub3[0][0], max4 = sub4[0][0];

for (i=0; i<2; i++) {

for (j=0; j<2; j++) {

printf("%d\t", sub1[i][j]);

if (max1 < sub1[i][j]) {

max1 = sub1[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub2[i][j]);

if (max2 < sub2[i][j]) {

max2 = sub2[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub3[i][j]);

if (max3 < sub3[i][j]) {

max3 = sub3[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub4[i][j]);

if (max4 < sub4[i][j]) {

max4 = sub4[i][j];

}

}

printf("\n");

}

printf("\n");

int output[2][2] = {{max1,max2},{max3,max4}};

printf("Output:\n");

for (i=0; i<2; i++) {

for (j=0; j<2; j++) {

printf("%d\t", output[i][j]);

}

printf("\n");

}

}

void matriceEIGHT() {

int i, j;

int matrice[4][4];

int sub1[2][2], sub2[2][2], sub3[2][2], sub4[2][2], sub5[2][2], sub6[2][2], sub7[2][2], sub8[2][2];

int sub9[2][2], sub10[2][2], sub11[2][2], sub12[2][2], sub13[2][2], sub14[2][2], sub15[2][2], sub16[2][2];

printf("Enter the elements of 8x8 matrice:\n");

for (i=0; i<4; i++) {

for (j=0; j<4; j++) {

scanf("%d", & matrice[i][j]);

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub1[i][j] = matrice[i][j];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub2[i][j] = matrice[i][j + 2];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub3[i][j] = matrice[i + 2][j];

}

}

for (int i=0; i<2; i++) {

for (int j=0; j<2; j++) {

sub4[i][j] = matrice[i + 2][j + 2];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub5[i][j] = matrice[i + 2][j];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub6[i][j] = matrice[i + 2][j + 2];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub7[i][j] = matrice[i + 2][j + 4];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub8[i][j] = matrice[i + 2][j + 6];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub9[i][j] = matrice[i + 4][j];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub10[i][j] = matrice[i + 4][j + 2];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub11[i][j] = matrice[i + 4][j + 4];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub12[i][j] = matrice[i + 4][j + 6];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub13[i][j] = matrice[i + 6][j];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub14[i][j] = matrice[i + 6][j + 2];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub15[i][j] = matrice[i + 6][j + 4];

}

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

sub16[i][j] = matrice[i + 6][j + 6];

}

}

int max1 = sub1[0][0], max2 = sub2[0][0], max3 = sub3[0][0], max4 = sub4[0][0], max5 = sub5[0][0], max6 = sub6[0][0], max7 = sub7[0][0], max8 = sub8[0][0], max9 = sub9[0][0], max10 = sub10[0][0], max11 = sub11[0][0], max12 = sub12[0][0], max13 = sub13[0][0], max14 = sub14[0][0], max15 = sub15[0][0], max16 = sub16[0][0];

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub1[i][j]);

if (max1 < sub1[i][j]) {

max1 = sub1[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub2[i][j]);

if (max2 < sub2[i][j]) {

max2 = sub2[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub3[i][j]);

if (max3 < sub3[i][j]) {

max3 = sub3[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub4[i][j]);

if (max4 < sub4[i][j]) {

max4 = sub4[i][j];

}

}

printf("\n");

}

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub5[i][j]);

if (max5 < sub5[i][j]) {

max5 = sub5[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub6[i][j]);

if (max6 < sub6[i][j]) {

max6 = sub6[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub7[i][j]);

if (max7 < sub7[i][j]) {

max7 = sub7[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub8[i][j]);

if (max8 < sub8[i][j]) {

max8 = sub8[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub9[i][j]);

if (max9 < sub9[i][j]) {

max9 = sub9[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub10[i][j]);

if (max10 < sub10[i][j]) {

max10 = sub10[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub11[i][j]);

if (max11 < sub11[i][j]) {

max11 = sub11[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub12[i][j]);

if (max12 < sub12[i][j]) {

max12 = sub12[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub13[i][j]);

if (max13 < sub13[i][j]) {

max13 = sub13[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub14[i][j]);

if (max14 < sub14[i][j]) {

max14 = sub14[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub15[i][j]);

if (max15 < sub15[i][j]) {

max15 = sub15[i][j];

}

}

printf("\n");

}

printf("\n");

for (i = 0; i < 2; i++) {

for (j = 0; j < 2; j++) {

printf("%d\t", sub16[i][j]);

if (max16 < sub16[i][j]) {

max16 = sub16[i][j];

}

}

printf("\n");

}

printf("\n");

int output[4][4] = {{max1,max2,max3,max4},{max5, max6,max7,max8},{max9,max10,max11,max12},{max13,max14,max15,max16}};

printf("Output:\n");

for (i = 0; i < 4; i++) {

for (j = 0; j < 4; j++) {

printf("%d\t", output[i][j]);

}

printf("\n");

}

printf("\n");

}

int main() {

printf("Muhammad Hammad (23K-2005)\n");

int choice;

printf("1) 2x2 matrice\n2) 4x4 matrice\n3) 8x8 matrice\n4) Exit\n");

printf("choose a matrix to continue: ");

scanf("%d", &choice);

printf("\n");

switch (choice) {

case 1:

matriceTWO();

break;

case 2:

matriceFOUR();

break;

case 3:

matriceEIGHT();

break;

case 4:

printf("The program has exited.\n");

break;

default:

printf("Invalid input. Please select from the above.\n");

}

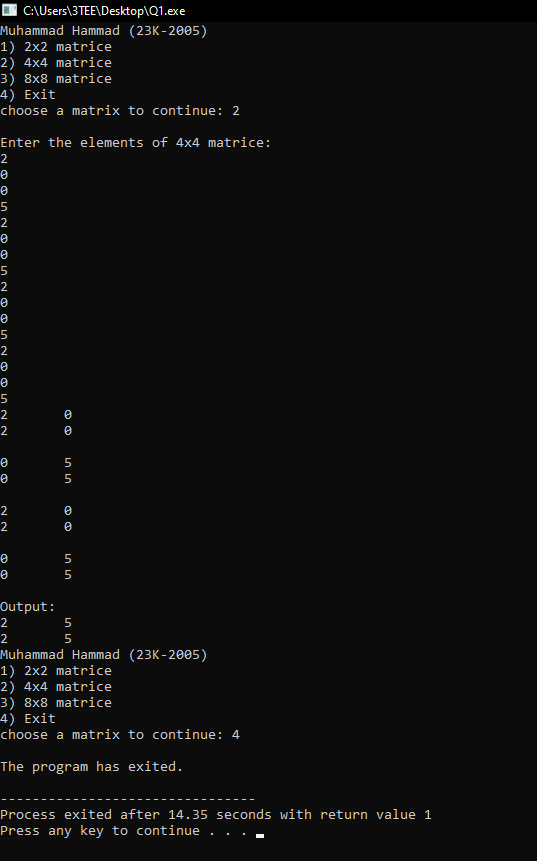
if (choice>=4 || choice <=0 || choice>='a') {

exit(1);

}

main ();

return 0; }



**Question #03:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct EmployeePersonal {

int id;

char name[40];

} EmployeePersonal;

typedef struct EmployeeDepartment {

int id;

int salary;

} EmployeeDepartment;

void combineEmployees(int n);

int main(void) {

printf("Muhammad Hammad (23K-2005)\n\n");

int i=0, n=0;

printf("Please enter the number of employees: ");

scanf("%d", &n);

EmployeePersonal personalInfo[n];

for (i = 0; i < n; i++) {

printf("Enter the id of employee %d: ", i+1);

scanf("%d", &personalInfo[i].id);

fflush(stdin);

printf("Enter the name of employee %d: ", i+1);

fflush(stdin);

gets(personalInfo[i].name);

}

FILE \*personalFile = fopen("Personal.txt", "w");

if (personalFile == NULL) {

printf("Error opening the file\n");

exit(0);

}

for (i=0; i<n; i++) {

fprintf(personalFile, "%d %s\n", personalInfo[i].id, personalInfo[i].name);

}

fclose(personalFile);

EmployeeDepartment departmentInfo[n];

for (i=0; i<n; i++) {

printf("Enter id of employee %d: ", i + 1);

scanf("%d", &departmentInfo[i].id);

printf("Enter salary of employee %d: ", i + 1);

scanf("%d", &departmentInfo[i].salary);

}

FILE \*departmentFile = fopen("Department.txt", "w");

if (departmentFile == NULL) {

printf("Error opening the file\n"); // agar file na bani tou, jo sir ne bataya tha

exit(0);

}

for (i=0; i<n; i++) {

fprintf(departmentFile, "%d %d\n", departmentInfo[i].id, departmentInfo[i].salary);

}

fclose(departmentFile);

combineEmployees(n);

return 0;

}

void combineEmployees(int n) {

int i, j;

EmployeePersonal personalData[n];

EmployeeDepartment departmentData[n];

FILE \*personalFile = fopen("Personal.txt", "r");

if (personalFile == NULL) {

printf("Error opening the file\n"); // agar file na bani tou, jo sir ne bataya tha

exit(0);

}

for (i=0; i<n; i++) {

fscanf(personalFile, "%d %s\n", &personalData[i].id, &personalData[i].name);

}

fclose(personalFile);

FILE \*departmentFile = fopen("Department.txt", "r");

if (departmentFile == NULL) {

printf("Error opening the file\n"); // agar file na bani tou, jo sir ne bataya tha

exit(0);

}

for (i=0; i<n; i++) {

fscanf(departmentFile, "%d %d\n", &departmentData[i].id, &departmentData[i].salary);

}

fclose(departmentFile);

FILE \*mergedFile = fopen("Combine.txt", "w");

if (mergedFile == NULL) {

printf("Error opening the file\n");

exit(0);

}

for (i=0; i<n; i++) {

for (j=0; j<n; j++) {

if (personalData[i].id == departmentData[j].id) {

fprintf(mergedFile, "%d %s %d\n", personalData[i].id, personalData[i].name, departmentData[j].salary);

}

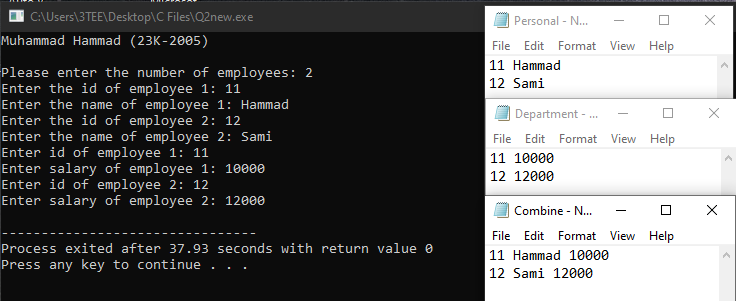
}

}

fclose(mergedFile);

}

Output:



**Question #04:**

#include<stdio.h>

#include<string.h>

struct joining\_date {

int year;

int month;

int date;

int hour;

int minute;

int second;

} j\_d;

struct worker {

int worker\_id;

char first\_name[20];

char last\_name[20];

int salary;

char department[20];

struct joining\_date j\_d;

};

struct bonus\_date {

int year;

int month;

int date;

} b\_d;

struct bonus {

int worker\_ref\_id;

int bonus\_amount;

struct bonus\_date b\_d;

};

struct affected\_from {

int year;

int month;

int date;

int hour;

int minute;

int second;

} a\_f;

struct title {

int worker\_ref\_id;

char worker\_title[20];

struct affected\_from a\_f;

};

int main(void) {

struct worker wk[8];

struct bonus bonus[5];

struct title title[8];

int i, j, n;

printf("Enter the number of workers: ");

scanf("%d", &n);

printf("\n");

for (i = 0; i < n; i++) {

printf("\tData for Worker %d:\n", i + 1);

printf("Enter the worker id: ");

scanf("%d", &wk[i].worker\_id);

printf("Enter the first name: ");

fflush(stdin);

gets(wk[i].first\_name);

printf("Enter the last name: ");

fflush(stdin);

gets(wk[i].last\_name);

printf("Enter the salary: ");

scanf("%d", &wk[i].salary);

printf("Enter the year of joining: ");

scanf("%d", &wk[i].j\_d.year);

printf("Enter the month of joining: ");

scanf("%d", &wk[i].j\_d.month);

printf("Enter the date of joining: ");

scanf("%d", &wk[i].j\_d.date);

printf("Enter the department: ");

fflush(stdin);

gets(wk[i].department);

printf("\n");

}

// Part A:

printf("Workers having the maximum salary from each department:\n");

int max1 = 0, sum1 = 0, index1 = -1;

for (i = 0; i < n; i++) {

if (strcmp(wk[i].department, "HR") == 0) {

sum1 = sum1 + wk[i].salary;

if (max1 < wk[i].salary) {

max1 = wk[i].salary;

index1 = i; // Update the index when a higher salary is found

}

}

}

if (index1 != -1) {

printf("%d\t", wk[index1].worker\_id);

printf("%s\t", wk[index1].first\_name);

printf("%s\t", wk[index1].last\_name);

printf("%d\t", wk[index1].salary);

printf("%d-%d-%d\t", wk[index1].j\_d.year, wk[index1].j\_d.month, wk[index1].j\_d.date);

printf("%s\t", wk[index1].department);

}

// Similar logic for the Admin department

int max2 = 0, sum2 = 0, index2 = -1;

for (i = 0; i < n; i++) {

if (strcmp(wk[i].department, "Admin") == 0) {

sum2 = sum2 + wk[i].salary;

if (max2 < wk[i].salary) {

max2 = wk[i].salary;

index2 = i; // Update the index when a higher salary is found

}

}

}

if (index2 != -1) {

printf("\n%d\t", wk[index2].worker\_id);

printf("%s\t", wk[index2].first\_name);

printf("%s\t", wk[index2].last\_name);

printf("%d\t", wk[index2].salary);

printf("%d-%d-%d\t", wk[index2].j\_d.year, wk[index2].j\_d.month, wk[index2].j\_d.date);

printf("%s\t", wk[index2].department);

}

// Similar logic for the Account department

int max3 = 0, sum3 = 0, index3 = -1;

for (i = 0; i < n; i++) {

if (strcmp(wk[i].department, "Account") == 0) {

sum3 = sum3 + wk[i].salary;

if (max3 < wk[i].salary) {

max3 = wk[i].salary;

index3 = i; // Update the index when a higher salary is found

}

}

}

if (index3 != -1) {

printf("\n%d\t", wk[index3].worker\_id);

printf("%s\t", wk[index3].first\_name);

printf("%s\t", wk[index3].last\_name);

printf("%d\t", wk[index3].salary);

printf("%d-%d-%d\t", wk[index3].j\_d.year, wk[index3].j\_d.month, wk[index3].j\_d.date);

printf("%s\t", wk[index3].department);

}

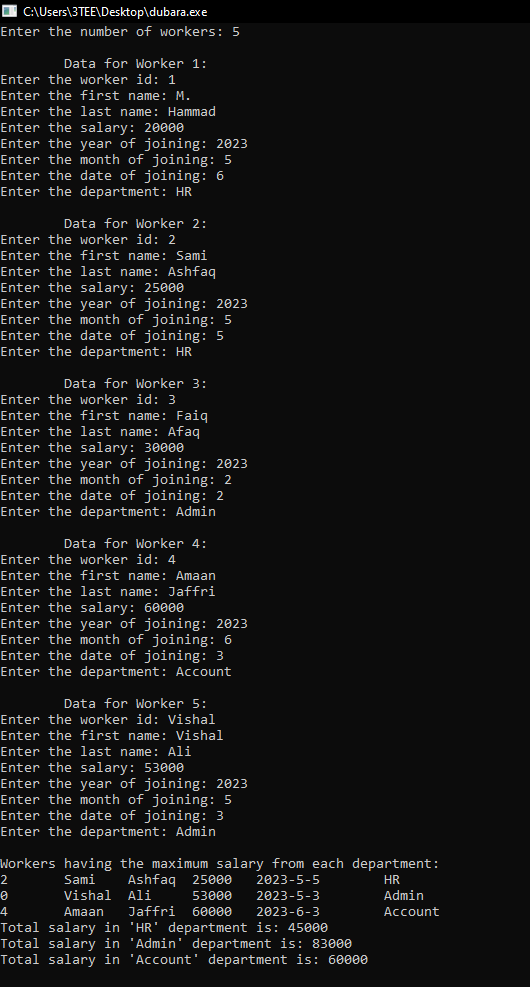
// Part B:

printf("\n");

printf("Total salary in 'HR' department is: %d\n", sum1);

printf("Total salary in 'Admin' department is: %d\n", sum2);

printf("Total salary in 'Account' department is: %d\n\n", sum3);}



**Question #02:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Employee {

char name[20];

char role[15];

int communication;

int teamwork;

int creativity;

};

void generateRandomName(char \*name) {

static const char \*names[] = {

"Abdul", "Bushra", "Chaudhry", "Danish", "Eman", "Faisal", "Gulzar", "Hina", "Irfan", "Javeria",

"Khalid", "Laila", "Mansoor", "Naima", "Omar", "Parveen", "Qasim", "Rukhsar", "Saadia", "Tariq"

};

int numNames = sizeof(names) / sizeof(names[0]);

int index = rand() % numNames;

strncpy(name, names[index], sizeof(names[index]));

}

struct Employee initializeEmployee(const char \*role) {

struct Employee employee;

generateRandomName(employee.name);

strncpy(employee.role, role, sizeof(employee.role));

employee.communication = rand() % 100 + 1;

employee.teamwork = rand() % 100 + 1;

employee.creativity = rand() % 100 + 1;

return employee;

}

int main() {

int i, j;

const char \*departments[] = {"HR", "Finance", "Marketing", "Logistics"};

const char \*roles[] = {"Director", "Manager", "Coordinator", "Team Member", "Trainee"};

struct Employee employeesByDepartment[4][5];

printf("Muhammad Hammad (23K-2005)\n\n");

for (i = 0; i < 4; ++i) {

for (j = 0; j < 5; ++j) {

generateRandomName(employeesByDepartment[i][j].name);

strncpy(employeesByDepartment[i][j].role, roles[j], sizeof(employeesByDepartment[i][j].role));

employeesByDepartment[i][j].communication = rand() % 100 + 1;

employeesByDepartment[i][j].teamwork = rand() % 100 + 1;

employeesByDepartment[i][j].creativity = rand() % 100 + 1;

}

}

// Printing the employee directory

for (i = 0; i < 4; ++i) {

printf("Department: %s\n", departments[i]);

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("Name\t\tRole\t\tCommunication\tTeamwork\tCreativity\n");

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

for (j = 0; j < 5; ++j) {

printf("%s\t\t%s\t\t%d\t\t%d\t\t%d\n", employeesByDepartment[i][j].name,

employeesByDepartment[i][j].role, employeesByDepartment[i][j].communication,

employeesByDepartment[i][j].teamwork, employeesByDepartment[i][j].creativity);

}

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

}

// Calculate the total scores for each department

int departmentSums[4] = {0};

for (i = 0; i < 4; ++i) {

for (j = 0; j < 5; ++j) {

departmentSums[i] += employeesByDepartment[i][j].communication +

employeesByDepartment[i][j].teamwork +

employeesByDepartment[i][j].creativity;

}

}

// Find the best department

int bestDepartmentIndex = 0;

for (i = 1; i < 4; ++i) {

if (departmentSums[i] > departmentSums[bestDepartmentIndex]) {

bestDepartmentIndex = i;

}

}

// Print department sums

printf("Department Sums:\n");

for (i = 0; i < 4; ++i) {

printf("%s: %d\n", departments[i], departmentSums[i]);

}

printf("\nBest Department: %s\n", departments[bestDepartmentIndex]);

// Print details for the best department

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("Name\t\tRole\t\tCommunication\tTeamwork\tCreativity\n");

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

// Use the bestDepartmentIndex to print details for the best department

for (j = 0; j < 5; ++j) {

printf("%s\t\t%s\t\t%d\t\t%d\t\t%d\n", employeesByDepartment[bestDepartmentIndex][j].name,

employeesByDepartment[bestDepartmentIndex][j].role,

employeesByDepartment[bestDepartmentIndex][j].communication,

employeesByDepartment[bestDepartmentIndex][j].teamwork,

employeesByDepartment[bestDepartmentIndex][j].creativity);

}

return 0;

}

Output:

