**ASSIGNMENT-2**

***Name: Jathin patel***

***Reg no:RA2311003050426***

***Year & Branch: II B.Tech CSE - F Section***

***Assignment No: 2***

***Github profile: https://github.com/Jathinpatel/dsa-assignment2***

***Implementation of Matrix Multiplication using Dynamic Memory Allocation. Ensureto allocate the memory using appropriate functions and access the array using pointers.***

***ANS:***

***#include<stdio.h>***

***#include<stdlib.h>***

***int main(){***

***int r,c,i,j,k;***

***//input no of rows and columns***

***printf("enter no of rows:");***

***scanf("%d",&r);***

***printf("enter no of colomns:");***

***scanf("%d",&c);***

***// dynamic memory allocation for matrix a,matrix b and resultant matrix***

***int \*\*a=(int\*\*)malloc(r\* sizeof(int\*));***

***for(i=0;i<r;i++){***

***a[i]=(int\*)malloc(c\*sizeof(int));***

***}***

***int \*\*b=(int\*\*)malloc(r\*sizeof(int\*));***

***for(i=0;i<r;i++){***

***b[i]=(int\*)malloc(c\*sizeof(int));***

***}***

***int \*\*mul =(int\*\*)malloc(r\*sizeof(int\*));***

***for(i=0;i<r;i++){***

***mul[i]=(int\*)malloc(c\*sizeof(int));***

***}***

***//get matrix 1 and 2 value***

***printf("enter matrix 1 values:\n");***

***for(i=0;i<r;i++){***

***for(j=0;j<c;j++){***

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

***}***

***}***

***printf("enter matrix 2 value:\n");***

***for(i=0;i<r;i++){***

***for(j=0;j<c;j++){***

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

***}***

***}***

***//matrix multiplication***

***printf("matrix multiplication:\n");***

***for(i=0;i<r;i++){***

***for(j=0;j<c;j++){***

***mul[i][j]=0;***

***for(k=0;k<c;k++){***

***mul[i][j]+=a[i][k]\*b[k][j];***

***}***

***}***

***}***

***//resultant matrix***

***for(i=0;i<r;i++){***

***for(j=0;j<c;j++){***

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

***}***

***printf("\n");***

***}***

***//free dynamically allocated memory***

***for(i=0;i<r;i++){***

***free(a[i]);***

***free(b[i]);***

***free(mul[i]);***

***}***

***free(a);***

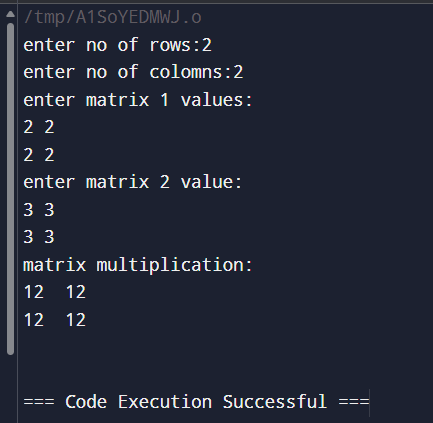
***free(b);***

***free(mul);***

***return 0;***

***}***

***OUTPUT:***

******

***2. You are given a task with creating a simple student management system using arraysthat will allow the user to manage student names. Implement the following operationson a list of student names using switch-case and arrays. After every operation,display the current list of students.***

***The operations to implement are:***

***(i) Creation of the list: Allow the user to create a list of student names by enteringthem one by one.***

***(ii) Insertion of a new student: Insert a new student's name into a specific positionin the list. The user should provide the name and the index at which it shouldbe inserted.***

***(iii) Deletion of a student: Delete a student's name from the list based on theirposition or name. Ask the user whether they want to delete by name or by index.***

***(iv) Traversal of the list: Display all the student names in the current order.***

***(v) Search for a student: Search for a student's name in the list and displaywhether or not the student is found, along with their position if present.***

***ANS:***

***#include <stdio.h>***

***#include <string.h>***

***#define MAX 100***

***#define MAX\_NAME\_LEN 50***

***void displayList(char students[][MAX\_NAME\_LEN], int size) {***

***printf("Student list: ");***

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

***printf("%s", students[i]);***

***if (i < size - 1) printf(", ");***

***}***

***printf("\n");***

***}***

***void createList(char students[][MAX\_NAME\_LEN], int \*size) {***

***printf("Enter the number of students: ");***

***scanf("%d", size);***

***for (int i = 0; i < \*size; i++) {***

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

***scanf("%s", students[i]);***

***}***

***}***

***void insertStudent(char students[][MAX\_NAME\_LEN], int \*size) {***

***char name[MAX\_NAME\_LEN];***

***int pos;***

***printf("Enter the student's name to insert: ");***

***scanf("%s", name);***

***printf("Enter the position (0-based index): ");***

***scanf("%d", &pos);***

***if (pos < 0 || pos > \*size) {***

***printf("Invalid position!\n");***

***return;***

***}***

***for (int i = \*size; i > pos; i--) {***

***strcpy(students[i], students[i - 1]);***

***}***

***strcpy(students[pos], name);***

***(\*size)++;***

***}***

***void deleteStudent(char students[][MAX\_NAME\_LEN], int \*size) {***

***int pos;***

***printf("Enter the position (0-based index) to delete: ");***

***scanf("%d", &pos);***

***if (pos < 0 || pos >= \*size) {***

***printf("Invalid position!\n");***

***return;***

***}***

***for (int i = pos; i < \*size - 1; i++) {***

***strcpy(students[i], students[i + 1]);***

***}***

***(\*size)--;***

***}***

***void searchStudent(char students[][MAX\_NAME\_LEN], int size) {***

***char name[MAX\_NAME\_LEN];***

***printf("Enter the student's name to search: ");***

***scanf("%s", name);***

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

***if (strcmp(students[i], name) == 0) {***

***printf("%s found at position %d\n", name, i);***

***return;***

***}***

***}***

***printf("%s not found!\n", name);***

***}***

***int main() {***

***char students[MAX][MAX\_NAME\_LEN];***

***int size = 0, choice;***

***do {***

***printf("\n1. Create the list of students\n");***

***printf("2. Insert a new student\n");***

***printf("3. Delete a student\n");***

***printf("4. Display student list\n");***

***printf("5. Search for a student\n");***

***printf("6. Exit\n");***

***printf("Enter your choice: ");***

***scanf("%d", &choice);***

***switch (choice) {***

***case 1:***

***createList(students, &size);***

***displayList(students, size);***

***break;***

***case 2:***

***insertStudent(students, &size);***

***displayList(students, size);***

***break;***

***case 3:***

***deleteStudent(students, &size);***

***displayList(students, size);***

***break;***

***case 4:***

***displayList(students, size);***

***break;***

***case 5:***

***searchStudent(students, size);***

***break;***

***case 6:***

***printf("Exiting the program...\n");***

***break;***

***default:***

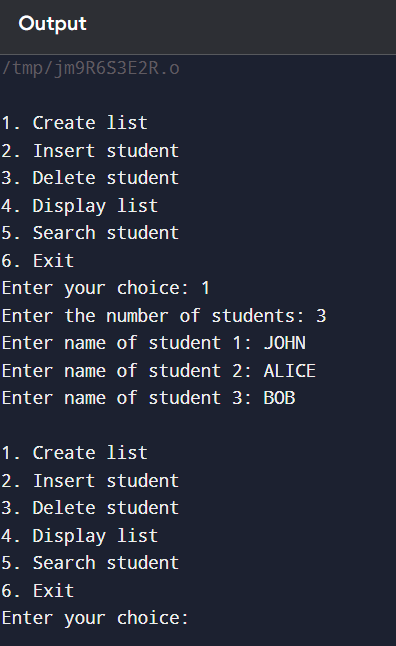
***printf("Invalid choice!\n");***

***}***

***} while (choice != 6);***

***return 0;***

***}***

****