**TASK1**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int summation(vector<int> arr) {

int sum = 0;

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

sum += arr[i];

}

return sum;

}

int maximum(vector<int> arr) {

return \*max\_element(arr.begin(), arr.end());

}

int main() {

int n;

cout << "Enter the size of the array: ";

cin >> n;

vector<int> arr(n);

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

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

cin >> arr[i];

}

cout << "The sum of the array is: " << summation(arr) << endl;

cout << "The maximum of the array is: " << maximum(arr) << endl;

return 0;

}

**TASK 2**

#include <stdio.h>

#include <string.h>

// Define the structure for the course type

struct course {

char course\_code[10];

char course\_name[50];

};

// Define the structure for the grade type

struct grade {

int mark;

char the\_grade;

};

// Define the structure for the student type

struct student {

char reg\_no[20];

char name[50];

int age;

struct course course;

struct grade grade;

};

// Define a global array of student structures, which can store up to 40 students

#define MAX\_STUDENTS 40

struct student students[MAX\_STUDENTS];

int count = 0; // Keep track of the number of students in the array

// Define a function to calculate the grade from the mark

char calculate\_grade(int mark) {

if (mark > 69) {

return 'A';

} else if (mark > 59) {

return 'B';

} else if (mark > 49) {

return 'C';

} else if (mark > 39) {

return 'D';

} else {

return 'E';

}

}

// Define a function to add a new student to the array

void add\_student() {

// Check if the array is full

if (count == MAX\_STUDENTS) {

printf("The array is full. Cannot add more students.\n");

return;

}

// Get the details of the new student from the user

struct student new\_student;

printf("Enter the registration number: ");

scanf("%s", new\_student.reg\_no);

printf("Enter the name: ");

scanf("%s", new\_student.name);

printf("Enter the age: ");

scanf("%d", &new\_student.age);

printf("Enter the course code: ");

scanf("%s", new\_student.course.course\_code);

printf("Enter the course name: ");

scanf("%s", new\_student.course.course\_name);

printf("Enter the mark: ");

scanf("%d", &new\_student.grade.mark);

// Calculate the grade from the mark

new\_student.grade.the\_grade = calculate\_grade(new\_student.grade.mark);

// Add the new student to the array

students[count] = new\_student;

count++;

// Display a success message

printf("The student has been added successfully.\n");

}

// Define a function to edit an existing student in the array

void edit\_student() {

// Get the registration number of the student to be edited from the user

char reg\_no[20];

printf("Enter the registration number of the student to be edited: ");

scanf("%s", reg\_no);

// Search for the student in the array

int i, found = 0;

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

if (strcmp(students[i].reg\_no, reg\_no) == 0) {

// The student is found

found = 1;

break;

}

}

// If the student is not found, display an error message and return

if (found == 0) {

printf("The student with the given registration number does not exist.\n");

return;

}

// If the student is found, display the current details and get the new details from the user

printf("The current details of the student are:\n");

printf("Registration number: %s\n", students[i].reg\_no);

printf("Name: %s\n", students[i].name);

printf("Age: %d\n", students[i].age);

printf("Course code: %s\n", students[i].course.course\_code);

printf("Course name: %s\n", students[i].course.course\_name);

printf("Mark: %d\n", students[i].grade.mark);

printf("Grade: %c\n", students[i].grade.the\_grade);

printf("Enter the new details of the student:\n");

printf("Enter the new name: ");

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

printf("Enter the new age: ");

scanf("%d", &students[i].age);

printf("Enter the new course code: ");

scanf("%s", students[i].course.course\_code);

printf("Enter the new course name: ");

scanf("%s", students[i].course.course\_name);

printf("Enter the new mark: ");

scanf("%d", &students[i].grade.mark);

// Calculate the new grade from the new mark

students[i].grade.the\_grade = calculate\_grade(students[i].grade.mark);

// Display a success message

printf("The student has been edited successfully.\n");

}

// Define a function to display all the students in the array

void display\_students() {

// Check if the array is empty

if (count == 0) {

printf("The array is empty. No students to display.\n");

return;

}

// Display the details of each student in the array

printf("The details of the students are:\n");

int i;

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

printf("Registration number: %s\n", students[i].reg\_no);

printf("Name: %s\n", students[i].name);

printf("Age: %d\n", students[i].age);

printf("Course code: %s\n", students[i].course.course\_code);

printf("Course name: %s\n", students[i].course.course\_name);

printf("Mark: %d\n", students[i].grade.mark);

printf("Grade: %c\n", students[i].grade.the\_grade);

printf("\n");

}

}

// Define a function to search for a student in the array by registration number

void search\_student() {

// Get the registration number of the student to be searched from the user

char reg\_no[20];

printf("Enter the registration number of the student to be searched: ");

scanf("%s", reg\_no);

// Search for the student in the array

int i, found = 0;

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

if (strcmp(students[i].reg\_no, reg\_no) == 0) {

// The student is found

found = 1;

break;

}

}

// If the student is not found, display an error message and return

if (found == 0) {

printf("The student with the given registration number does not exist.\n");

return;

}

// If the student is found, display the details of the student

printf("The details of the student are:\n");

printf("Registration number: %s\n", students[i].reg\_no);

printf("Name: %s\n", students[i].name);

printf("Age: %d\n", students[i].age);

printf("Course code: %s\n", students[i].course.course\_code);

printf("Course name: %s\n", students[i].course.course\_name);

printf("Mark: %d\n", students[i].grade.mark);

printf("Grade: %c\n", students[i].grade.the\_grade);

}

// Define the main function

int main() {

// Display a menu of options for the user

int choice;

do {

printf("Welcome to the student management system.\n");

printf("Please choose an option from the menu:\n");

printf("1. Add a new student\n");

printf("2. Edit an existing student\n");

printf("3. Display all the students\n");

printf("4. Search for a student by registration number\n");

printf("5. Exit the program\n");

printf("Enter your choice: ");

scanf("%d", &choice);

// Perform the corresponding action based on the user's choice

switch (choice) {

case 1:

add\_student();

break;

case 2:

edit\_student();

break;

case 3:

display\_students();

break;

case 4:

search\_student();

break;

case 5:

printf("Thank you for using the student management system. Goodbye.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

break;

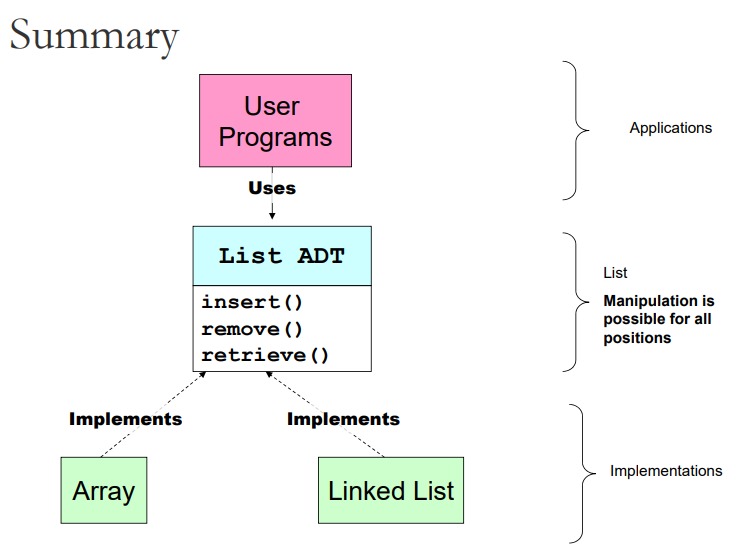
}

} while (choice != 5);

return 0;

}

**TASK 3**

****