calculator

#include<iostream>

using namespace std;

int addition(int a,int b)

{

return a+b;

}

int subtract(int a,int b)

{

return b-a;

}

int multiply(int a,int b)

{

return a\*b;

}

int divide(int a,int b)

{

return a/b;

}

int main()

{

char op;

int num1;

int num2;

cout<<"enter the values of num1 "<<endl;

cin>>num1;

cout<<"enter the value of num2"<<endl;

cin>>num2;

cout << "Enter operator (+, -, \*, /): ";

cin>>op;

switch (op) {

case '+':

cout <<"sum is"<< addition(num1,num2) << endl;

break;

case '-':

cout << "subtraction is " << subtract(num1,num2) << endl;

break;

case '\*':

cout << "multiplication is " << multiply(num1,num2) << endl;

break;

case '/':

cout << "division is " << divide(num1,num2) << endl;

break;

default:

cout << "Error! Invalid operator." << endl;

break;

}

return 0;

}

Random num

#include<iostream>

#include<cstdlib>

using namespace std;

int feedback(int a,int b)

{

cin>>b;

if(b==a)

{

cout<<"the guess is correct"<<endl;

}

if(b<a)

{

cout<<"the guess i too low"<<endl;

}

if(b>a)

{

cout<<"the guess is too high"<<endl;

}

}

int main()

{

int a=rand();

int n;

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

{

cout<<a<<endl;

}

feedback(a,b);

return 0;

}

To-do list

#include <iostream>

#include <vector>

#include <string>

using namespace std;

struct Task {

string description;

bool completed;

};

vector<Task> tasks;

void addTask(const string& description) {

Task task;

task.description = description;

task.completed = false;

tasks.push\_back(task);

cout << "Task added successfully.\n";

}

void viewTasks() {

cout << "Tasks:\n";

for (size\_t i = 0; i < tasks.size(); ++i) {

cout << i + 1 << ". " << tasks[i].description << " - " << (tasks[i].completed ? "Completed" : "Pending") << endl;

}

}

void markTaskCompleted(int index) {

if (index >= 1 && index <= tasks.size()) {

tasks[index - 1].completed = true;

cout << "Task marked as completed.\n";

} else {

cout << "Invalid task index.\n";

}

}

void removeTask(int index) {

if (index >= 1 && index <= tasks.size()) {

tasks.erase(tasks.begin() + index - 1);

cout << "Task removed successfully.\n";

} else {

cout << "Invalid task index.\n";

}

}

void displayMenu() {

cout << "To-Do List Manager\n";

cout << "1. Add Task\n";

cout << "2. View Tasks\n";

cout << "3. Mark Task as Completed\n";

cout << "4. Remove Task\n";

cout << "5. Exit\n";

}

int main() {

while (true) {

displayMenu();

int choice;

cout << "Enter your choice: ";

cin >> choice;

cin.ignore(); // Clear input buffer

switch (choice) {

case 1: {

cout << "Enter task description: ";

string description;

getline(cin, description);

addTask(description);

break;

}

case 2:

viewTasks();

break;

case 3: {

cout << "Enter task index to mark as completed: ";

int index;

cin >> index;

markTaskCompleted(index);

break;

}

case 4: {

cout << "Enter task index to remove: ";

int index;

cin >> index;

removeTask(index);

break;

}

case 5:

return 0;

default:

cout << "Invalid choice. Please try again.\n";

break;

}

}

return 0;

}